Applying Minimum Spanning Tree for Portfolio Diversification and Market Analysis in the LQ45 Index

Peter Wongsoredjo - 13523039[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523039@itb.ac.id , [2]peterwongsoredjo@gmail.com

***Abstract*—Leveraging the Graph theory, the study applies the Minimum Spanning Tree algorithm to enhance portfolio diversification, and analyze the market in the LQ45 Index. By identifying minimally correlated stocks, the MST algorithm is able to construct diversified portfolios and provide stock recommendations. Results in the research highlight the algorithm's efficacy in optimizing risk-return profile while its limitations in the financial market dynamics are acknowledged. The research in this paper offers a practical framework and a new perspective towards the financial market by simplifying the complex connection that the network of stocks in the LQ45 index have.**

***Keywords*—Minimum Spanning Tree, Portfolio Diversification, LQ45 Index**

I. INTRODUCTION

Liquidity 45 or best known as the LQ45, consists of 45 stocks selected for their liquidity and significant market capitalization, being a prominent benchmark for the Indonesian stock market. As a representative index, it serves as a critical reference point for investors aiming to understand market trends and construct portfolios.

Investors and analysts rely heavily on diversification to balance out the risks and return in their portfolios. However, as financial markets grow increasingly complex, conventional diversification methods often fail in identifying clusters of highly correlated stock, leading to suboptimal diversification and higher portfolio risks. This limitation underscores the need for an advanced tool to simplify the connection between individual stocks in the vast stock market, in order to gain deeper market insights into market structures and avoid hidden risks during portfolio construction.

Graph theory offers a robust mathematical framework for analyzing complex networks, making it an ideal tool for financial market analysis. The Minimum Spanning Tree stands out for its ability in simplifying fully connected graphs by preserving only the most critical relationships. The LQ45, with its sectoral diversity, presents an opportunity for an application of the Minimum Spanning Tree (MST), to uncover clusters of highly correlated stocks, identify diversification opportunities, and provide actionable insights into market dynamics.

The research in this paper aims to demonstrate the application of the MST in optimizing portfolio diversification and enhancing market analysis within the LQ45 Index. This study aims to simplify the complex stocks network, uncovering clusters of highly correlated stocks, and identifying diversification opportunities by constructing an MST. This framework offers actionable insights for investors, facilitating better decision making and a more effective portfolio management of the Indonesian stock market.

II. THEORETICAL FRAMEWORK

2.1 Graph

Graph theory is a branch of mathematics that studies objects (vertices) and their relationships (edges). A graph is defined as $G = (V, E)$, where

- V is the sets of vertices
- E is the sets of edges that connects vertices

Graphs are often classified into various types, including directed and undirected graphs, based on whether or not the relationships have a direction. A weighted graph is a graph where each edge $(u, v) \in E$ is assigned a numerical value or weight $w = (u, v)$.

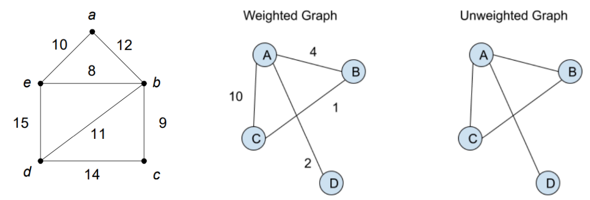

Figure 2.1 Graph and Weighted Graph
Source:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

2.2 Minimum Spanning Tree

A tree is a connected, undirected graph that contains no cycle, meaning there are no closed loops formed by the edges. Key properties of a tree include:

- a tree with n vertices has exactly n-1 edges
- there is a unique path between any two vertices in a tree

A spanning tree is a subgraph of a graph G that is a tree, yet it includes all the vertices of G, where for a given graph, there may be multiple spanning trees, each connected and acyclic.

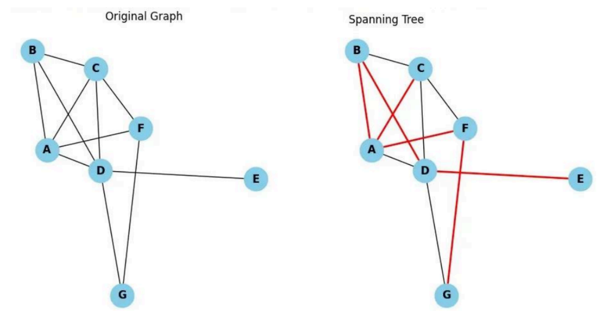

Figure 2.2 Minimum Spanning Tree
Source:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

Given a weighted graph, a Minimum Spanning Tree (MST) is a specific type of spanning tree where the total edge weight is minimized. MST is useful for problems where the goal is to connect all the nodes with the least edge cost.

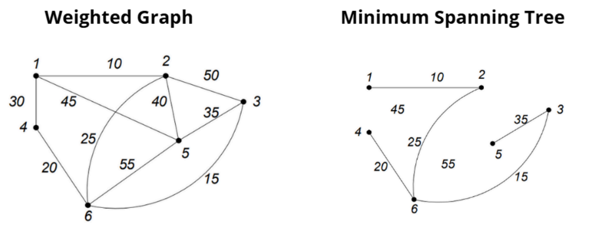

Figure 2.3 Minimum Spanning Tree
Source:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

There are two common algorithms used to construct a minimum spanning tree, the Prim and the Kruskal algorithm. To construct a minimum spanning tree using the Prim algorithm there are three steps.

1. Select an edge with the smallest weight from the Graph $G = (V, E)$, and include it in T.
2. Find the edge (u, v) with the lowest weight, adjacent to the vertex in T, yet keeps T acyclic. Include (u, v) in T.
3. Repeat step 2 for n -2 times.

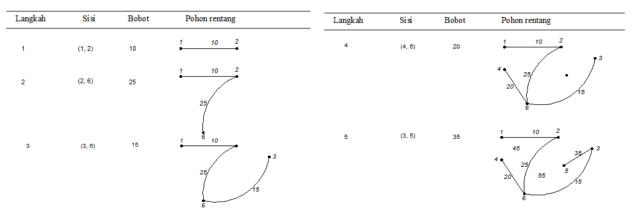

Figure 2.4 Prim Algorithm to Construct a Minimum Spanning Tree in the Figure 2.3

Source:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

2.3 Pearson Correlation Coefficient

The Pearson Correlation Coefficient (r), is a statistical measure used to quantify the linear relationship between two variables. The formula for calculating the Pearson Correlation Coefficient between two variables X and Y is

$$r_{XY} = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}}$$

The coefficient result ranges from -1 to 1, where:

- $r = 1$, indicates a perfect positive correlation, meaning both variables move in the same direction
- $r = -1$, indicates a perfect negative correlation, meaning one variables move in the opposite direction of the other
- $r = 0$, indicates no linear relationship between the two variables

2.4 MST in Financial Markets

Consider a stock as a vertex, and the correlation it has with another stock as an edge. In analyzing these stocks, MST enables investors to highlight only the most significant relationships between stocks by retaining edges with the smallest weight, derived from transformed correlations.

By organizing stocks into clusters, MST helps avoid overconcentration and identifies outlier stocks for inclusion in a balanced portfolio. Stocks located in different branches that exhibit lower correlations, reduces unsystematic risk, and enhances portfolio resilience, making MST an essential tool for constructing diversified portfolios that optimize returns while minimizing risk. Additionally, MST captures behavioral patterns like herding, where groups of stocks move together, reflecting collective investor behavior.

III. IMPLEMENTATION

3.1 LQ45 Stock Index

The LQ45 Stock Index is a benchmark index of the Indonesian Stock Exchange (IDX), composed of 45 stocks selected due to their high liquidity, market capitalization, and strong fundamentals. The LQ45 index is re-evaluated every six months to ensure it reflects the current market dynamics and includes only the most actively traded stocks.

This index is particularly relevant for financial analysis due to its diverse mix of sectors, including banking, mining, consumer goods, and telecommunications, and representation of market leaders, making it an ideal dataset for studying stock relationships and portfolio diversification optimization. The LQ45's sectoral diversity and high liquidity provide a robust foundation for an analysis using the Minimum Spanning Tree, ensuring meaningful and actionable insights.

IDX
Indonesia Stock Exchange
Bursa Efek Indonesia

#AKU INVESTOR SAHAM

Lampiran Pengumuman BEI No. Peng-00220/BEI.POP/10-2024 tanggal 25 Oktober 2024

Nama Indeks : **LQ45**
Evaluasi : **Mayor**
Periode Efektif Konsituen : **01 November 2024 s.d. 31 Januari 2025**
Periode Efektif Jumlah Saham Penghitungan Indeks : **01 November 2024 s.d. 31 Januari 2025**

No.	Kode	Rasio Free Float	Jumlah Saham untuk Indeks (lembar)			Bobot pada Indeks		
			Pra Evaluasi	Pasca Evaluasi * (15% Cap)	Keterangan Tetap/Naik/ Turun/Baru	Pra Evaluasi	Pasca Evaluasi	Keterangan Tetap/Naik/ Turun/Baru
1	ACES	40.02%	6,863,430,000	6,851,579,958	Turun	0.31%	0.32%	Naik
2	**ADMR**	**13.46%**	-	**5,502,761,820**	**Baru**	-	**0.40%**	**Baru**
3	ADRO	33.73%	12,765,597,434	10,374,898,008	Turun	2.29%	1.95%	Turun
4	AKRA	32.87%	6,824,981,364	6,598,151,101	Turun	0.48%	0.48%	Tetap
5	AMMN	17.98%	17,382,616,772	13,038,775,535	Turun	7.96%	6.25%	Turun
6	AMRT	42.39%	18,843,818,871	17,602,236,271	Turun	3.26%	3.18%	Turun
7	ANTM	34.84%	8,372,318,430	8,372,318,430	Tetap	0.67%	0.70%	Naik
8	ARTO	26.85%	3,801,382,135	3,683,403,476	Turun	0.55%	0.55%	Tetap

Figure 3.1 An Overview of LQ45
Source:
https://www.idx.co.id/id/data-pasar/data-saham/indeks-saham

3.2 Data Scraping

Historical price data of the 45 stocks that comprises the LQ45 Index were collected from *investing.com* to conduct the analysis. Each stock's data was collected by the daily intervals, starting from 1 January 2020, until 1 January 2025, on the day that the Indonesian stock market was open. The collected data includes essential information such as dates, closing prices, and percentage changes, formatted as CSV files. The 5 year range provides a comprehensive historical dataset in order to capture the market dynamics and stocks correlation. For stocks that IPO'd within the five year period and don't have a complete dataset, the data is still retained in order to ensure the analysis is complete.

3.3 Code Implementation

A. Data Loading and Preprocessing

The first step in the code implementation process involves leveraging python's data analysis libraries such as Numpy and also Pandas in order to load and preprocess the data for further analysis. To efficiently manage the data, each stock's DataFrame is stored in a dictionary, indexed by its ticker symbol.

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#LOADING ALL THE FILES
def load_csv_files(directory):
    dataframes = {}  # Dictionary to store dataframes
    for file in os.listdir(directory):
        if file.endswith('.csv'):
            df_name = file.replace('.csv', '')
            dataframes[df_name] = pd.read_csv(os.path.join(directory, file))
    return dataframes
```

```
# Directory containing the CSV files
csv_directory = 'C:\\Users\\ASUS\\Documents\\SEMESTER 3\\MATDIS\\MAKALAH\\KODE\\DATABASE'

# Load all CSV files into a dictionary of dataframes
dataframes = load_csv_files(csv_directory)

# Create a list of dataframes
dataframe_list = list(dataframes.values())
```

Figure 3.2 Data Loading

Once the data is loaded, preprocessing begins to clean and standardize the datasets. Missing or non-numeric entries are handled to prevent errors in subsequent analysis. The datasets are then aligned by merging them on the Date column using an outer join, retaining only the overlapping dates across all stocks for the Change% column. This ensures that correlation calculations are performed on a consistent time frame. The cleaned and aligned percentage changes are consolidated into a single DataFrame, with dates as the index and stocks as columns. This combined dataset serves as the foundation for constructing the correlation matrix and subsequent Minimum Spanning Tree analysis.

```
for name, df in dataframes.items():
    df['Date'] = pd.to_datetime(df['Date'])
    df['Change %'] = df['Change %'].str.replace('%', '').astype(float)
    dataframes[name] = df.set_index('Date')  # Set Tanggal as index for alignment

combined_data = pd.concat(
    {name: df['Change %'] for name, df in dataframes.items()}, axis=1, join="outer"
)

if isinstance(combined_data.columns, pd.MultiIndex):
    combined_data.columns = combined_data.columns.droplevel(0)  # Drop unnecessary level
else:
    combined_data.columns = dataframes.keys()  # Assign column names directly
```

Figure 3.3 Data Preprocessing

B. Correlation Matrix Construction

The correlation matrix quantifies the relationships between stocks in the LQ45 Index using the Pearson correlation coefficient. For each pair of stocks, their daily percentage changes over the specified period are used to calculate the correlation coefficient, resulting in a symmetric matrix where each element represents the correlation between two stocks, ranging from -1 to 1. This correlation matrix is then saved to csv for further MST analysis.

```
def pairwise_correlation(data, stock1, stock2):
    pair_data = data[[stock1, stock2]].dropna()  # Drop rows with NaN
    if pair_data.empty:  # If no valid rows, return NaN
        return np.nan
    return pair_data.corr().iloc[0, 1]  # Return the correlation coefficient
```

```
#Stopping per pair when encountering NaN
correlation_matrix = pd.DataFrame(index=combined_data.columns, columns=combined_data.columns)

for stock1 in combined_data.columns:
    for stock2 in combined_data.columns:
        correlation_matrix.loc[stock1, stock2] = pairwise_correlation(combined_data, stock1, stock2)
```

```
correlation_matrix = correlation_matrix.astype(float)

# Display the correlation matrix
print("\nCorrelation Matrix:")
print(correlation_matrix)
```

Figure 3.4 Correlation Matrix Construction

C. Minimum Spanning Tree Construction

The graph is constructed from the correlation matrix, which quantifies the transformed correlations between stocks in the LQ45 Index. Each stock is represented as a node, and edges between nodes represent the correlations. Using the NetworkX library, a weighted graph is generated, with the correlation matrix providing edge weights. This graph serves as the foundation for constructing the Minimum Spanning Tree (MST), representing the stock network in its fully connected form before being simplified by the MST algorithm.

```python
import pandas as pd
import numpy as np
import networkx as nx

correlation_file = "./correlation_matrix.csv"
correlation_matrix_df = pd.read_csv(correlation_file, index_col=0)

G = nx.from_pandas_adjacency(correlation_matrix_df)
```

Figure 3.5 Graph Construction

Prim's algorithm is then applied to the weighted graph to construct the Minimum Spanning Tree (MST). Starting from the edge with the lowest weight of the graph, the algorithm iteratively adds the smallest available edge that connects a new node to the growing tree, while ensuring no cycles are formed, until all nodes are included, resulting in a tree that minimizes the total edge weight while maintaining connectivity. The MST effectively reduces the complexity of the original graph, retaining only the most significant relationships between stocks, which are crucial for analyzing market clusters and dependencies.

```python
def prim_algorithm(G):
    mst_edges = []  # Edge List of the MST
    visited = set()  # Set of visited nodes
    edges = []  # Priority queue for edges

    min_edge = None
    min_weight = float('inf')
    for u, v, data in G.edges(data=True):
        if data['weight'] < min_weight:
            min_weight = data['weight']
            min_edge = (u, v, data['weight'])

    if min_edge:
        u, v, weight = min_edge
        mst_edges.append(min_edge)
        visited.update([u, v])

        for neighbor, data in G[u].items():
            if neighbor not in visited:
                edges.append((data['weight'], u, neighbor))
        for neighbor, data in G[v].items():
            if neighbor not in visited:
                edges.append((data['weight'], v, neighbor))

        edges = sorted(edges)

    while edges:
        weight, u, v = edges.pop(0)

        if v not in visited:
            mst_edges.append((u, v, weight))
            visited.add(v)
            for neighbor, data in G[v].items():
                if neighbor not in visited:
                    edges.append((data['weight'], v, neighbor))

            edges = sorted(edges)

    return mst_edges

mst = prim_algorithm(G)
print("Minimum Spanning Tree:")
for index, edge in enumerate(mst, start=1):
    print(f"{index}. {edge[0]} -- {edge[1]} (Weight: {edge[2]:.4f})")
```

Figure 3.6 MST Construction Using the Prim Algorithm

This Prim algorithm is then modified to be able to show the best MST for the top n many edge weights, providing usage for seeking out the top n combination of stocks within the LQ45 index.

```python
def find_best_mst_with_minimal_top_n_sum(G, n):
    def prim_with_start_edge(G, start_edge):
        mst_edges = []
        visited = set()
        edges = []

        # Start with each edge
        u, v, weight = start_edge
        mst_edges.append(start_edge)
        visited.update([u, v])

        for neighbor, data in G[u].items():
            if neighbor not in visited:
                edges.append((data['weight'], u, neighbor))
        for neighbor, data in G[v].items():
            if neighbor not in visited:
                edges.append((data['weight'], v, neighbor))
        edges = sorted(edges)

        while edges and len(mst_edges) < (n-1):
            weight, u, v = edges.pop(0)
            if v not in visited:
                mst_edges.append((u, v, weight))
                visited.add(v)
                for neighbor, data in G[v].items():
                    if neighbor not in visited:
                        edges.append((data['weight'], v, neighbor))
                edges = sorted(edges)

        return mst_edges

    best_result = None
    lowest_sum = float('inf')

    for u, v, data in G.edges(data=True):
        if u == v:
            continue

        start_edge = (u, v, data['weight'])
        mst_edges = prim_with_start_edge(G, start_edge)

        top_n_sum = sum(edge[2] for edge in mst_edges[:n-1])

        if top_n_sum < lowest_sum:
            lowest_sum = top_n_sum
            best_result = {
                "starting_edge": start_edge,
                "mst_edges": mst_edges,
```

Figure 3.7 Best MST Derived from the Top n Result

The prim algorithm is further modified to receive an input of a portfolio that an investor already has with stocks included in the LQ45, and the MST is further utilized to recommend additional stocks for portfolio diversification. Stocks from other branches, which exhibit a weaker correlation with the portfolio, are then suggested for diversification. To make sure that the stocks recommended aren't distorted, the stocks are tested with each of the existing stocks.

```python
def recommend_stocks_for_diversification(G, existing_stocks):
    # Stocks Validation
    if not all(stock in G.nodes for stock in existing_stocks):
        raise ValueError("Some of the existing stocks are not in the graph.")

    # MST for the existing portfolio
    subgraph = G.subgraph(existing_stocks)
    mst_edges = list(nx.minimum_spanning_edges(subgraph, data=True))
    mst_nodes = set(existing_stocks)

    print("Minimum Spanning Tree (MST) for the existing portfolio:")
    for u, v, data in mst_edges:
        print(f"({u}, {v}, {data['weight']})")

    potential_edges = []

    for node in G.nodes:
        if node not in mst_nodes:
            for mst_node in mst_nodes:
                if G.has_edge(node, mst_node):
                    weight = G[node][mst_node]['weight']
                    potential_edges.append((weight, mst_node, node))

    ranked_stocks = sorted(potential_edges, key=lambda x: x[0])

    unique_ranked_stocks = []
    seen_nodes = set()
    for weight, existing_stock, new_stock in ranked_stocks:
        if new_stock not in seen_nodes:
            unique_ranked_stocks.append((weight, existing_stock, new_stock))
            seen_nodes.add(new_stock)

    return unique_ranked_stocks

existing_stocks = ['BBCA', 'ANTM', 'BBRI', 'EXCL', 'ARTO']  # Example existing portfolio

recommended_stocks = recommend_stocks_for_diversification(G, existing_stocks)

print("Recommended stocks to add (sorted by suitability):")
for weight, existing_stock, new_stock in recommended_stocks:
    print(f"Stock: {new_stock}, Connected to: {existing_stock}, Distance: {weight:.4f}")
```

Figure 3.8 Stocks Recommendation using MST

3.4 Data Visualization

Using the Matplotlib and Seaborn libraries, various data visualizations are generated that provide intuitive insights to support the analysis. Some data visualization includes heatmap of the LQ45 stocks correlation, the cumulative portfolio value graph, and the daily volatility graph.

```python
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=False, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix for LQ45 Stocks (Stopping at NaN)")
plt.show()
```

Figure 3.8 Heatmap of the LQ45 Stocks Correlation

```python
plt.figure(figsize=(12, 6))
combined_data['Cumulative Portfolio Value'].plot(color='blue', label='Portfolio Value')
plt.title("Cumulative Portfolio Performance")
plt.xlabel("Date")
plt.ylabel("Portfolio Value")
plt.axhline(initial_investment, color='red', linestyle='--', linewidth=1, label='Initial Investment')
plt.legend()
plt.grid()
plt.show()

# Final portfolio value
final_value = combined_data['Cumulative Portfolio Value'].iloc[-1]
print(f"Final portfolio value: {final_value:.2f}")
```

Figure 3.9 Cumulative Portfolio Value

```python
plt.figure(figsize=(12, 6))
combined_data['Portfolio Change%'].plot(color='blue', label='Portfolio Change%')
plt.axhline(0, color='red', linestyle='--', linewidth=1, label='Zero Line')
plt.title("Combined Portfolio Daily Volatility")
plt.xlabel("Date")
plt.ylabel("Percentage Change (%)")
plt.legend()
plt.grid()
plt.show()
```

Figure 3.10 Daily Volatility Graph

The complete Python code, along with the resulting plots, can be reviewed in detail in the appendix.

IV. TESTING AND ANALYSIS

4.1 LQ45 Correlation Matrix

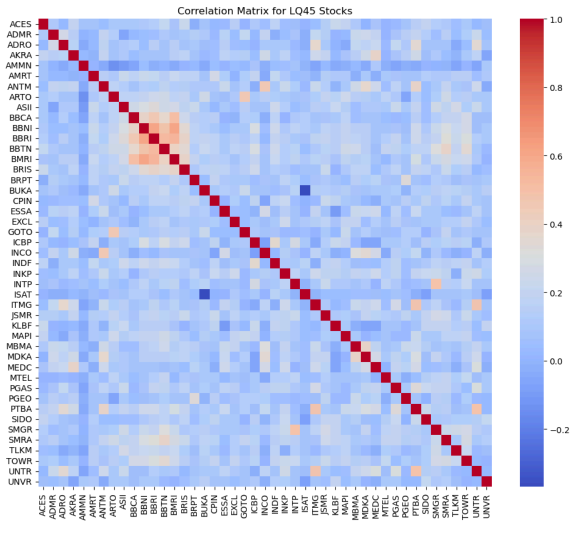

Figure 4.1 LQ45 Stocks Correlation Matrix Heatmap

The figure above shows the LQ45 Correlation Matrix, visualized as a heatmap, which provides an overview of the relationships between two stocks. Each cell represents the Pearson correlation coefficient between two stocks, stocks with strong positive correlations are indicated by red and orange hues, while stocks with weak or no correlations are displayed in shades of blue. The diagonal line of the heatmap, where all values are 1, represents the self-correlation of each stock.

From the heatmap, several key patterns emerge. The cluster of the red and orange hues in the top left side of the map represents the Banking stocks, that include BBCA, BBRI, BMRI, BBNI, etc. These stocks that are from the same sector show a positive correlation with each other, with each stock's price influenced by the same economic factors. Conversely, the majority of the stocks are within the shades of blue, indicating that each stock in the index isn't highly correlated to each other, especially stocks from opposing sectors in the Indonesian Stock Market. The correlation matrix above highlights how the LQ45 provides many potential as candidates for portfolio diversification. This heatmap serves as a critical first step in the analysis, offering insights into stock interdependencies.

4.2 The Minimum Spanning Tree of LQ45

The Minimum Spanning Tree of the LQ45 index, as shown by Figure 4.2 below represents the least significant relationships between stocks based on their correlations. By starting out with the two stocks with the two lowest correlation edge, i.e. BUKA and ISAT, the algorithm continues to add the lowest edges available until all the vertices (stocks) are included in the tree. While it's possible to adopt all 45 stocks in the LQ45 index to the portfolio, the MST's based usage is for targeted stocks, meaning this spanning tree can't really give investors a good market insight, meaning further MST implementations are needed.

```
Minimum Spanning Tree:
1. BUKA -- ISAT (Weight: -0.3689)
2. ISAT -- SIDO (Weight: -0.0806)
3. ISAT -- UNVR (Weight: -0.0714)
4. UNVR -- AKRA (Weight: -0.0584)
5. UNVR -- ANTM (Weight: -0.0425)
6. BUKA -- AMMN (Weight: -0.0386)
7. AMMN -- ARTO (Weight: -0.1355)
8. AMMN -- ASII (Weight: -0.1024)
9. AMMN -- TLKM (Weight: -0.0682)
10. AMMN -- UNTR (Weight: -0.0570)
11. AMMN -- ITMG (Weight: -0.0569)
12. ARTO -- PGEO (Weight: -0.0537)
13. AMMN -- BBCA (Weight: -0.0520)
14. AMMN -- PGAS (Weight: -0.0509)
15. ASII -- ADMR (Weight: -0.0495)
16. AMMN -- KLBF (Weight: -0.0471)
17. KLBF -- ESSA (Weight: -0.1003)
18. KLBF -- MDKA (Weight: -0.0652)
19. KLBF -- MEDC (Weight: -0.0612)
20. MEDC -- ICBP (Weight: -0.0648)
21. ICBP -- INCO (Weight: -0.0758)
22. INCO -- CPIN (Weight: -0.0619)
23. AMMN -- ADRO (Weight: -0.0463)
24. INCO -- MTEL (Weight: -0.0449)
25. AMMN -- BMRI (Weight: -0.0439)
26. ADRO -- MAPI (Weight: -0.0410)
27. INCO -- AMRT (Weight: -0.0406)
28. AMMN -- PTBA (Weight: -0.0367)
29. AMMN -- BBRI (Weight: -0.0290)
30. KLBF -- MBMA (Weight: -0.0259)
31. TLKM -- INKP (Weight: -0.0247)
32. ADMR -- INDF (Weight: -0.0238)
33. ISAT -- INTP (Weight: -0.0105)
34. AMMN -- SMRA (Weight: -0.0073)
35. BBCA -- EXCL (Weight: -0.0063)
36. AMMN -- BBTN (Weight: 0.0018)
37. MDKA -- ACES (Weight: 0.0026)
38. ACES -- BRPT (Weight: -0.0283)
39. AMMN -- SMGR (Weight: 0.0042)
40. INCO -- TOWR (Weight: 0.0105)
41. SIDO -- GOTO (Weight: 0.0114)
42. KLBF -- BRIS (Weight: 0.0118)
43. AMMN -- JSMR (Weight: 0.0226)
44. ADMR -- BBNI (Weight: 0.0401)
```

Figure 4.2 LQ45 Stocks Minimum Spanning Tree

4.3 Further MST Implementation

While the application of the Multiple Spanning Tree theory on the LQ45 is able to show us the Prim algorithm's result as a list of the correlation of each stock, which means that the 45 stocks would still be interconnected when all adopted in a single portfolio, diminishing the use of MST. In order to maximize the potential of the MST for portfolio diversification, further implementation of the MST algorithm is done.

A. N Stock Selection for Portfolio Diversification

This implementation builds on the MST to identify a subset of stocks from the LQ45 index, to form the most diverse portfolio by iterating over all possible starting edges. For each iteration, the algorithm calculates the cumulative weight of the selected stocks, prioritizing those with minimal correlation.

```
Best Starting Edge: AMMN -- BUKA (Weight: -0.0386)
Best Combination of 10 stocks in the LQ45 (9 edges):
1. AMMN -- BUKA (Weight: -0.0386)
2. BUKA -- ISAT (Weight: -0.3689)
3. AMMN -- ARTO (Weight: -0.1355)
4. AMMN -- ASII (Weight: -0.1024)
5. ISAT -- SIDO (Weight: -0.0806)
6. ISAT -- UNVR (Weight: -0.0714)
7. AMMN -- TLKM (Weight: -0.0682)
8. UNVR -- AKRA (Weight: -0.0584)
9. AMMN -- UNTR (Weight: -0.0570)
Smallest Sum of First 9 Edge Weights: -0.9809
Unique Stocks in the Best Combination (10 stocks): AKRA, AMMN, ARTO, ASII, BUKA, ISAT, SIDO, TLKM, UNTR, UNVR
```

Figure 4.3 10 Most Diverse Set of Stocks in LQ45

The combination of stocks displayed in the figure above represents the best 10-stock portfolio combination from the LQ45, optimized for maximum diversification. The cumulative weight of the first 10 edges, representing the correlation between stocks, is the lowest among all possible combinations. By leveraging the MST method, this portfolio effectively selects stocks that are least correlated with one another, which in return should result in a low cumulative volatility rate during the long run.

B. Diversification Recommendation for Existing Portfolio

The second implementation extends the MST's utility to recommend additional stocks for an existing portfolio. For each candidate stock not in the portfolio, the algorithm iteratively connects it to the existing portfolio and evaluates its impact on all the MST's total weight. Stocks that introduce the least correlation are ranked highest, making them ideal additions for diversification. Consider a portfolio consisting of the stocks BBCA, ANTM, BBRI, EXCL, and ARTO, all LQ45 stocks. In order to give out the best stock to recommend which maximizes the portfolio diversification, the algorithm first creates a Minimum Spanning Tree of the existing stocks. It then iterates all the remaining stock in the LQ45, and creates an MST with each, connecting to different stocks in the portfolio. The complete MST's weight is then compared to other remaining stocks.

```
Minimum Spanning Tree (MST) for the existing portfolio:
(BBCA, EXCL, -0.0063139324869497)
(EXCL, ANTM, 0.0661156321514626)
(EXCL, ARTO, 0.1169519504716866)
(BBRI, ANTM, 0.125555674025191)
Recommended stocks to add (sorted by suitability):
Stock: AMMN, Connected to: ARTO, Correlation: -0.1355
Stock: PGEO, Connected to: ARTO, Correlation: -0.0537
Stock: UNVR, Connected to: ANTM, Correlation: -0.0425
Stock: INCO, Connected to: BBCA, Correlation: -0.0254
Stock: ICBP, Connected to: ANTM, Correlation: -0.0220
Stock: KLBF, Connected to: ANTM, Correlation: -0.0023
Stock: AMRT, Connected to: EXCL, Correlation: 0.0048
Stock: INKP, Connected to: BBCA, Correlation: 0.0086
Stock: ESSA, Connected to: BBCA, Correlation: 0.0107
```

Figure 4.4 LQ45 Stock Recommendation for an Existing Portfolio

The results in Figure 4.4 above shows the recommendation stock in order to enhance the existing portfolio's diversification. The algorithm has identified AMMN as the best stock to add to the existing portfolio, with a total correlation of 1.3011 with all the possible connections displayed. These connections indicate that AMMN has relatively low correlation with most of the existing stocks, making it an ideal addition to increase the portfolio diversification, compared to other available stocks.

4.4 Performance Evaluation and Comparative Analysis

A. Stock Selection Analysis

In order to test the algorithm's proficiency in constructing a diverse portfolio, two portfolios are created, one being the algorithm's best 10 stocks combination from the LQ45 index, while the second portfolio is a portfolio composed of the stocks in the banking sector listed in the LQ45.

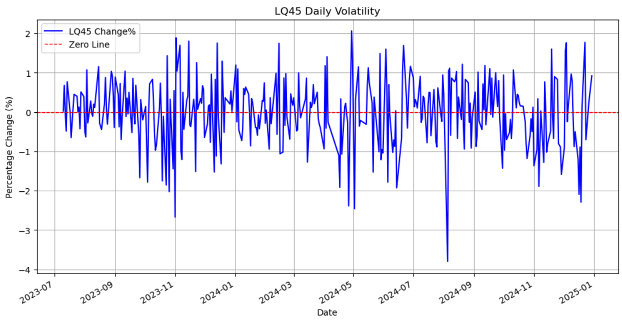

Figure 4.5 LQ45 Daily Volatility

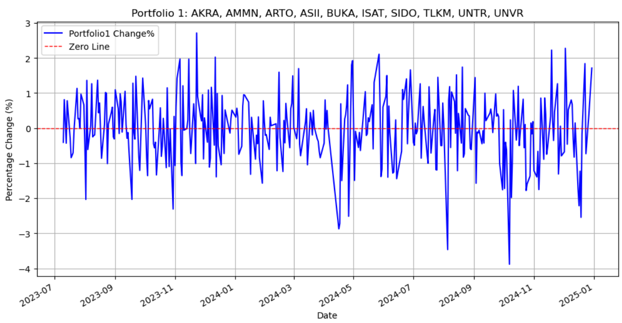

Figure 4.6 Portfolio 1 Daily Volatility

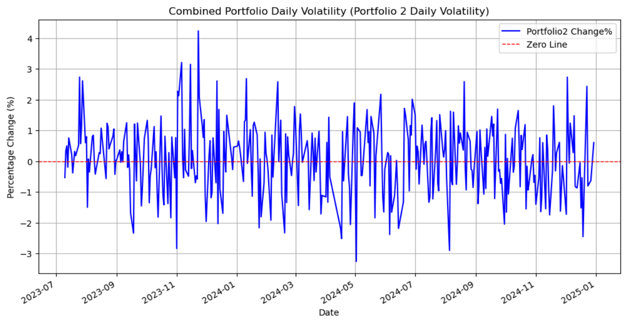

Figure 4.7 Portfolio 2 Daily Volatility

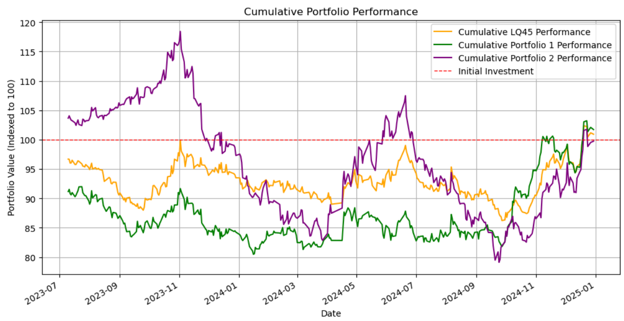

Figure 4.8 Cumulative Portfolio Performance

The stock selection analysis provided by the figures above provides key insights for the algorithm’s proficiency. Portfolio 1, constructed by the algorithm, exhibits noticeably lower volatility compared to Portfolio 2, a group with high internal correlation. However, when compared to the LQ45 daily volatility index as shown in Figure 4.5, Portfolio 1’s daily volatility mirrors that of the index, but with a slight tendency towards positive changes, indicating a balanced yet potentially optimistic upside.

A closer look at the cumulative portfolio performance, shown in Figure 4.8, shows that the LQ45 index emerges as the least volatile out of the three. The performance of portfolio 1 shows a similar movement to the LQ45 index, beating portfolio 2’s volatility. This aligns with the Efficient Market Hypothesis (EMH), which suggests that the LQ45 index, with its diverse composition of 45 stocks, benefits from the market's natural efficiency, achieving a more stable performance than the narrower, less diversified portfolios. Nevertheless, the algorithm showed its effectiveness in constructing a diverse portfolio, enough to beat the clustered up group.

B. Diversification Recommendation

The second test aims to evaluate the effectiveness of the algorithm's recommendation in improving portfolio diversification and reducing volatility. Portfolio A consists of a randomly selected group of stocks, including BBCA, ANTM, EXCL, ARTO, and BBRI, representing a moderately diversified set. Portfolio B, on the other hand, is an enhanced version of Portfolio A, with the addition of AMMN, as recommended by the algorithm to maximize diversification and minimize correlation among the stocks.

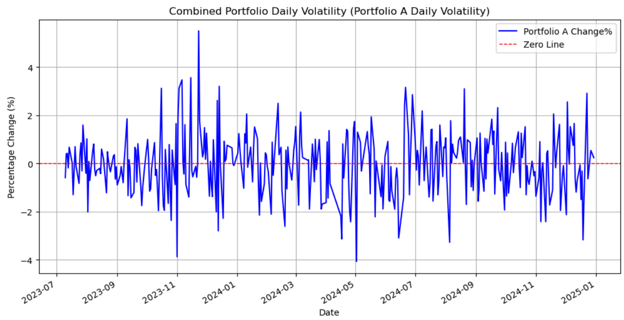

Figure 4.9 Portfolio A Daily Volatility

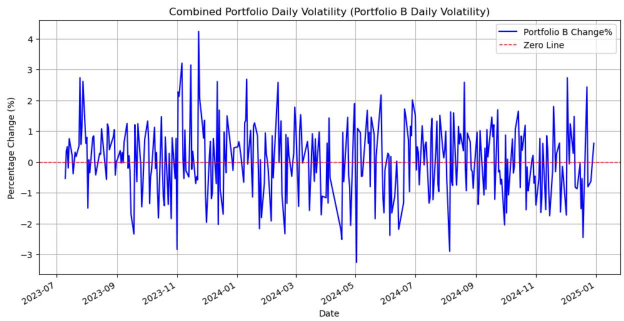

Figure 4.10 Portfolio B Daily Volatility

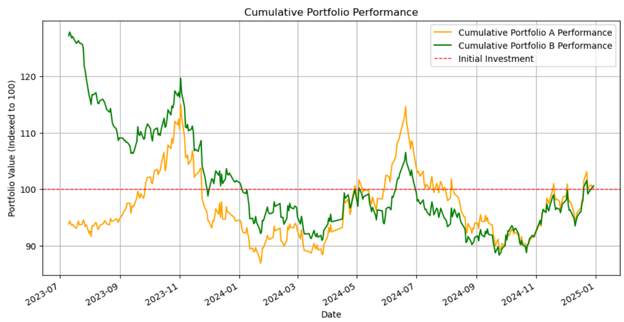

Figure 4.11 Cumulative Portfolio A vs B Performance

The results in the Figures above indicate a notable diversification improvement in Portfolio B’s performance after the addition of AMMN. A review of the daily volatility graph shows a significant reduction in extreme price movements, with the maximum drop off in Portfolio B capped at approximately 3%, as compared to Portfolio A’s 4%. While the adjustment reduces the portfolio’s susceptibility to sharp losses, it also limits the potential for large positive movement.

The cumulative performance of both portfolio A, and B follows a similar trajectory, yet Portfolio B exhibits a more stable pattern, showing the algorithm’s effectiveness in improving portfolio diversification, and achieving a more balanced risk-return profile for the portfolio.

4.5 Implication and Recommendation

The Minimum Spanning Tree algorithm, when applied to the LQ45 index shows efficacy in constructing a portfolio with a strong diversification. By selecting a number of stocks based on their correlation to each other, the algorithm effectively composed a portfolio that outperforms clusters of highly correlated stocks in terms of stability and reduced volatility, while still showing strong similarity to the LQ45 index itself, a benchmark for diversified investments.

In addition to being able to construct a diversified portfolio, the algorithm also demonstrated its capacity in recommending unrelated stocks to existing portfolios in order to further enhance diversification. By analyzing the correlation between stocks, the algorithm is able to identify stocks with minimal connections to the portfolio, recommending an addition that successfully reduces volatility and achieves a more balanced risk-return profile.

However, it is essential to recognize the inherent limitations of using MST for portfolio construction. Stocks are inherently interdependent, making their relationships more complex than the MST assumes. Unlike classical MST problems, such as the traveling postman problem, it's impossible to draw a clear graph of networks between stocks in LQ45. Thus, while the MST algorithm shows a robust performance in selecting stocks and giving recommendations during the tests, it should be complemented with additional methods to address these nuances, and optimize portfolio performance.

V. CONCLUSION

The application of Minimum Spanning Tree for portfolio diversification and market analysis in the LQ45 index demonstrates potential to simplify complex relationships into actionable insights. Utilizing MST, the algorithm in this research is able to construct diversified portfolios and provide recommendation to existing portfolios by identifying stocks with minimal correlation, enabling a more enhanced portfolio with a more balanced risk-reward ratio. The algorithm proved capable of outperforming clusters of highly correlated stocks while delivering results comparable to the broader LQ45 index, which serves as a benchmark for diversified investments.

However, the study also acknowledges the limitations of using MST for portfolio construction. The absence of a clear "path" for stock relationships challenges the algorithm's assumptions, making it less suitable as a standalone tool. Future research could enhance this approach by integrating MST with other financial modeling techniques to address its limitations.

VI. APPENDIX

The complete Python code referenced in this study can be accessed in full at https://github.com/PeterWongsoredjo/MST-for-Portfolio-Diversification. Further explanations of the research are available in the video link at https://youtu.be/89fyPj7ZpNg.

VII. ACKNOWLEDGEMENT

The author would like to express utmost gratitude to Mr. Dr. Ir. Rinaldi Munir, M.T., as the lecturer of IF1220 Discrete Mathematics for his valuable guidance and passion to spread knowledge, which in turn contributes greatly to the completion of this paper. The author would also like to thank all his friends and family members for their moral support throughout the process of researching and writing this paper. .

REFERENCES

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf (Accessed 4 January 2025)
[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf (Accessed 4 January 2025)
[3] https://www.britannica.com/topic/Pearsons-correlation-coefficient (Accessed 4 January 2025)
[4] https://www.idx.co.id/id/data-pasar/data-saham/indeks-saham (Accessed 4 January 2025)
[5] https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp (Accessed 5 January 2025)
[6] https://id.investing.com/equities (Accessed 5 January 2025)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 Desember 2024

Peter Wongsoredjo
13523039