Optimizing Stock Portfolio Allocations Through Combinatorial Analysis to Achieve Maximum Sortino Ratio

Elbert Chailes - 13522045
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13522045@std.stei.itb.ac.id

***Abstract*—There has been rapid increasing numbers of investors all over the world. However, there are many new investors that is confused how to allocate optimal portfolios or even experienced investors that lack of tools that help them find the most optimized portfolio. This research paper is made to analyze on using combinatorial algorithms, especially star and bar approaches, to determine the best performance stock portfolio by looking for the highest Sortino ratio that could be achieved by all combinations that is possible to occur.**

***Keywords*—Star and bar, Combinatorial, Stocks, Sortino Ratio**

I. Introduction

Investment, defined as an asset or item acquired for generating income or appreciation, has experienced significant growth in recent years. The compound annual growth rate (CAGR) of investment rose from $3532.16 billion in 2022 to $3827.1 billion in 2023, marking an 8.6% increase. This surge underscores the pivotal role of investment in navigating the complexities of the global economy and maintaining a competitive edge in the market. Diverse investment vehicles, such as currencies, stocks, cryptocurrencies, and gold, offer varied risk and return profiles, catering to different investor preferences.

Particularly in developing countries of Southeast Asia, such as Indonesia, there has been a remarkable increase in the number of new investors. The count soared to 11.96 million in 2022 and continues to rise. The Indonesian Stock Exchange (Bursa Efek Indonesia, BEI) aims to attract 2 million additional investors by 2024. However, newcomers often face challenges in navigating the investment landscape, especially in selecting profitable investment types or stocks while minimizing risk. A notable phenomenon is the high dropout rate among these new investors following initial losses, leading to a reduction in active participation and lower transaction volumes in the IDX (Indonesian Stock Exchange).

To mitigate these risks and losses for new investors, it is essential to optimize their investment portfolios. This optimization aligns with their risk profiles, ensuring controlled and calculated exposure to risks. The goal is to identify the best combination of stocks for purchase, balancing optimal returns with minimal risk. New investors can thus benefit from a structured approach to analyzing stock performance and arranging their portfolios with calculated risk-return trade-offs.

One effective method for achieving this optimization is the combinatorial approach, specifically the star and bar method, which calculates all possible portfolio arrangements. This method focuses on maximizing the Sortino ratio, indicating lower potential loss risks and alignment with the expected returns of new investors. The highest Sortino ratio represents the best combination, signifying an optimal balance between risk and potential gains for new market entrants.

II. Theoretical Framework

A. Combination

In mathematics, a combination is a technique used to determine the number of possible arrangements in a collection of items, where the order of selection is not important. This concept is distinct from permutations, where the order of selection is significant. The study of combinations and permutations falls within the branch of mathematics known as combinatorics.

The formula for combinations can be used to calculate the number of possible arrangements from a set of *n* distinct objects. In this context, each arrangement is unique regardless of the order of objects. Therefore, different orders of the same objects are counted as one. This property is what fundamentally differentiates combinations from permutations. Combinations are typically calculated when the question of order does not affect the outcome of a scenario, focusing instead on the selection of items or elements from a larger set. The formula for combinations could be expressed as follows.

$$C(n,k) = \binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

where :
n = the total number of distinct objects available for selection.
k = the number of elements to be selected from the n objects or the size of of each combination, or the set that is formed from the large set of n objects.

B. Stars and Bars

Stars and Bars is a combinatorial technique used as a visual aid to solve counting problems, particularly those involving combinations with repetitions. This method simplifies problems where quantities are distributed across different categories or bins.

Consider a scenario where you have *r* identical balls to be placed in *n* distinguishable bins. There are two key cases to consider.

(1) Maximum One Ball per Bin - If each bin can hold at most one ball, the number of ways to distribute the balls is *C(n,r)*, where *C* represents the combination formula. This is a straightforward application of combinations without repetition.

(2) No Limit on Balls per Bin - If there are no restrictions on the number of balls per bin (i.e., a bin can contain more than one ball or none at all), the number of ways to distribute the balls changes. It is calculated as *C(n+r-1,r)* or equivalently *C(n+r-1,n-1)*. This is the case where Stars and Bars method is particularly useful.

In the Stars and Bars approach, the problems could be represented as stars (★) and bars (|). Referring to the case above, stars represent the items (balls) which is being distributed and bars represent the divider between bins. The number of stars is equal to the number of items to distribute (*n*), and the number of bars is *r – 1*, which the separator between bins. So, to visualize cases like distributing 8 balls into 3 bins could be visualized as follows:

★ ★ ★ ★ | ★ | ★ ★

Therefore, find the total combinations possible with this arrangement, we could use combination formula which is $\binom{n}{k} = \binom{9}{2} = \frac{9!}{2!*7!} = 36\ ways$. This means there are 36 different ways to distribute the 8 balls across the 3 bins.

C. Compound Annual Growth Rate (CAGR)

The Compound Annual Growth Rate (CAGR) represents the rate of return (*RoR*) necessary for an investment to grow from its initial balance to its final balance, presuming that the profits are reinvested at the conclusion of each period throughout the investment's duration. It is one of the most precise methods to calculate and assess returns for assets that can increase or decrease in value over time. Essentially, CAGR provides a smoothed annual rate of return, mitigating the effects of volatility of periodic returns.

To get data of return of an investment each year, we could just recap the 5 years historically upside and downside percentage of the specific investment. For example, in the following figure, the green candle represents the upside percentage, and the white candle represents downside percentage.

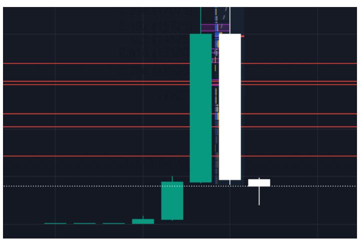

Figure 1. ARTO price candle 1 year timeframe (source: TradingView.com)

The calculation of CAGR can be performed using the following formula:

$$\text{CAGR} = \left(\prod_{i=1}^{n}(1 + r_i)\right)^{\frac{1}{n}} - 1$$

Where:

- r_i is the return for each year *i*, expressed as decimal.
- n is the number of years.

By counting CAGR, useful information is retrieved where it can tell representational figure that describes the rate at which an investment would have grown if it had grown at the various of rate every year and the profits were reinvested at the end of each year. This sort of performance is unlikely to occur because stock performance is fluctuating and volatile, which is impossible to predict. However, with this CAGR result, we could smooth returns so that the information could be understood by investors and could be used in other financial calculations methods.

D. Downside Deviation (DD)

Downside Deviation is a risk metric that zeroes in on returns falling below a minimum threshold, known as the minimum acceptable return (MAR). This measure is integral to the computation of the Sortino Ratio, which evaluates risk-adjusted returns. Unlike standard deviation, which considers all variability, Downside Deviation specifically quantifies the extent of potential loss in an investment, offering a clearer perspective on downside risk. However, it is important to note that Downside Deviation does not account for any upside potential. Consequently, it presents a one-sided view, reflecting only the negative fluctuations and omitting the historical performance of gains.

To get the downside rate, we could look at the downside percentage on the investment price in the *n* periods of time. By looking only at the downside potential, or only the white candle referring to the figures below.

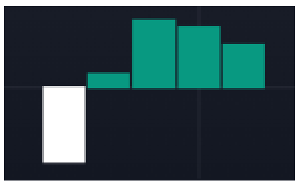

Figure 2. EMTK price candle annually (source: TradingView.com)

Therefore, suppose that the downside percentage of the white candle in the N-(N-1) year is -10%, then the representation as a list of data that count downside percentage is [-10,0,0,0,0]. Then this data could be proceeded to calculate the downside deviation.

To calculate downside deviation, we could follow the steps as follows.

(1) Calculate the mean (average) downside rate over a historical period of *N* years with following formula.

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

(2) Calculate the squared deviations of each downside rate from the mean and sum up the squared deviations and calculate the variance.

$$Variance = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

(3) Finally, calculate the Downside Deviation.

$$Downside\ Deviation = \sqrt{Variance}$$

Downside Deviation provides a more focused assessment of the potential losses an investment might incur. Utilizing this metric allows investors to gauge the level of "risk" associated with an investment, particularly in terms of its downside potential.

E. Sortino Ratio

The Sortino Ratio is a modification of the Sharpe Ratio, designed to distinguish detrimental volatility from overall volatility. It does this by employing the downside deviation—which is the standard deviation of negative asset returns—rather than the total standard deviation of portfolio returns used in the Sharpe Ratio. The Sortino Ratio calculates the excess return per unit of downside risk by subtracting the risk-free rate from the asset or portfolio's return and then dividing the result by the asset's downside deviation.

$$Sortino\ Ratio = \frac{R_p - r_f}{\sigma_d}$$

Where :

R_p is actual or expected portfolio return.

r_f is risk-free rate (usually adapted to local government interest rate).

σ_d is standard deviation of the downside.

Just like a Sharpe Ratio, that higher Sortino Ratio result is better. Investor usually would prefer the one with the higher Sortino Ratio because it means that the investment is earning more return per unit of the bad risk that it takes on,

III. METHODOLOGY

A. Limitations

In this research, several constraints have been established by the researcher to enable the creation of an algorithm for optimizing portfolios based on their Sortino Ratio and respective weightings. These limitations are outlined as follows.

1. The risk associated with each stock is calculated using the downside deviation rate, which considers only the annual downside percentage over a historical five-year period for each stock.
2. The expected return is not specified by individual investors, Instead, it is ascertained by calculating the compound annual growth rate (CAGR) for each selected stock over a five-year period, using the annual historical upside percentage.
3. The risk-free rate is set by the researcher at a constant 2.5%, which reflects the average interest rate encountered globally. This rate can be adjusted according to investor preference.
4. Portfolio optimization in this context focuses solely on the trade-off between risk and reward, generally quantified by the Sortino Ratio, and does not account for other non-quantifiable factors such as news, market reactions, or additional qualitative variables.
5. The algorithm is designed to determine the most advantageous combination for the investor's total investment value, mandating the full allocation of the investor's capital. If the input data includes fewer desirable stocks, the algorithm will identify the best combination with the highest Sortino Ratio from this subset without providing further recommendations. It is important for investors to note that financial decisions should not rely solely on the algorithm's output or the Sortino Ratio. A comprehensive evaluation of potential investments should be considered.
6. The data used is only data sample from Indonesian Stock Exchange because the limitation of researcher finding API that gives data of Indonesian stocks that included annual return for free. Therefore, researcher is forced to transfer data hardly by reading graph from various sources that is contributed to Indonesian Stock Exchange market.

B. Data Sample

The data utilized in this study were sourced from TradingView.com, a platform that provides historical and current pricing information for stocks. For the purpose of this research, the data were specifically selected from Indonesian Stocks listed on the Indonesia Stock Exchange (IDX). The researcher compiled data on the latest stock prices, as well as the annual downside and upside percentages for the past five years for each stock in the sample, as of December 6th, 2023. The collected data samples are presented in the table below.

CODE	LAST PRICE (Rupiah)	RETURN (percentage %)				
		2019	2020	2021	2022	2023
BBCA	8800	28.56	1.27	7.83	17.12	2.63
BMRI	5800	4.07	-17.59	11.07	41.28	15.37
BBNI	5225	-10.8	-21.34	9.31	36.37	10.57
BBRI	5425	20.22	-5.23	1.03	20.19	10.32
BBTN	1265	-16.54	-18.63	0.29	-19.07	-10.00
AGRO	308	-36.13	422.73	78.74	-77.68	-27.72
BJTM	615	-0.72	-0.73	10.29	-5.33	-13.38
BGTG	78	-19.51	12.12	229.73	-62.72	-9.2
ASII	5725	-15.81	-13.00	-5.39	0	-0.88
BRIS	1740	-37.14	581.82	-20.89	-25.69	28.68

Table 1. Data sample used in this research study

Return percentage that is negative represents that the stock prices are plummeted, and positive percentage represents vice versa.

C. Problem Modeling

Step 1 (Input Stock Selection). The investor inputs the stocks they are interested in investing in for their portfolio. If the required data is not readily available, the investor may need to conduct individual research to obtain annual return data for these stocks, which should be recorded in a table.

Step 2 (Determine Investment Amount). The investor specifies the total amount they wish to invest. This figure will be used by the algorithm to calculate the maximum number of shares that can be purchased.

Step 3 (Calculate Minimum Investment Value). Before finding possible combinations, determine the minimum amount of money to be spent. This is calculated as: (Total Investment Amount) - (Total Investment Amount mod (100 x Price of the Most Expensive Stock Chose by Investor)).

Step 4 (Generate Possible Combinations). Utilizing the star and bar method, the algorithm identifies all possible stock combinations that meet the investment criteria. For instance, if the investor is interested in stocks such as BBCA and BBRI, the algorithm will find all combinations where the total cost of BBCA and BBRI lots does not exceed the available investment funds and the total cost is greater than the minimum investment value. In this phase, the amount of lot that is possible to be invest is represented as the star, and the bar will be the separator between the star which is put in stock BBCA (or BBCA bin) and in stock BBRI (or BBRI bin). The star number is not constant in this case which it changes depends on the maximum and minimum investment value.

The possible combination if the investor invest one million rupiah, is represented in the star and bar representation as follows.

BBRI	**BBCA**
★	-
BBRI	**BBCA**
-	★

Figure 3. Illustration of possible combination of the study case

In the above figure, there are two combinations that are possible to be made from BBRI and BBCA with one million rupiah of investment. If it is represented in mathematical way, then the following equation would best present the case.

$$Minimal\ investment\ value < (Amount\ lot\ of\ BBCA) * (BBCA\ stock\ price) + (Amount\ lot\ of\ BBRI) * (BBRI\ stock\ price) \leq Investment\ Amount$$

The combination of (Amount lot of BBCA, Amount lot of BBRI) will be (0,1) and (1,0) because it is only combinations that fulfill the prerequisites. The case (0,0) would not be considered because it is below 0 rupiah is below the minimal investment value range and case (1,1) is not considered also because it is higher than investment amount which is 1 million rupiah.

Step 5 (Calculate Downside Deviation and CAGR). The algorithm calculates the downside deviation and Compound Annual Growth Rate (CAGR) for each stock. The CAGR represents the expected return and is based on the data provided in the table.

Step 6 (Compute Sortino Ratio for Each Combination). The algorithm calculates the Sortino Ratio for each combination, factoring in the CAGR and downside deviation. This calculation is weighted according to the number of lots purchased for each stock.

Step 7 (Output Optimal Stock Combination). The algorithm outputs the most advantageous combination of stocks for the portfolio. This is the combination that yields the highest weighted Sortino Ratio, indicating an optimal balance of risk and potential return.

D. Program Implementation to Find Optimal Portfolio with Risk and Reward Considerations (Python language programming based)

In the theoretical framework and problem modeling conducted by the researcher, the realization of portfolio optimization is achieved using the Python programming language. The researcher has translated mathematical concepts into functional code, enabling the identification of the most advantageous combinations of stocks for an investor's portfolio.

During the implementation phase, the researcher presents not only the optimal combination of stocks but also enumerates all possible combinations. Additionally, the Sortino ratio for each combination is displayed. This is intended to encourage investors to consider the stock allocation more deeply. Even if a combination is said to be the 'best' by the program, investors are prompted to evaluate whether the Sortino ratio is satisfactory.

In order to implement the function, firstly the researcher transfers the data that the researcher found from various resources into static data setup in the code. To clarify that because researcher is not using any API, the researcher manually writes it into the code editor. The snippet of the static data could be seen in Figure 4.

```python
# Static data setup for each stock
stock_data = {
    # Example: 'STOCK_CODE': {'price': ..., 'DD': ...}
    # DD is last five year downside percentage
    # AR is last five year price movement percentage
    'BBCA': {'price': 8800, 'DD': [0,0,0,0,-0.28], 'AR': [28.56,1.27,7.83,17.12,2.63]},
    'BMRI': {'price': 5800, 'DD': [0,-17.59,0,0,0], 'AR': [4.07,-17.59,11.07,41.28,15.37]},
    'BBNI': {'price': 5225, 'DD': [-10.8,-21.34,0,0,0], 'AR': [-10.8,-21.34,9.31,36.37,10.57]},
    'BBRI': {'price': 5425, 'DD': [0,-5.23,0,0,0], 'AR': [20.22,-5.23,1.03,20.19,10.32]},
    'BBTN': {'price': 1265, 'DD': [-28.85,-16.54,-18.63,-19.07,-9.26], 'AR': [-16.54,-18.63,0.29,-19.07,-10.00]},
    'AGRO': {'price': 308, 'DD': [-36.13,0,0,-77.68,-26.24], 'AR': [-36.13,422.73,78.74,-77.68,-27.72]},
    'BJTM': {'price': 615, 'DD': [-0.72,-0.73,0,-5.33,-13.38], 'AR': [-0.72,-0.73,10.29,-5.33,-13.38]},
    'BGTG': {'price': 78, 'DD': [-19.51,0,0,-62.72,-9.2], 'AR': [-19.51,12.12,229.73,-62.72,-9.2]},
    'ASII': {'price': 5725, 'DD': [-15.81,-13.00,-5.39,0,-0.44], 'AR': [-15.81,-13.00,-5.39,0,-0.88]},
    'BRIS': {'price': 1740, 'DD': [-37.14,0,-20.89,-25.69,0], 'AR': [-37.14,581.82,-20.89,-25.69,28.68]},
}
```

Figure 4. Static data written in code editor snippet

To implement the Compound Annual Growth Rate (CGAR), as outlined in the theoretical framework, the researcher has translated it into the Python code, as illustrated in Figure 5.

```python
def calculate_annualized_return(annual_returns):
    """
    Calculate the annualized return given a list of annual returns.

    Parameters:
    annual_returns (list): List of annual returns (expressed as percentages).

    Returns:
    float: The annualized return.
    """
    compounded_return = 1
    for return_percent in annual_returns:
        compounded_return *= (1 + return_percent / 100)

    years = len(annual_returns)

    return ((compounded_return ** (1 / years)) - 1)*100
```

Figure 5. The implementation of CGAR formula into function

Here, the function **calculate_annualized_return** is introduced. It takes a parameter - a list containing data on a stock's return over five years, as explained in the theoretical framework. This function returns a float representing the CGAR percentage, which is then utilized in other functions, particularly in calculating the Sortino Ratio for each stock combination.

Additionally, the researcher has developed a function for calculating the downside deviation of stocks, named **calculate_downside_deviation**. This function (Figure 6) takes **downside_rates** as a parameter, which is a list containing data on the five-year downside percentage of a stock. The following figures show snippets of the code implementation and the integration of the downside deviation function with the CGAR formula in the Sortino ratio calculation (Figure 7).

```python
def calculate_downside_deviation(downside_rates):
    if not downside_rates:
        return 0  # Avoid division by zero if the list is empty

    mean_rate = sum(downside_rates) / len(downside_rates)
    squared_deviations = [(rate - mean_rate) ** 2 for rate in downside_rates]
    variance = sum(squared_deviations) / (len(downside_rates) - 1)
    return variance ** 0.5  # Square root of variance
```

Figure 6. The implementation of DD formula into function

```python
def get_sortino_ratio(expected_return, downside_deviation):
    """Calculates the Sortino Ratio for a given expected return and downside deviation."""
    if downside_deviation == 0:
        return float('inf')  # Prevent division by zero
    return (expected_return-2.5) / downside_deviation # 2.5% is risk free rate
```

Figure 7. The implementation of DD and CGAR formula into Sortino Ratio Calculation function

Moreover, the researcher has created a function to determine the minimum investment value (Figure 8), a crucial factor in generating stock combinations. This is aligned with the "Minimal Investment Value" variable defined in the theoretical framework.

```python
def calculate_min_investment(interested_stocks, stock_data, total_investment):
    """Calculates the minimum investment required to optimize the portfolio."""
    min_investment = float('inf')
    for stock in interested_stocks:
        cost_of_100_shares = stock_data[stock]['price'] * 100
        if cost_of_100_shares < min_investment:
            min_investment = cost_of_100_shares
    return total_investment - (total_investment % min_investment)
```

Figure 8. The implementation of Minimal Investment Value calculation

Finally, the researcher has designed a function to generate stock combinations, each of which will have its Sortino Ratio calculated which could be seen on following snippets, which the function is named **generate_combination** for generating stars and bar combination and **find_best_combination** to find the highest Sortino Ratio combination of stocks. The snippet of the code could be seen in Figure 9.

```python
def generate_combinations(interested_stocks, stock_data, total_investment, min_investment):
    """Generates all possible unique combinations of stock lots within the investment limits."""
    max_lots = {stock: int(total_investment // (stock_data[stock]['price'] * 100)) for stock in interested_stocks}
    all_combinations = []
    unique_combinations = set()

    for combination in itertools.product(*(range(max_lots[stock] + 1) for stock in interested_stocks)):
        total_cost = sum(stock_data[stock]['price'] * 100 * combination[idx] for idx, stock in enumerate(interested_stocks))
        if min_investment <= total_cost <= total_investment:
            formatted_combination = tuple((stock, combination[idx]) for idx, stock in enumerate(interested_stocks))
            if formatted_combination not in unique_combinations:
                unique_combinations.add(formatted_combination)
                all_combinations.append(list(formatted_combination))
    return all_combinations

def find_best_combination(all_combinations, sortino_ratios):
    """Finds the best stock combination based on the highest number of Sortino Ratio."""
    best_combo = None
    best_ratio = -1

    for combo in all_combinations:
        total_lots = sum(lots for _, lots in combo)
        if total_lots == 0:
            continue
        weighted_sortino_ratio = sum(sortino_ratios[stock] * lots for stock, lots in combo) / total_lots
        combo.append(weighted_sortino_ratio)
        ratio = combo[-1]
        if ratio > best_ratio:
            best_ratio = ratio
            best_combo = combo
    return best_combo
```

Figure 9. Code snippet of function that returns the best combination stock portfolio base on Sortino Ratio

IV. Testing and Analysis

A. Testing

The result of the program execution after combining all the functions that could be seen in methodology chapter could be seen in following figures.

```
Enter the stock codes you're interested in (separated by space): BBCA BBRI BGTG
Enter your total investment amount: 10000000
[('BBCA', 0), ('BBRI', 0), ('BGTG', 1282), -0.09019299316781461]
[('BBCA', 1), ('BBRI', 4), ('BGTG', 891), -0.0015787231741492944]
[('BBCA', 2), ('BBRI', 8), ('BGTG', 500), 0.22117322610404466]
[('BBCA', 3), ('BBRI', 1), ('BGTG', 874), 0.14609615455538352]
[('BBCA', 4), ('BBRI', 5), ('BGTG', 483), 0.49285631092830745]
[('BBCA', 6), ('BBRI', 2), ('BGTG', 466), 0.7851735540936536]
[('BBCA', 7), ('BBRI', 6), ('BGTG', 75), 5.527103930880595]
[('BBCA', 9), ('BBRI', 3), ('BGTG', 58), 8.801030079730813]
[('BBCA', 11), ('BBRI', 0), ('BGTG', 41), 14.341520485477334]
Total Combination:  9
Best Combination: [('BBCA', 11), ('BBRI', 0), ('BGTG', 41), 14.341520485477334]
```

Figure 10. Program execution to find the most optimal portfolio by allocating one million rupiah to buy 3 distinct stocks (BBCA, BBRI, BGTG)

```
[('ASII', 6), ('BBTN', 34), ('BRIS', 1), ('BBNI', 4), -1.8221199030453725]
[('ASII', 6), ('BBTN', 34), ('BRIS', 4), ('BBNI', 3), -1.6520063661222648]
[('ASII', 6), ('BBTN', 34), ('BRIS', 7), ('BBNI', 2), -1.4957796485398185]
[('ASII', 6), ('BBTN', 34), ('BRIS', 10), ('BBNI', 1), -1.3518060068461917]
[('ASII', 6), ('BBTN', 34), ('BRIS', 13), ('BBNI', 0), -1.2186983003747258]
[('ASII', 6), ('BBTN', 45), ('BRIS', 2), ('BBNI', 1), -1.9465569999445924]
[('ASII', 6), ('BBTN', 45), ('BRIS', 5), ('BBNI', 0), -1.7993389565662976]
[('ASII', 8), ('BBTN', 7), ('BRIS', 2), ('BBNI', 8), -0.9216989497180843]
[('ASII', 9), ('BBTN', 8), ('BRIS', 1), ('BBNI', 7), -1.125092783848807]
[('ASII', 9), ('BBTN', 8), ('BRIS', 4), ('BBNI', 6), -0.8806008580342537]
[('ASII', 9), ('BBTN', 8), ('BRIS', 7), ('BBNI', 5), -0.6698319564699836]
[('ASII', 9), ('BBTN', 8), ('BRIS', 10), ('BBNI', 4), -0.48625904220432914]
[('ASII', 9), ('BBTN', 8), ('BRIS', 13), ('BBNI', 3), -0.3249373902739054]
[('ASII', 9), ('BBTN', 8), ('BRIS', 16), ('BBNI', 2), -0.18205249856410155]
[('ASII', 9), ('BBTN', 19), ('BRIS', 2), ('BBNI', 4), -1.5072353516525412]
[('ASII', 9), ('BBTN', 19), ('BRIS', 5), ('BBNI', 3), -1.3026362646358631]
[('ASII', 10), ('BBTN', 9), ('BRIS', 0), ('BBNI', 6), -1.3284866179795298]
[('ASII', 10), ('BBTN', 9), ('BRIS', 3), ('BBNI', 5), -1.0689284822293674]
[('ASII', 10), ('BBTN', 9), ('BRIS', 6), ('BBNI', 4), -0.8451714686516412]
[('ASII', 10), ('BBTN', 9), ('BRIS', 9), ('BBNI', 3), -0.6502863277936217]
[('ASII', 10), ('BBTN', 9), ('BRIS', 12), ('BBNI', 2), -0.4790236282517256]
[('ASII', 10), ('BBTN', 9), ('BRIS', 15), ('BBNI', 1), -0.3273338086574749]
[('ASII', 10), ('BBTN', 9), ('BRIS', 18), ('BBNI', 0), -0.19204288847881887]
[('ASII', 10), ('BBTN', 20), ('BRIS', 1), ('BBNI', 3), -1.6567896414545429]
[('ASII', 10), ('BBTN', 20), ('BRIS', 4), ('BBNI', 2), -1.4438819827821983]
[('ASII', 10), ('BBTN', 20), ('BRIS', 7), ('BBNI', 1), -1.253385656601679]
[('ASII', 10), ('BBTN', 20), ('BRIS', 10), ('BBNI', 0), -1.0819389630392122]
[('ASII', 10), ('BBTN', 31), ('BRIS', 2), ('BBNI', 0), -1.847663492312109]
[('ASII', 11), ('BBTN', 10), ('BRIS', 5), ('BBNI', 3), -1.0205109808332986]
[('ASII', 11), ('BBTN', 10), ('BRIS', 8), ('BBNI', 2), -0.8143136133829141]
[('ASII', 11), ('BBTN', 10), ('BRIS', 11), ('BBNI', 1), -0.6331098662295458]
[('ASII', 11), ('BBTN', 10), ('BRIS', 14), ('BBNI', 0), -0.4726151187508482]
[('ASII', 11), ('BBTN', 21), ('BRIS', 0), ('BBNI', 2), -1.806343931256545]
[('ASII', 11), ('BBTN', 21), ('BRIS', 3), ('BBNI', 1), -1.5851277009285336]
[('ASII', 11), ('BBTN', 21), ('BRIS', 6), ('BBNI', 0), -1.387197389582418]
[('ASII', 15), ('BBTN', 7), ('BRIS', 0), ('BBNI', 1), -1.5543976450058792]
[('ASII', 15), ('BBTN', 7), ('BRIS', 3), ('BBNI', 0), -1.2560019762335959]
Total Combination:  199
Best Combination: [('ASII', 0), ('BBTN', 2), ('BRIS', 56), ('BBNI', 0), 1.338884652385915 7]
```

Figure 11. Cropped program execution to find the most

optimal portfolio by allocation ten million rupiah to buy 4 distinct stocks (ASII, BBTN, BRIS, BBNI)

B. Test Explanation

In figure 10, there is an information retrieved which there are nine possible stock combinations that fulfill the prerequisites that has been set in the theoretical framework chapter. It shows that the best combination is [('BBCA', 11), ('BBRI', 0), ('BGTG', 41), 14.341520485477334]. The meaning of those array is the best portfolio could be made is allocating 11 lots to BBCA, 0 lot to BBRI, and 41 lots to BGTG, with the Sortino Ratio 14.34, which means that this combination of portfolio would give the investor 14.34% return per unit risk. But need to note that this is based on last five years price, not prediction to the future. So, this is just going to be one of many supporting analyses made by the investor, not the main analysis. From figure 10, we could also see that there is some combination that has negative Sortino Ratio which means that combination has negative return per unit risk which is not recommended to be part of investor's portfolio.

In figure 11, there is the same execution program as Figure 10, but with different money invested and the stocks that the investor is interested in. In figure 11, the investor wanted to invest ten million rupiah in four distinct stocks, which is ASII, BBTN, BRIS, and BBNI. It produces 199 combinations that fulfilled the prerequisite and the best combination is [('ASII', 0), ('BBTN', 2), ('BRIS', 56), ('BBNI', 0), 1.3388846523859157], which means the best combination of portfolio is allocating 0 lot to ASII, 2 lots to BBTN, 56 lots to BRIS, and 0 lots to BBNI, with the highest Sortino Ratio which is 1.34.

From the two test cases provided above, we could conclude that the program algorithm works well and gives the expected Sortino Ratio for positive return stocks for last five years and give negative Sortino Ratio for bad performance stocks. But, as we can see in the second test, the number of Sortino Ratio is 1.34 which is lower than the Sortino Ratio of first test. We could say it does mean that the first portfolio would have better performance than second test portfolio, but it still depends on the investor analysis on making financial decisions. Which kind of portfolio at last still depends on the profile risk and trading style of the investor that uses this program. This program would just help investors as one of their analyses to strengthen their decision-making.

C. Algorithm and Method Analysis

Based on the algorithm that is made by the researcher, we could see that the algorithm would have an enormous time complexity since the algorithm is brute forcing all possibilities using itertools, The researcher also found out that the program would take a really long execution time when meeting investor that wanted to invest more than 100 million rupiah which will increase the possibility of portfolio combinations exponentially. The heaviest function to be executed is *generate_combination* function which would take $O(m*p^m)$ time complexity with m is the number of interested stocks and p is the number possible for lots of one stock.

There may be other way to find the most optimized portfolio from a bunch of stock chose by investor, but the researcher felt this is a good way to find the most optimal portfolio because this algorithm literally do seek for every combinations possible so there is no combinations that is missed, so the investor that use this algorithm still got the best combination even though with ineffective program time execution.

This kind of method may be just helping the investor to strengthen their analysis not as their main factor to make trade-off decisions. This implementation is more fit to the investor that accounts more to the risk then the volatility of profit. It is recommended that if the investor is interested on taking volatility into account, then Sharpe Ratio would be better implemented onto this program.

V. CONCLUSION

The program accurately generates all feasible combinations and precisely calculates the Sortino Ratio for each, aligning with the researcher's objectives for this study. The algorithm is a valuable tool for investors, aiding in the identification of the best portfolio combinations using the star and bar method.

However, it should be noted that the algorithm may not perform efficiently with large-scale data sets, such as substantial investment amounts. This is due to the exponential increase in execution time for larger data, which could potentially cause the program to halt. Despite this limitation, the researcher believes that this algorithm represents an optimal approach for determining the best performance portfolio based on the Sortino Ratio. By employing a brute-force method, complemented by the star and bar technique for combination analysis, the algorithm ensures that no potential investment combination is missed.

VI. APPENDIX

The complete portfolio optimalization algorithm could be found below.
https://github.com/ChaiGans/StockPortfolioOptimilization-StarAndBar-SortinoRatio

VII. ACKNOWLEDGMENT

First and foremost, researcher would like to thank my discrete math lecturer which is known as Dr. Nur Ulfa Maulidevi, S.T., M.Sc. which has shared its knowledge to her fellow students, which researcher is one of them. Without my discrete math lecturer, the researcher is skeptical that he could finish this research smoothly.

Researcher would also thank to other lecturers that contributed and be the part of IF2120 Discrete Math class because their hard work is the only reason, the researcher writing this research paper.

Lastly, the researcher would like to thank his fellow friends that has been supporting him on finishing the Discrete Math class and give him chance to learn and develop together as the time passes.

REFERENCES

[1] R. Munir, "Homepage Rinaldi Munir," [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/matdis23-24.htm. [Accessed 8 December 2023].

[2] W. Kenton, "Sortino Ratio: Definition, Formula, Calculation, and Example," 9 August 2020. [Online]. Available: https://www.investopedia.com/terms/s/sortinoratio.asp. [Accessed 8 December 2023].

[3] J. Fernando, "Compound Annual Growth Rate (CAGR) Formula and Calculation," 24 May 2023. [Online]. Available: https://www.investopedia.com/terms/c/cagr.asp. [Accessed 8 December 2023].

[4] M. Jahn, "Downside Deviation: Definition, Uses, Calculation Example," Investopedia, 24 July 2022. [Online]. Available: https://www.investopedia.com/terms/d/downside-deviation.asp. [Accessed 8 December 2023].

[5] J. Riordan, "Permutations and Combinations," in *Introduction to Combinatorial Analysis*, New York, John Wiley & Sons, Inc., 2023, p. 244.

[6] T. N. Rollinger, "Sortino: A 'Sharper' Ratio," [Online]. Available: http://www.redrockcapital.com/Sortino__A__Sharper__Ratio_Red_Rock_Capital.pdf. [Accessed 8 December 2023].

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, 8 Desember 2023

Elbert Chailes (13522045)