Utrecht University

Department of Information and Computing Science

Artificial Intelligence Master Thesis

Cryptocurrency portfolio optimization through Gene Pool Optimal Mixing Evolutionary Algorithms

First examiner:

Dr. Dirk Thierens

Second examiner:

Dr. Slinger Jansen

Candidate:

Arturo Garcia

May 9, 2024

Abstract

Portfolio optimization of cryptocurrencies with Evolutionary Algorithms is a fairly new topic in financial literature. New and upcoming studies are addressing portfolio optimization problems through a wide array of evolutionary and other novel algorithmic approaches. This study compares the Gene-Pool Optimal Mixing Evolutionary Algorithm (GOMEA) with the Genetic Algorithm and the Particle Swarm Optimization through an evaluation of each algorithm's capabilities for portfolio risk management. Specifically, we use the Conditional Value at Risk (CVaR) as our risk metric for optimization and construct an efficient frontier for the portfolios generated to examine the performance of the algorithms. Making use of both simulated and historical data, our analysis focuses on these algorithms' capacity to manage the intricate risk/reward trade-off inherent in cryptocurrencies. We construct a theoretical framework that supports the assumption behind the preference of GOMEA and conduct an empirical analysis to test whether our assumptions hold under the two distinctive datasets. Our results suggest that GOMEA presents an overall better performance in portfolio risk management through its optimization approach of the cryptocurrency portfolios. These results underscore the potential benefits of employing advanced evolutionary algorithms that exploit the inherent interdependencies found in cryptocurrencies.

Keywords: *Evolutionary Algorithm, Conditional Value at Risk, Portfolio Optimization, Gene-Pool Optimal Mixing Evolutionary Algorithm, Cryptocurrency*

Contents

1. Introduction

1.1 Background

Innovation has been a constant throughout human history; our species has continuously expanded its reach through new creations and discoveries. That curiosity to explore and boldness to create is what defines us. In the realm of financial markets, various products have shaken the status quo. For instance, the establishment of early stock markets, such as the Amsterdam Stock Exchange in 1602, significantly impacted economies across Europe, opening the doors to a new economic era. Similarly, the introduction of derivative assets revolutionized modern financial markets, offering opportunities and challenges. This innovation trend continued with the emergence of cryptocurrencies, marking the beginning of a new age – an age of discovery and speculation, but mainly, an age of opportunity. The most notable cryptocurrency, Bitcoin, was introduced in a white paper by the enigmatic Satoshi Nakamoto [1], whose identity remains a mystery. As of 2023, digital assets such as cryptocurrencies have become commonplace in the portfolios of financial managers and investors despite their inherent risk and volatility. We are still constructing our understanding of these assets as we uncover their characteristics, behaviors, and patterns. However, two clear intrinsic aspects are their high volatility and risk compared to traditional financial instruments. This realization has led several studies to attempt to mitigate these inherent characteristics. This paper focuses on an innovative approach to portfolio risk management through the optimization of a cryptocurrency portfolio by applying the Gene-Pool Optimal Mixing Evolutionary Algorithm (GOMEA). We aim to optimize the weights of a portfolio comprising five cryptocurrencies.

1.1.1 Evolution of portfolio optimization

The financial strategy behind portfolio optimization consists of the practice of balancing the maximization of returns while minimizing risk, subject to specific investment goals and constraints. Portfolio managers and investors have devised different approaches to exploit this trade-off over time. Portfolio optimization has undergone some significant evolution over time. The first optimization approach to be academically recorded was provided by Markowitz, in which he presented the mean-variance model [2]. Such a model sought to identify the portfolio that provided the maximum expected return (the mean return) for a predetermined level of risk (variance). Though considered limited by today's standard models, this method was revolutionary at its time, given the innovative approach taken through a new scientific approach in portfolio management through its quantification of the concept of diversification and formalization of it as a mathematical model. Before Markowitz's approach, investment strategies mostly conformed to qualitative analysis, rule-of-thumb practices, and loosely formalized methods. From that point onwards, the industry underwent a paradigm shift with the surge of innovative optimization techniques. Building on the foundational mean-variance model, Samuelson introduced variance with skewness, enriching the model by incorporating the skewness of the return distribution [3]. The progression continued with Konno's introduction of the mean absolute deviation method, which employed mean absolute deviation as a measure of risk instead of variance to quantify risk [4]. Seven years later, the Minimax approach emerged, which is an approach focused on minimizing the maximum possible loss [5]. Transferring the point of focus to the potential losses that could be caused by extreme market volatility.

The mentioned traditional approaches, while innovative on their own merits, still carried with them a relevant oversight as they relied heavily on the assumption that the returns followed a normal distribution. This, as it is known, is not the case, particularly during times of extreme market events. Most of the classical approaches mentioned are based on assets and cardinality constraints. Nonetheless, two particular statistical measures were developed to address these oversights. Jorian presented the Value at risk (VaR), which is a value that estimates the maximum expected loss over a given time window at a specific confidence level [6]. This measure, while helpful, still fails to capture the tail risk as VaR fails to capture the size of the loss beyond the identified threshold value. This limitation was addressed through the introduction of the Conditional Value at Risk (CVaR), also known as Expected Shortfall (ES), which is an extended version of the VaR [7]. The CVaR estimates the expected average loss beyond the VaR threshold, offering a more tangible measure of potential extreme losses in worst-case scenarios. Both the VaR and the CVaR have intrinsic characteristics that allow them to be applied to non-normal and asymmetric distributions of returns, which are distributions that are closer to those that the returns tend to follow.

The classical portfolio optimization approaches were enhanced and transformed by the disruption of advanced algorithms in the fields of machine learning, probabilistic theory, and quantum computing, among many others. Within the realm of machine learning, we have seen the development of innovative financial portfolio management tools that leverage deep graph convolutional reinforcement learning to take advantage of financial interrelations [8], Neural network models [9], Reinforcement Learning [10, 11], and evolutionary algorithms [12, 13]. Regarding the probability theory field, we can find some studies made through Bayesian approaches. Such approaches use semi-definite relaxation in portfolio selection [14], minimax as a risk measure [15], and portfolio selection under cardinality constraints [16]. Finally, we also find studies in the field of quantum computing that use quantum combinatorial optimization [17] and a Variational Quantum Eigensolver [18].

1.1.2 Emergence of Cryptocurrencies as an asset class

As of 2023, nobody can argue about the disruptive relevance that cryptocurrency and digital assets have had in the current financial systems. Crypto assets have evolved from being suspicious and frowned upon financial instruments to becoming valid alternatives for investment and diversification. In particular, given their seemingly disconnected relation to other mainstream assets [19], they represent a viable alternative asset for portfolio managers to consider in their portfolio construction process for short-term horizons [20].

Furthermore, as this asset class becomes more prominent, we are starting to see the intention to take regulatory oversight from regulatory entities like the SEC in the United States and the ECB, through their MICAR regulation, which signals a potential adoption of these new instruments in regular financial systems. Nonetheless, the endogenous characteristics of crypto-assets are still being studied and analyzed. While there is no clear consensus on how we should treat crypto-assets, most investors and portfolio managers consider them to be speculative assets due to their volatile nature [21]. The mentioned volatility highlights the limitations of traditional portfolio optimization methods, which were primarily designed for more conventional financial instruments. Consequently, this discrepancy highlights the need for dynamic and adaptable approaches, such as evolutionary algorithms, which are equipped to handle the complexities and unpredictability inherent in crypto-asset portfolios.

1.2 Evolutionary Algorithms in Portfolio Optimization

For contextual purposes, we first need to understand that within the realm of artificial intelligence, we have a particular branch called evolutionary computing, which is a field based on biological evolution. Inside this field of evolutionary computing, we can find the evolutionary algorithms. Evolutionary algorithms (EAs) are often described as bio-inspired heuristic optimization procedures that mimic behaviors and mechanisms we find in nature. While there is some level of disagreement on what exactly should be considered an evolutionary algorithm, the broad consensus is that it is a computational representation of a process found in nature, not necessarily an optimization one [22]. Our study will focus on evolutionary optimization algorithms, given their stochastic optimization techniques that allow us to navigate the search space through their population-based search strategies [12]. In particular, we will use the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA), which is a model-based Evolutionary Algorithm.

GOMEA's distinctive characteristic is that in each generation, it identifies and learns the linkage patterns among the variables through a probabilistic model. Through the process of gene-pool optimal mixing, the GOMEA algorithm selectively combines the best set of solutions available in the solution pool by focusing on combining sub-solutions without breaking beneficial linkages. This approach allows the algorithm to capture and identify crucial relationships between the variables [23].

The GOMEA algorithm is well suited for addressing optimization problems, which is why we consider it an optimal candidate for addressing the CVaR portfolio optimization problem. Particularly given that GOMEA is known for handling complex landscapes [23], including characteristics such as non-linearity and non-convexity, two attributes potentially inherent in the CVaR problem structure of a portfolio of cryptocurrencies. Additionally, given that we will be working with a portfolio of crypto-assets, exploiting the underlying relationships between these assets is critical, given their known interdependency. GOMEA's ability to identify and leverage these relationships when iterating over their partial solutions makes it a good vehicle for addressing this optimization problem.

It is relevant to note that GOMEA is one of the many evolutionary algorithms that can be used for this particular optimization problem. Some examples of the use of evolutionary algorithms in the CVaR optimization problem are found in Setiwan's work in which he deploys genetic algorithms, grasshopper optimization, firefly optimization, moth flame optimization, particle swarm optimization, grey wolf optimization, and dragonfly optimization [24]. We can also find some evolutionary approaches in the work of Clement et al., in which they analyze a cryptocurrency portfolio using a Particle Swarm Optimization Copula-based approach [12].

1.3 Main contributions of this study

This paper aims to evaluate the effectiveness of the Black-Box GOMEA, as presented in Bouter and Bosman's study [25], in optimizing cryptocurrency portfolio weights by comparing its performance with various known evolutionary algorithms used in portfolio optimization. Our research seeks to ascertain whether GOMEA outperforms these algorithms in the context of portfolio risk management through cryptocurrency portfolio optimization.

This study is significant as it contributes to the growing literature about evolutionary algorithm deployment in financial optimization problems. While the optimization task has been one of the main focuses of evolutionary algorithms, there is a lack of studies that focus on the use of cryptocurrencies as financial elements of a portfolio, given their volatile and risky nature. The GOMEA, along with the methodology implemented in this paper, provides a potential vehicle for the investigation of novel approaches to portfolio risk management of cryptocurrencies.

The paper is structured as follows: Chapter 2 investigates the unique characteristics of cryptocurrencies as financial assets. Chapter 3 analyzes portfolio optimization techniques, including a tailored approach for cryptocurrency scenarios. Chapter 4 explores Evolutionary Algorithms (EAs), particularly the GOMEA's customization for the CVaR optimization problem. Chapter 5 details our study's methodology, Chapter 6 presents our findings, Chapter 7 introduces the discussion, and Chapter 8 presents our conclusion.

2. Cryptocurrencies

2.1 Decrypting Cryptocurrencies

For this study, we will use cryptocurrencies as our portfolios' financial assets. In a systematic analysis of cryptocurrencies as an asset, Corbet defines them as peer-to-peer electronic cash systems that allow direct transit of online payments between parties without the need for a financial institution as an intermediary [19]. This implies that no third party acts as a regulator or mandatory element of any transaction. Cryptocurrencies are primarily anonymous as there is no need for the parties to provide any identification aside from the address of their corresponding wallets, making them an attractive tool for maintaining privacy [26].

The underlying technology behind cryptocurrencies is known as Distributed Ledger Technology (DLT), defined as a decentralized database managed by the nodes that make up its network. This technology operates on principles of decentralized storage networks and architectures that promote transparency, incentivizing nodes (members of the network) to share storage space through the use of native tokens and transforming cloud storage into algorithmic markets [27]. Among various types of Distributed Ledger Technologies (DLTs), including Directed Acyclic Graph (DAG), Hashgraphs, Tangle, and Holochain, blockchain is the most commonly adopted type in cryptocurrencies and the one employed by the assets examined in this study. Blockchain is a type of DLT in which data is structured into blocks that are cryptographically linked and distributed across a network of independent nodes. Whenever an individual makes a transaction with a particular cryptocurrency, it is recorded in that asset's blockchain. The blockchain network is composed of a chain of different blocks of records that keep track of all the transactions that have ever taken place within the network.

As with any complex societal and economic system in which multiple stakeholders possess decision-making roles and authority, clear governance is essential to ensure the enforcement of the system's objectives. Cryptocurrencies and their underlying blockchain technologies are no exception. In a study in which they introduce a blockchain governance framework, Slinger et al. explain that blockchain governance often mirrors strategies observed in traditional Open-Source Software (OSS) projects, sharing aspects such as the development and release processes, involvement of external parties, political debates over data (de)centralization, reliance on contributions from users and developers, motivational factors (incentives), and a layered approach to governance. Moreover, they explain that blockchain governance acts as the means to provide guidance, management, and collaboration among stakeholders within a blockchain environment [28].

To maintain the integrity, security, and decentralized nature of the network, cryptocurrencies use consensus mechanisms as a core governance tool. These mechanisms dictate how transactions are verified and how new blocks are created and added to the blockchain. Depending on the type of consensus mechanisms used, validators can be considered as miners (for Proof of Work) or stakers (in Proof of Stake systems). The most common consensus mechanisms are:

Proof of work (PoW) consensus mechanism requires miners to find the solution to complex mathematical puzzles. Once the puzzle is solved, the miner can validate a transaction and add a new block to the network. This is rewarded with a token of the network (in the case of the Bitcoin network, a miner would receive a Bitcoin token for solving the puzzle).

Proof of Stake (PoS) is a consensus mechanism in which validators are selected for new block creations based on the number of coins they hold in the network. This means that the higher the number of coins they have, the higher the likelihood that they will be selected as validators. An important point to mention is that selected validators need to commit some coins as collateral for the validation process (this action is known as staking).

Proof of Authority (PoA) It is a more centralized approach to consensus mechanism. The validation process is entirely managed by selected validators who are pre-selected by the network founders. The selected validators are expected to act responsibly and always to benefit the network's health.

One of the most debated questions in financial literature is how cryptocurrencies derive value [29]. From a technical perspective, a cryptocurrency is considered as valuable as the combination of cryptographic algorithms, consensus rules, and network protocols (the security of the algorithms that keep track of the transactions) [30]. There are several ways to obtain cryptocurrencies. First, you can contribute to the network of a cryptocurrency project and get paid in their token/coin. This is called being a miner or validator. Another common way is to buy cryptocurrencies from online exchanges. You can also use peer-to-peer (P2P) platforms to trade cryptocurrencies directly with other people. Also, if you have a digital wallet that works with the cryptocurrency you're interested in, you can send or receive it. Finally, you can buy cryptocurrencies early in their launch through something called Initial Coin Offerings (ICOs) or at later stages in their project life through token sales. Each of these alternatives caters to the level of expertise and exposure that the investor has to the cryptocurrency world.

In an interesting study on the statistical characteristics of the top seven cryptocurrencies (according to market capitalization), Chan et al. found some particular insights [31]. They found that while none of the returns of the cryptocurrencies followed a normal distribution, there was no single distribution able to capture the behavior of all returns for all the cryptocurrencies. Nonetheless, in some encouraging findings, they were able to fit the returns of Bitcoin and Litecoin to the generalized hyperbolic distribution (GHD). This distribution's primary attribute is its flexibility in modeling heavy tails and skewness, which are predominantly seen in cryptocurrencies.

2.2 Cryptocurrencies vs Traditional Financial Assets

Cryptocurrencies behave differently than traditional financial assets. Corbet et al. found that cryptocurrencies are very connected to each other but isolated from other traditional financial assets [19]. Moreover, aside from the mentioned detachment, they present evidence that cryptocurrencies are unaffected by market shocks from traditional assets. Such a finding is reinforced by the findings of Baur et al., who recognize in their study that cryptocurrencies such as bitcoin present no correlation with traditional assets during periods of financial turmoil [20]. This dissociation from the traditional financial landscape offers an alluring opportunity for investors interested in hedging or diversifying their investments[32, 33, 34].

Cryptocurrencies are considered speculative assets rather than conventional vehicles of traditional investment. Certain studies found that crypto-assets' characteristics make them prone to being used as vehicles for speculation. Characteristics such as their high growth potential, highly liquid market, volatile return, unregulated nature, and short-term sight invite investors to see this asset as a pump-and-dump scheme opportunity [35]. Baek's study suggests that Bitcoin can be considered a speculative asset that is guided by the exchange between buyers and sellers and not by fundamental economic factors [36].

We also see a distinctive reaction to market shocks in the case of cryptocurrencies. In terms of liquidity, cryptocurrency markets seem to be heavily influenced by uninformed investors during positive market movements and by informed investors after negative shocks. As Baur et al. conclude in their study, the behavior of uninformed noise traders seems to be driven by a 'fear of missing out' (FOMO) [37]. They explain that both reactions, to either positive or negative shocks, suggest a distinct investor psychology, one different from that observed with traditional assets. The main driver appears to be of a speculative nature rather than analysis and price discovery of the asset's fundamental value. We also observe different behavior in cryptocurrencies regarding co-explosivity among its asset class. Co-explosivity is defined as a moment when different assets experience sudden, simultaneous, and violent increases or decreases in price, driven more by speculative than fundamental factors. In a study about the price explosivity of seven of the largest cryptocurrencies, Bouri et al. found evidence that the cryptocurrencies studied present co-explosivity behavior, which means that regardless of the size of the cryptocurrency, explosivity in one cryptocurrency can lead to explosivity in other cryptocurrencies [38]. During global crises and sector-specific bubbles, it's not unusual for traditional assets to experience simultaneous co-explosivity behavior. However, this phenomenon occurs more frequently in cryptocurrency markets and with greater intensity and speed.

2.3 The Risk and Volatility Characteristics of Cryptocurrencies

The challenge of accurately capturing volatility in cryptocurrencies is notably different from that in traditional financial markets. Traditional volatility models are built around the behaviors of conventional financial assets, which explains why they often fall short in addressing the unique volatility seen in cryptocurrencies. Unlike traditional assets, cryptocurrencies exhibit distinctive features such as extreme volatility, non-stationarity, volatility clustering, and frequent structural breaks in their volatility patterns [39, 40]. These assets are also heavily influenced by factors like investor sentiment, speculative trading, and regulatory changes [41, 42]. Traditional models, which do not account for these specific characteristics, struggle to model cryptocurrency volatility accurately. However, some adaptations of GARCH models, known for their flexibility, have been able to capture the volatility of assets such as Bitcoin [43, 44, 45]. In a study of seven of the most common cryptocurrencies in which they use twelve different types of univariate GARCH models, Chu found that Integrated Generalized Autoregressive Conditional Heteroskedasticity (IGARCH) and Glosten-Jagannathan-Runkle Generalized Autoregressive Conditional Heteroskedasticity (GJR-

GARCH) provide the best fit for modeling volatility in the studied cryptocurrencies [46]. In particular, the IGARCH seems to be a good fit, considering it is known for having a conditional volatility process that is labeled as highly persistent with an infinite memory, which allows the model to capture persistence in volatility and long-term dependencies [47, 48]. The formulation for the IGARCH and GJR GARCH are as follows:

An IGARCH $(1,1)$ model can be written as

$$a_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \alpha_0 + \beta_1 \sigma_{t-1}^2 + (1 - \beta_1) a_{t-1}^2. \tag{2.1}$$

to which a_t is the asset return at time t, σ_t is the conditional standard deviation of a_t, and ϵ_t is the innovation (white noise) term at time t. We also define σ_t^2 as the conditional variance at time t, α_0 as the constant term, β_1 as the autoregressive parameter for past variance, and $(1 - \beta_1)a_{t-1}^2$ as the expression that adjusts the influence of the past squared shocks based on the autoregressive volatility component.

An GJR-GARCH $(1,1)$ model can be written as

$$a_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \gamma_1 a_{t-1}^2 I_{t-1} + \beta_1 \sigma_{t-1}^2. \tag{2.2}$$

where ϵ_t is a white noise process, α_1 is the coefficient for the squared residual from the previous time period a_{t-1}^2. γ_1 is the asymmetric impact coefficient, I_{t-1} is the shock indicator variable that equals 1 if a_{t-1} was negative and 0 if a_{t-1} was positive, and σ_{t-1}^2 is the conditional variance from the previous period. Additionally, $\alpha_0 > 0$, $\alpha_1 \geq 0$, $\gamma_1 \geq 0$, and $\beta_1 \geq 0$.

In a different study regarding the volatility of 292 cryptocurrencies using TGARCH and EGARCH models, Panagiotidis et al. found the existence of an inverse leverage effect in the great majority of the cryptocurrencies analyzed [39]. An inverse leverage effect means that past returns, when positive, tend to have a greater influence on the volatility of cryptocurrencies than negative past returns. This implies that as a cryptocurrency's volatility increases, its price also tends to increase. We refer to this as an 'inverse effect' because, in traditional financial assets, an increase in volatility is typically observed when prices fall. Baur et al. further explains this positive asymmetric behavior in volatility, suggesting that it is likely caused by uninformed investors' herding and FOMO [37]. They also suggest that the response to negative shocks can potentially be explained by informed investors deliberately going against predominant market trends in an opportunistic attempt to profit from the uninformed investors' herding behavior. A formulation of the EGARCH model can be seen in equation 2.4 and for the TGARCH we can find formulate it as follows:

$$\sigma_t^2 = \omega + \alpha \cdot (\varepsilon_{t-1}^-)^2 + \gamma \cdot (\varepsilon_{t-1}^+)^2 + \beta \cdot \sigma_{t-1}^2 \tag{2.3}$$

In which ω is the constant term, α is the Coefficient for past squared negative shocks $(\varepsilon_{t-1}^-)^2$, γ is the coefficient for past squared positive shocks $(\varepsilon_{t-1}^+)^2$, β is the Autoregressive parameter for past variance, ε_{t-1}^- and ε_{t-1}^+ are the negative and positive components of the lagged error term respectively, σ_t^2 is the conditional variance at time t, r_t is the return at time t, μ is the mean return of the series, and z_t is the innovation term at time t.

2.4 Asset selection for this study

For the purposes of this study, we select the following cryptocurrencies: Bitcoin, Ethereum, Litecoin, Ripple, and Monero. This decision is based on multiple criteria that these assets collectively fulfill, primarily their market dynamics, technological diversity, liquidity, and data availability.

Both Bitcoin and Ethereum have the largest market capitalizations among all cryptocurrencies, making them pivotal elements of our portfolio due to their significant relevance. These assets are not only the most traded, ensuring high liquidity, but also provide extensive data records, which are crucial for analysis. Litecoin introduces a different market dynamic with its Scrypt hashing mechanism, which leads to a distinct volatility profile and market response to disruptive events. This unique profile will test the adaptability and efficiency of our evolutionary algorithms under varied technological constraints. Ripple, with its integration into traditional banking systems, works as a anchoring point between decentralized cryptocurrencies and conventional financial systems providing a unique challenge to our optimization algorithms as it provides a mild link towards conventional financial systems [49]. Monero's strong focus on transaction privacy results in particular price behaviours, as investors operate with limited transactional information [50]. This opacity adds an additional layer of complexity to our optimization algorithms, which must adapt to effectively interpret price movements that reflect these constrained investor behaviours.

Together, these cryptocurrencies not only offer diverse consensus mechanisms but also ensure that the portfolio covers a wide spectrum of technological, economic and volatility behaviours. This strategic selection is designed to challenge our optimization algorithms with a realistic array of asset profiles in the portfolio optimization process. In our next chapter, we will examine in detail what portfolio optimization means and the challenges that portfolio managers encounter when faced with the daunting task of portfolio risk management.

3. Investment Portfolios

3.1 Deep dive into Portfolios

3.1.1 Definitions

We begin this chapter by defining what a portfolio is. Gunjan defines a portfolio as a collection of assets and/or investments [51]. A portfolio can comprise different types of assets such as equities, bonds, cash equivalent, real estate properties, commodities, digital assets, or alternative investments (collectible items, private equity, art pieces, etc). The portfolio assets must follow a particular asset allocation and selection strategy that reflects the investors' preference for a predetermined objective. An investor can choose to follow strategies such as active, passive, value-oriented, risk-averse, aggressive, tax-efficient, etc. The portfolio manager should also consider the time horizons that best align with their strategy to fully capitalize on the portfolios at hand. There are additional considerations to have, such as liquidity needs, market conditions, cost management, and performance monitoring, among many others. The task of managing a portfolio is complex, particularly given the various considerations to have when constructing a portfolio, which is why there have been so many principles and approaches presented over the years.

Next, we define portfolio risk management. We define it as the systematic process of identifying, analyzing, and managing the various financial risks to which a portfolio is exposed across different periods of time with the objective of safeguarding and optimizing the returns generated by the portfolio. Portfolio risk management aims to model and anticipate potential losses and implement strategies that mitigate these risks while aligning with the investor's return objectives and risk preferences [52]. Ideal portfolio risk management approaches tend to focus on diversification [53], strategic asset allocation tailored to the investor's risk appetite, dynamic responsiveness [54], and the use of quantitative measures that allow investors to exploit the risk/return tradeoff [55]. These strategies to control or mitigate risk are considered the overarching framework that portfolio managers typically adhere to. Once a strategic foundation is established, the portfolio manager can refine the details to meet the specific goals behind the strategy. A critical element within this framework is portfolio optimization, which involves the continuous adjustment of asset allocation to align with risk tolerance and market conditions, thereby ensuring that the portfolio maintains an optimal balance between risk and return.

We move on to define what we understand as portfolio optimization. The definition of portfolio optimization has evolved over time. Markowitz was the first person to introduce the mean-variance model for optimization, which was defined as selecting proportions of portfolio assets through maximizing expected returns for a specific level of risk or vice-versa [2]. Sharpe refined the definition by incorporating risk into the equation using a beta to measure an asset's volatility in his Capital Asset Pricing Model (CAPM) [56]. This model provided some guidance for investors on formally pricing an asset's market risk. Merton extended the definition by adding the time element [57]. He used multiple-period investments and continuous-time optimization in his Intertemporal Capital Asset Pricing model (ICAPM). Black went further and created a model that integrated investor views and expectations into the optimization process using a Bayesian framework for updating expected returns [58]. Lo introduced the idea of a dynamic interaction between market and investor behavior through his Adaptive Market Hypothesis [59]. He believed that the interaction between these two elements was a two-way street with influence circling between them. Rachev et al. presented stochastic models incorporating fat-tail distributions and asymmetric risk measures in a book that questioned traditional assumptions known up to that point within the mean-variance framework [60]. This last work paved the way for incorporating more sophisticated statistical and mathematical models in the portfolio optimization process. In the past decade, we have seen the use of machine learning models [9, 61] and quantitative risk management tools [62, 63], among many other approaches making use of different elements from other disciplines other than computer science or math [64, 65]. All these different approaches propose different methods for the optimization process, but they all share a common understanding of what the goal is. Ultimately, portfolio optimization is about selecting the best mix of assets that provide the highest possible return while controlling for risk and other necessary constraints defined by the portfolio manager.

3.1.2 The Portfolio Manager's Challenge

While the challenges of the portfolio manager are various, we focus on the most relevant to our study. Those are related to capturing asset volatility and asset allocation.

3.1.2.1 *Volatility*

We start by defining the volatility of an asset as a measure that captures the variability of a given asset's prices over a specific period. Each asset has its own volatility characteristics depending on various factors related to the nature of the asset or the market in which it is traded. The volatility of assets can be calculated through different methods. Among the most commonly found methods for capturing the volatility of an asset are:

Historical Volatility: is estimated through the calculation of the standard deviation of the returns of an asset within a specific time period.

$$HV = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(R_i - \bar{R})^2} \tag{3.1}$$

where N is the number of observations, R_i is the return at time i, and $\bar{R}$ is the average return.

Exponential moving average: is often used for assets with rapidly changing volatility.

$$\text{EMA}_t = \lambda \times \text{EMA}_{t-1} + (1-\lambda) \times R_t^2 \tag{3.2}$$

where λ is the smoothing parameter (usually between 0.9 and 0.99), R_t is the return at time t and $\text{EMA}_t - 1$ is the Exponential Moving average of the previous period.

GARCH models: Defined as Generalized autoregressive conditional heteroskedasticity (GARCH) [66], these are models mainly employed to model and forecast volatility based on past variances and covariances from a particular time series [63]. The most significant assumption for this model is that past behavior can tell us something about future behavior. For a given time series (X_t), we can have a GARCH (p, q) process that models it in which p and q are the non-negative integers that indicate the order of the model. The parameter p tells us the number of past lagged conditional variance terms σ^2 in consideration for the model. This means that it dictates how many past squared returns X_{t-i}^2 are taken into consideration. The higher the value of p, the further we look back into the past squared returns in order to predict current volatility. The parameter q tells us how many past variances σ_{t-j}^2 are included in our model. The higher the value of q, the higher the number of past volatility terms that we use to forecast the current volatility. There are two main conditions that a process must satisfy in order to be considered a GARCH process: **strict stationarity** and **strict positive valued process**. Strict stationarity refers to the statistical property of maintaining the mean and variance unchanged over time. A strict positive valued process requires the volatility σ_t to remain strictly positive to resemble real-life scenarios. An example of a GARCH(1,1) is:

$$\sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{3.3}$$

In which, σ_t^2 is the conditional variance at time t, α_0 is the Constant term which must be positive to ensure $\sigma_t^2 > 0$, α_1 is the coefficient for the last period's squared return, X_{t-1}^2 is the last period's squared return, β_1 is the coefficient for the last period's conditional variance. Lastly, σ_{t-1}^2 is the conditional variance of the last period.

All sorts of factors can influence the volatility of an asset. The main known ones are related to macroeconomic factors [67], Geopolitical risks [68], investor sentiment [69], and regulatory and policy changes [70]. While influential factors vary depending on the asset class, the mentioned ones have some degree of effect on most of the different asset classes. The assessment and understanding of volatility posits a demanding challenge to portfolio managers as it introduces an uncertainty factor in the price dynamics that guides the portfolio construction process. In particular, the portfolio manager needs to consider all the potentially influential factors that can impact the volatility of a specific asset of interest and the repercussions that this might have on an established portfolio.

The Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) variation of the GARCH was introduced by Daniel Nelson in 1991 [71]. This variation is considered to be more flexible in capturing traditional and inverse leverage effects of the assets it models. An EGARCH(1,1) formulation in detail would look like:

$$\log(\sigma_t^2) = \alpha_0 + \alpha_1 \log(\sigma_{t-1}^2) + \beta_1 |Z_{t-1}| + \gamma_1 Z_{t-1} \tag{3.4}$$

Most of the terms are defined similarly to the GARCH model seen in equation 2.3. The term $\gamma_j Z_{t-j}$ captures the leverage effect. The sign and value of the γ parameter determine the direction and strength of the leverage effect. A negative γ implies a traditional leverage effect, while a positive γ implies an inverse leverage effect.

3.1.2.2 *Asset Allocation*

The selection of assets that constitute a portfolio represents one of the most crucial decisions for a portfolio manager, as these assets must align with the investor's investment strategies, risk tolerance, financial models, and overall return expectations. According to Babaei et al., effective asset allocation involves identifying the efficient frontier, a cornerstone concept of Modern Portfolio Theory introduced by Markowitz in 1952 [72]. The efficient frontier consists of portfolios that optimize the expected return at each level of risk. Markowitz explains that a portfolio belongs to the efficient frontier if it delivers the maximum expected return for a given level of risk, thereby establishing an optimal balance between risk and return [2]. The efficient frontier also serves as a visual tool to identify the best-performing results of a given strategy; in our case, it highlights the outcomes of different optimization algorithms, as evidenced in various studies [10, 73, 74].

3.2 Portfolios Optimization Models

3.2.1 Portfolios Optimization Problems

For a given portfolio of assets, we can use a mathematical formulation to determine an optimal allocation that meets specific objectives and constraints. Such mathematical formulation is centered around an objective function, which quantifies the goal of the optimization process. Among the most known and relevant portfolio optimization problems we can find:

Mean-variance optimization

The mean-variance model is one of the first and simplest models to exist. The principle behind the mean-variance theory is that we are looking to minimize the variance of a given expected return. The model quantifies the risk level through the covariance between the pairing of the portfolio assets. The only constraints imposed on the model are related to budget non-negativity and risk tolerance. While simple, this model does distance itself from actual trading conditions and particularly from the many constraints that portfolios have in real financial markets. The model initially proposed by Markowitz [2] is expressed as:

$$
\begin{aligned}
\min_{\mathbf{w}} \quad & \mathbf{w}^{\top}\Sigma\mathbf{w} \\
\text{subject to} \quad & \mathbf{w}^{\top -} = \mu_p, \\
& \sum_{i=1}^{n} w_i = 1.
\end{aligned}
\tag{3.5}
$$

In which $\mathbf{w}$ is the vector of portfolio weights, Σ is the covariance matrix of asset returns, $\mathbf{w}^{\top}\Sigma\mathbf{w}$ is the variance of the portfolio's return, $\boldsymbol{\mu}$ is the vector of expected returns for each asset, $\mathbf{w}^{\top -} = \mu_p$ is the constraint that the portfolio's expected return equals a target return μ_p, and $\sum_{i=1}^{n} w_i = 1$ is the constraint that all portfolio weights sum to 1.

Risk-Parity

The concept of risk parity is commonly defined as an approach for asset allocation in which the portfolio created has assets that provide an equal risk contribution [75]. Some studies even go further and relax the risk-parity condition in order to enable more room for maneuvering diversification approaches [76, 77]. The common mathematical formulation for this approach is:

$$
\text{Minimize} \left(\sum_{i=1}^{n} \left(w_i \frac{\partial \sigma_p}{\partial w_i} - \frac{1}{n} \right)^2 \right) \tag{3.6}
$$

where n is the total number of assets, σ_η is the portfolio standard deviation, and w_i is the weight of the i-th asset.

CAPM-based optimization

Chen defines the capital asset pricing model (CAPM) as the formalization of the mean-variance optimization of a risky portfolio when interacting with a risk-free investment [78]. In his study, he uses bonds as a risk-free measure. Ultimately, the CAPM presents a tradeoff between market risk and expected return under the assumption of an efficient market. It is mathematically formalized as:

$$E\left(R_i\right) = R_f + \beta_i\left(E\left(R_m\right) - R_f\right) \tag{3.7}$$

Where $\beta_i = \frac{\text{Cov}(R_i,R_m)}{\text{Var}(R_m)}$ Here, $E\left(R_i\right)$ is the expected return on asset i, R_f is the risk-free rate, β_i is the beta of asset i, $E\left(R_m\right)$ is the expected market return, Cov $\left(R_i, R_m\right)$ is the covariance between the return of asset i and the market return, and Var $\left(R_m\right)$ is the variance of the market return.

Value at Risk (VaR)

It provides a single-point estimate of the worst-case scenario (the loss). It addresses the question: what is the chance that my losses won't exceed a certain amount of X? If that chance equals my confidence level (alpha), then X is my value at risk (VAR). The VaR fails to satisfy the sub-additivity axiom, which, according to Clement et al. [12], is an expected requirement for any proper risk measure. The sub-additivity axiom states the risk of two portfolios combined should always be less or equal to the sum of their risks (individually). This axiom basically hints that diversification should not increase risk. Another shortcoming of VaR is that it fails to address tail risk events [10]. The VaR is often formulated as:

$$\text{VaR}_\alpha = \mu_p - z_\alpha \sigma_p \tag{3.8}$$

where μ_p is the mean return of the portfolio, σ_p is the standard deviation of the portfolio returns, and z_α is the z-score corresponding to the confidence level α.

Conditional Value at Risk (CVaR)

Rockefellar et al. introduced the Conditional Value at Risk (CVaR) as a measure of risk that provided a better comprehensive risk assessment, particularly at the end of the loss distribution (something that the VaR fails to do) [7]. The CVaR (or Expected Shortfall) captures the tail risk by averaging the extreme losses, allowing the investor to determine on average how much he/she could lose after a given alpha. This is particularly useful for turbulent economic times and/or assets with considerable tail risk [10]. They formulated the CVaR in the following way:

$$\text{CVaR}_\alpha = \frac{1}{1-\alpha} \int_\alpha^1 \text{VaR}_\gamma \, d\gamma \tag{3.9}$$

where VaR_γ is the VaR at confidence level γ and α is the specified confidence level for CVaR.

3.2.2 Bridging Scenarios with Reality: Copulas

The use of copulas allows us to create a wide range of plausible market scenarios, particularly ones in which we see extreme events (the tails of the distribution). The scenarios created by the copulas provide an advantageous starting point for our optimization algorithms, given that the data that they simulate will model the underlying dependencies between the assets of the portfolio. These dependencies are different than those exploited through the GOMEA, which focuses on exploiting the dependencies within its evolving solutions. In a sense, copulas work as a potential bridge between simulated scenarios and actual market dynamics.

A copula is a mathematical function that isolates the dependence structure from the marginal distributions (individual behavior of each variable) in the relationship between two or more variables. Copulas can capture linear and nonlinear dependencies, meaning we can understand dependence from a more profound perspective, other than correlation [63]. Traditional correlation measures tend to focus on central tendencies, such as the mean. In contrast, copulas use quantile-based dependence, which enables the depiction of the dependence of extreme outcomes. Such characteristics benefit volatile assets such as cryptocurrencies, which are known for having extreme price movements. From a mathematical standpoint, a d-dimensional copula is defined by McNeil as a distribution function on $[0,1]^d$ with standard uniform marginal distributions [63]. In addition to the definition, they provide three mandatory properties needed for a distribution function to hold:

- **Boundary Conditions with 0:** This property suggests that if a copula takes as an input a value of 0 then the output should also be 0. This is expressed mathematically as $C(u_1, \ldots, u_d) = 0$ if $u_i = 0$ for any i .
- **Uniform Marginals:** This property states that when observing a variable within a copula in isolation, we should see a predictable and uniform pattern. This is expressed mathematically as $C(1, \ldots, 1, u_i, 1, \ldots, 1) = u_i$ for all $i \in \{1, \ldots, d\}, u_i \in [0, 1]$.
- **Non-negative Volume:** This property explains that a copula should respect the basic non-negativity rules of probability in order to be considered a valid distribution function. This means that all regions within the domain of a copula should be non-negative.

There are various families of copulas, each with their own merits and suitability for different types of problems. For this study, we consider three particular families of copulas as potential candidates: Elliptical copulas, Archimedean copulas, and Vine Copulas.

Elliptical Copulas

This family of copulas comes from elliptical distributions and comprises the multivariate t-distributions and the multivariate normal Gaussian. The Gaussian (normal) copula is mainly preferred for its capacity to model linear correlation structures, but it is also known to lack the capacity for capturing tail dependence. On the other hand, the t-copula does capture tail dependence, making it an interesting alternative for assessing returns. Paolella et al. uses the t-copula in their study about portfolio optimization with T-GARCH univariate margins [79]. They mention that given the simplicity of the copula and the lack of robust literature to support the use of other more complex copulas, they opted for a straightforward approach.

Archimedean Copulas

This family of copulas has as its core component the use of a univariate function as a generator function, simplifying the multivariate dependence structure definition process. Typical examples are the Gumbel, Clayton, and Frank copulas. Gumbel copulas tend to show strong upper tail dependence, Clayton copulas are more inclined towards showing strong lower tail dependence, and Frank copulas do not exhibit any tail dependence at all. We can find portfolio optimization and allocation studies that use this family of copulas, given they tend to perform well in simple and low-dimensional dependencies [80, 81].

Vine Copulas

This family of copulas is built using hierarchical structures of trees, in which each branch represents different levels of dependencies between variables. Vine copulas construct the dependency structure using simpler bivariate copulas (making them highly flexible to customize pairwise relationships between assets). Such capacities allow them to handle high-dimensional data and permit them to explore more complex relationship dynamics among the assets. There are three main subclasses of vine copulas: C-vine, R-vine, and D-vine. *Canonical Vine (C-vine)* has a structured approach in which one variable (considered a central node) is placed at the center of the dependency structure. *Regular Vine (R-vine)* is the most flexible one, as no strict hierarchical structures are imposed during the dependency modeling process. *Drawable vine (D-vine)* follows a sequential format in which the dependencies are structures along a path of variables. In their study of Particle Swarm Optimization for a cryptocurrency portfolio, Clement and Mbong explain that they use the C-vine copula specification given that it allows modeling the dependence around a preferred variable [12]. Given the known influential prevalence of bitcoin with other cryptocurrencies, we can see the benefit of such a configuration.

The decision on which copula family is best for portfolio optimization depends on our goals and approaches. We need to weigh our needs and the specific problem structure in order to make a selection, given that there is no clear framework for the selection [82]. However, given the nature of our assets of study, cryptocurrencies, we opted to make use of the Vine family of Copulas. The advantage of using the Vine family, particularly through the `pyvinecopulib` Python library, lies in its `Vinecop` function, which automatically fits the best structure from its three main subclasses based on our data. This decision to rely on the `Vinecop` automated selection is driven by the absence of conclusive evidence of a marked hierarchical structure among our cryptocurrencies. This is corroborated by a study conducted by Kokoszczyński et al., who used data spanning from September 2015 to May 2018, where they used a Minimum Spanning Tree analysis—a method in network analysis that identifies crucial connections within a network—to understand the hierarchical structure of the cryptocurrency market [83]. Their findings exhibit that Bitcoin acts as a 'superhub' within the network, suggesting it has a massively influential role among other cryptocurrencies. Nonetheless, they also pointed out that this influence waned in the later periods of the study, persisting but to a lesser degree.

3.3 Systematic Overview

Optimizing a portfolio with traditional assets like stocks and bonds is already complex. However, integrating cryptocurrencies adds a further layer of complexity. This is especially true given the volatile and dynamic nature of crypto-assets, which often makes conventional financial principles and heuristics less effective or even irrelevant. One major challenge is that traditional models are based on certain assumptions that don't hold up well with cryptocurrencies. For instance, these models often assume that returns follow a normal distribution and struggle with accurately capturing the high volatility of crypto assets, even with advanced tools like ARCH and GARCH models. Additionally, dynamic correlations are unique to these assets, and specific risks, such as cybersecurity and fraud, are more pronounced when using cryptocurrencies. A potential approach for cryptocurrency portfolio optimization is metaheuristics. Metaheuristics are considered algorithms devised to solve different types of problems by exploring the search space for potential solutions in an effective yet non-deterministic manner. They are mainly employed for problems in which the search space is vast and there is a likely presence of multiple local optima. What makes them particularly useful in problems such as optimization problems are their adaptability, scalability, hybridization capabilities, stochastic nature, and convergence properties, among others. The nature of metaheuristics allows them to tackle problems that traditional methods tend to struggle with, such as high dimensionality, non-linearity, deceptive convergence, and dynamic dependence structures. This paper uses a particular type of metaheuristics given their innovative approach: Evolutionary algorithms.

In our next chapter, we will examine some studies that employ well-known evolutionary algorithms for cryptocurrency portfolio optimization alongside an alternative machine learning approach. We will explore these methodologies to contrast them with our proposed algorithm, the Gene-Pool Optimal Mixing Algorithm (GOMEA), analyzing the theoretical differences between these approaches. Ultimately, we aim to set the stage for the experimentation process, where we will test whether GOMEA outperforms the other algorithms.

4. Evolutionary Algorithms

4.1 EAs and Alternative Approaches in Portfolio Optimization

4.1.1 On evolutionary algorithms and their advantages

As Hurbans explains, the living organisms we see today are the product of a chaotic, non-linear evolutionary process in which the best traits of the organisms result from the most adequate fit to their environment, leading to their survival [84]. Evolutionary Algorithms are considered a class of metaheuristics that leverage the principles of Darwinian theory (natural selection) to evolve a set of solutions for a specific problem, iteratively enhancing their fitness across successive generations [85]. Our study focuses on single-objective constrained evolutionary algorithms, which optimize a specific objective function while incorporating relevant real-world constraints. This method aims to develop a model-free strategy, with the aim of avoiding unrealistic assumptions often found in traditional models. Evolutionary algorithms have an advantage over traditional methods given that they employ stochastic (through recombination and mutation processes) and dynamic (part of the adaptation process) methods during the exploration process of the search space, which are more adept to the financial environments [51].

EAs are also known for working well on non-linear and complex optimization problems. They can transfer the information gained from one task to solve another, focus on more promising solutions in the search space based on probabilistic models, and transform the representation of solutions or the search space (by mapping to a different dimensional space, transforming the features, or warping the landscape, among other transformation methods) [86]. Some studies also point out how EAs are well-suited for handling high-dimensional problems. In their study, Bhattacharya et al. explain that EAs are not only capable of handling high-dimensional problems given their adaptability and flexibility but also that if we design the proper test functions and evaluation sets, we can evaluate and refine the EAs to be even more efficient in these convoluted problems [87]. Another relevant area in which EAs excel is in problems with multiple local optima. Singh explains that EAs have the capacity to find multiple solutions within each iteration [88]. This is ignited by its population-based approach that opens up the possibility of exploration of different regions of the search space. In particular, they explain that specific EAs tend to deploy niche techniques (such as dividing the population into sub-populations), which allows for segmentation of the search space and prevents falling prey to sub-optimal local optima solutions.

4.1.2 Relevant EAs in Portfolio Optimization

Genetic Algorithms

A well-known approach to the portfolio optimization problem is the use of genetic algorithms (GAs). GAs were first introduced by Holland in 1975 as a heuristic optimization technique designed to mimic the Darwinian principle that only the fittest individuals survive [89]. Essentially, a GA follows the process observed in biological evolution, which includes the phases of initialization, crossover, mutation, and selection. Studies in portfolio optimization [90] utilizing this approach include research on the Indonesian stock market [91, 24], analyses of European Exchange-Traded Funds [92], and investigations into stocks from different geographic regions [93]. A representation of the steps taken in genetic algorithms can be:

Algorithm 1: Genetic Algorithm Procedure

```
Function GeneticAlgorithm():
    Generate an initial population
    Evaluate fitness of individuals in the population
    repeat
        Select parents from the population
        Recombine (mate) parents to produce children using
          crossover and mutation operators
        Evaluate fitness of the children
        Replace some or all of the population with the children
    until a satisfactory solution has been found
```

While there is a wide array of studies that make use of the GA with different risk measures, there are very few that employ this algorithm for CVaR portfolio optimization. Setiwan et al. is one the few who take the CVaR as their risk measure [91]. In their study, they employ the following formulation for the CVaR as their objective function:

$$
\begin{aligned}
&\min \alpha + \frac{1}{t(1-\beta)} \sum_{k=1}^{t} u_k \\
&\text{subject to} \\
&\sum_{j=1}^{m} x_j c_j \leq b, \\
&\mathbf{x}^T \mathbf{y}_k \geq -\alpha - u_k, \\
&x_j \in \mathbb{Z},\ \forall j = 1, \ldots, m.
\end{aligned}
\tag{4.1}
$$

In which α is the threshold VaR value, $1-\beta$ is the proportion of worst-case scenarios considered, and β is the confidence level. Additionally, we define $u_k = [-\mathbf{x}^T \mathbf{y}_k - \alpha]$ in which $\mathbf{x}^T$ is the transposed weight vector, $\mathbf{y}_k$ is a simulated return vector, also considered to be a specific scenario k part of the total number of scenarios t. The objective function is subject to three constraints. The first is a budget constraint $\sum_{j=1}^{m} x_j c_j \leq b$, which prevents the allocation of assets from exceeding the available resources. The second is a loss constraint $\mathbf{x}^T\mathbf{y}_k \geq -\alpha - u_k$, which bounds the potential loss per scenario. Lastly, the integrality constraint $x_j \in \mathbb{Z}, \ \forall j = 1, \ldots, m$ which states that all the decision variables x_j must be integers. Furthermore, the framework of the study is based on a specific constraint: the minimum transaction lot, which is defined as the smallest quantity of an asset that can be traded in a financial market. Although this imposed constraint is relevant, considering that financial markets often operate with predetermined lot sizes, the authors acknowledge the need for additional constraints commonly found in similar frameworks, such as cardinality or leverage constraints. Nevertheless, the authors conclude that genetic algorithms significantly impact the portfolio optimization process. They highlight that the selection of parameters, including the number of generations, confidence level, and mutation/crossover properties, greatly influences the algorithm's performance in terms of the Sharpe ratio scores.

Particle Swarm Optimization

A different approach towards portfolio optimization can be found through another evolutionary stochastic optimization algorithm named Particle Swarm Optimization (PSO). Firstly introduced by Kennedy in 1995 [94], the algorithm is based on the social behaviors of animals, such as flocks of birds or schools of fish. In particular, the algorithm seeks to focus on how individual members of these large groups of animals modify their behavior based on the result of their actions and those around them. This allows for a balanced exploration and exploitation of areas of the solution space. A pseudocode for the PSO algorithms can be:

Algorithm 2: Particle Swarm Optimization Procedure

```
Function ParticleSwarmOptimization():
    Initialize:
        Set number of particles in the swarm and maximum number
          of iterations
        for each particle do
            Set initial position randomly within the problem space
            Set initial velocity randomly
            Record initial position as the particle's personal best
            Determine the global best position among all particles
    PSO Loop:
        for each iteration do
            for each particle do
                Update velocity based on current velocity, personal
                  best, and global best
                Update position by applying the new velocity
                if new position is better than personal best then
                    Update personal best to the new position
                if new position is better than global best then
                    Update global best to the new position
            Repeat PSO Loop until the maximum number of
              iterations is reached
    Output:
        Return the global best position as the solution
```

In a study that employs the use of a Copula Particle Swarm Optimization (CPSO) portfolio strategy, Clement et al. address the CVaR optimization problem [12]. The particular formulation of the CVaR that they optimize is the following:

$$\arg\min_{\omega} \left\{ \xi + [(1-\alpha)n]^{-1} \sum_{i=1}^{n} \max \{ f_i(\omega, r) - \xi, 0 \} \right\}$$
$$\text{subject to} \tag{4.2}$$
$$\begin{cases} \sum_{i=1}^{n} \omega_i = 1, \\ \sum_{i=1}^{n} E[c_i]\, \omega_i \geq 0, \\ \omega_i \geq 0, \qquad i = 1, \ldots, n. \end{cases}$$

In which $\omega = (\omega_1, \ldots, \omega_n)$ is the weights vector for the assets, ξ is the Value at Risk (VaR) value, α is the confidence level, and n is the number of total assets. The term $\sum_{i=1}^{n}$ is the summation of all assets. The function $f_i(\omega, r)$ is the loss function for a portfolio characterized by a weights vector ω and a return vector r, which is composed of n assets. Their study compares the performance of the CPSO with three different strategies: Copula Differential Evolution (CDE), Global minimum variance (GVM), and minimum tail dependent (MTD). The results suggest that CPSO is an auspicious alternative for managing risk during volatile periods. The study also suggests that through the use of stablecoins in the portfolio, the CPSO effectively hedges market volatility during periods of market turmoil. While the study shows some promising results, some design decisions could be called into question. Decisions such as selecting the GARCH models for the volatility modeling [43] or selecting a specific family of copulas (Vine Copulas in this case) can all be examined.

4.1.3 Alternative approach through Machine Learning

Deep Reinforced Learning

An alternative approach to the CVaR portfolio optimization problem regards the use of Deep Reinforcement Learning (DRL). An example of such an approach is seen in the work of Cui, who uses a copula-based approach in a scenario-based study that employs a DRL algorithm to optimize a cryptocurrency portfolio [10]. They explain that DRL has two significant advantages over other novel techniques, such as Deep Learning (DL) or machine learning approaches, namely that their DRL model works under realistic model assumptions and that the model allows for a continuous asset relocation process. The DRL overcomes certain limitations seen in other approaches that are heavily dependent on price trend prediction models, which tend to carry hefty, unpractical assumptions. The DRL model is based on the Proximal Policy Optimization (PPO) algorithm created by OpenAI as their base Reinforcement Learning algorithm [95]. The pseudocode for the PPO algorithm is the following:

Algorithm 3: Proximal Policy Optimization (PPO) Algorithm

```
Function ProximalPolicyOptimization(θ, N, T, I, K, M):
    for iteration = 1, 2, ..., I do
        for actor = 1, 2, ..., N do
            Run policy π_θold in the environment for T timesteps
            Compute advantage estimates Â_1, ..., Â_T
        Optimize surrogate L with respect to θ for K epochs and
         minibatch size M ≤ NT
        θ_old ← θ
```

In which θ represents the initial policy parameters, which are the weights and biases of the neural network representing the policy, N is the number of actors (or agents) used to collect data, T is the number of timesteps each actor collects data for in each iteration I, K is the number of epochs for which the surrogate objective L is optimized in each iteration, and M is the minibatch size used in the optimization, which must be less than or equal to $N \times T$.

An important aspect relevant to the framework of Cui's paper is the selection of the CVaR formulation. The objective function selected by Cui seems to follow the same CVaR structure as Clement and Mbong [12] but differs by taking a more probabilistic approach. In their formulation, they make use of the parameter p_i, which is the probability of scenario i. Through this parameter, the authors assign a specific likelihood to each scenario, allowing the calculation of a weighted average and avoiding the need for a parameter such as n seen in equation 4.1. In that equation, n is used to calculate an average of the scenarios by summing over the scenarios and dividing by n. The mathematical formulation for Cui's CVaR format which they optimize is:

$$
\begin{aligned}
&\min \left[\alpha + (1-\beta)^{-1} \sum_{i=1}^{N} p_i z_i\right] \\
&\text{s.t.} \\
&z_i = \left(f\left(x, y_i\right) - \alpha\right)^{+}, \quad i = 1, \ldots, N \\
&z_i \geq f\left(x, y_i\right) - \alpha, \quad i = 1, \ldots, N \\
&z_i \geq 0, \quad i = 1, \ldots, N \\
&\sum_{i=1}^{N} p_i R^i \geq R^* \\
&\alpha \in \mathbb{R} \\
&x \in \mathbb{R}^n
\end{aligned}
\tag{4.3}
$$

In which α is the Value-at-Risk (VaR) value, β is the confidence level, N is the total number of scenarios, p_i is the probability of the i-the scenario and z_i is variable defined in terms of the loss function $f(x, y_i)$ and α. Within the loss function, x is the weights vector (the decision vector), and y_i is the i-th scenario created from a copula rather than using probability density function $p(\xi)$. The term $(f(x, y_i) - \alpha)^{+}$ represents the loss beyond the VaR threshold, which cannot be negative according to the constraints.

Ultimately, the result of this study shows that the portfolios constructed with the DRL and the CVaR framework succeed in capturing tail risk and outperform other portfolios constructed under different risk measures in terms of offering higher returns and lower risks.

4.2 GOMEA in Portfolio Optimization

4.2.1 Original formulation and CVaR adaptation

The main algorithm of interest for the purpose of this study is the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA). Introduced by Thierens and Bosman, it is defined as an evolutionary algorithm that performs a memetic variation of solutions by exploiting linkage information between the elements of the genotype [96]. Optimal mixing can be defined as the capacity of an algorithm to combine partial solutions from parent solutions, specifically their beneficial traits or those traits that approximate them to better scores in the fitness function, to produce improved offspring solutions in each iteration of new generations. The overarching general algorithms can be described as:

Algorithm 4: GOMEA General Outline

Function RUNGOMEA(n):
    $\rho \leftarrow$ initializePopulation(n)
    while *-shouldTerminate()* **do**
        $F \leftarrow$ buildLinkageModel(P)
        for $\mathcal{P}_i \in \mathscr{P}$ **do**
            $O_i \leftarrow \text{GOM}(\mathcal{P}_i, F, \mathcal{P})$
        $\mathcal{P} \leftarrow O = \{O_1, \ldots, O_n\}$

The GOMEA algorithm has the following key components:

Population Initialization

The GOMEA's main formulation employs the Interleaved Multi-start Scheme (IMS) to initialize and manage its populations [25]. This approach entails operating multiple populations of varying sizes simultaneously yet alternatively. Consequently, these populations are at different stages of evolution at any given time, effectively leveraging the strengths of both small and large populations. This interleaved management of populations contributes to a diverse solution space exploration, helping the algorithm avoid premature convergence and getting trapped in local optima. The initial creation of these populations incorporates randomness, ensuring a wide-ranging search across potential solutions.

Build-up of the linkage models: Family of Subset Structures (FOS)

Once the algorithm has initialized the population and selected the parent solutions, typically based on their fitness, it proceeds to build the linkage models. The models capture the dependencies structures found within the optimization problems. Identifying the complex linkages is an imperative step in the process of building optimal solutions, given that if ignored, the algorithm will miss the potential of constructing better solutions. This linkage learning process is achieved by identifying substructures (genes) that tend to co-occur in high-quality solutions. The identified linkages are then systematically incorporated into the Family of subsets (FOS) structure for use in the subsequent steps. The FOS describes the linkage models by creating a set of subsets of genes that the GOM variation operator will use. A Family of Subsets can be represented as FOS $\mathcal{F} = \{\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{k-q}\}$, in which each $\mathcal{F}_i$ is known as a linkage set and is made of a number of indices that represent the location of specific genes. It is important to mention that the genes within a linkage set are considered to be jointly dependent, establishing a closed dependency among the variables of such a set. We will take into consideration three linkage model structures as potential candidates:

Marginal Product (MP) linkage models:

It is a type of linkage model that allows us to group all the variables from the problem into different linkage sets without creating any overlap. This model can only be used in real-valued optimization and can take two forms: The univariate model and the Full linkage model. The *univariate model* is selected under the assumption that each variable in the problem is independent of each other. This is represented through $\mathcal{F}^{\text{Uni}} = \{\{0\}, \{1\}, \ldots, \{\ell - 1\}\}$. The *full linkage model* runs under the assumption that all variables are interdependent, suggesting that a change to one variable will affect all the other variables. Such assumption implies that the entire set of indices is considered, directly modeling scenarios based on existent dependencies across all variable pairs. This is represented as $\mathcal{F}^{\text{Full}} = \{\{0, 1, \ldots, \ell - 1\}\}$.

Linkage Tree (LT) model:

This type of linkage model allows for more complex and nuanced dependencies among the variables, incorporating various levels of dependencies. The model is considered to be a hierarchical model as it shows how variables of a problem are related to each other across different dependency levels. The construction of a LT is guided through the Unweighted Pair Group Method with the Arithmetic mean (UPGMA) clustering method. This method continuously merges the most similar linkage sets (starting with the single-variable linkage sets) until one linkage set remains (composed of all the variables from the problem).

Conditional models:

This type of model allows us to introduce variation into a small subset of variables while keeping the remaining variables unchanged from their original 'parent' setup. The intuition behind this approach is to analyze how minor adjustments in the variables impact the overall function of the algorithm. These models are particularly advantageous in optimization problems in which the variables have intricate and complex interdependencies. By focusing on implementing changes in a few variables at a time, the Conditional models allow us to better understand the dependency structures by gradually exploring them.

Variation Operators: GOM

The Gene-pool Optimal Mixing (GOM) is the engine behind the GOMEA algorithm as it is its variation operator. The GOM, guided by the FOS structure and through an iterative process, mixes the genes from different parent solutions into an offspring solution. The offspring solutions are evaluated in terms of their fitness and compared to the best available solution. If the offspring solutions demonstrate better fitness than the best solution, the genes in the subset of the best solution are replaced by those from the offspring, and their fitness is updated accordingly. On the other hand, if the offspring do not offer a better fitness than the best solution, then genes within the offspring subset are reverted to those in the best solution. At the end of a complete run of the GOM algorithm, the offspring is either (1) the same as the best solution or (2) a new, better solution. The algorithm can then be represented through pseudocode in the following manner:

Algorithm 5: GOM pseudocode

```
Function OMEA::GOM(x):
    b ← o ← x
    fitness[b] ← fitness[o] ← fitness[x]
    for i ∈ {0, 1, ..., |F| − 1} do
        p ← RANDOM ({P_0, P_1, ..., P_{n−1}})
        o_{F^i} ← p_{F^i}
        if o_{F^i} ≠ b_{F^i} then
            EVALUATEFITNESS (o)
            if fitness[o] > fitness[b] then
                b_{F^i} ← o_{F^i}
                fitness[b] ← fitness[o]
            else
                o_{F^i} ← b_{F^i}
                fitness[o] ← fitness[b]
    return o
```

Survivor Selection

The process conducted in the GOM operator is replicated across the population of solutions throughout multiple generations of the algorithm. While the algorithm initially does not necessarily find the most optimal solutions, it will gradually increase the quality of the solutions.

4.2.2 Problem structure

For this study, we will make use of the GOMEA algorithm as developed in the work of Bouter and Bosman [25]. In their study, they introduce a GOMEA library in Python that is based on the original formulation of the GOMEA algorithm. This library offers both Grey Box Optimization

(GBO) and Black Box Optimization (BBO), each available in discrete and real-valued versions, with the option to customize each version. The GBO is recommended for scenarios in which the problem structure is known and can be broken down into subfunctions with exploitable interactions, facilitating partial evaluations of variable subsets to guide the optimization process. On the other hand, the BBO takes an unbiased approach towards a presented problem, making it an attractive, flexible version for high complex problems in which solutions are analysed based solely on their output or performance. For the purposes of this study we tested both the GBO and the BBO in their respective real valued versions, given that our variables, namely the weights of each asset of our portfolio, are real numbers. However, during our initial tests, it became apparent that our problem structure could not be decomposed into subfunctions, leading us to select the real-valued BBO version of GOMEA for its ability to handle single-objective optimization effectively.

The objective function that we will seek to minimize is the CVaR formulation found in the work of Cui, which is known to be compatible with optimization algorithms [10]. Additionally, we will make a specific selection from the sub-packages from the `consval` branch of the `Gomea` package. First, we will use the base class type `BBOFitnessFunctionRealValued`, which we will customize accordingly to the CVaR portfolio optimization problem. Additionally, we will override the following methods based on the number of assets in our portfolio:

- `objective_function(self, objective_index, variables)`
- `constraint_function(self, variables)`

We make the following design choices for the sub-packages for our custom BBO GOMEA algorithm,further detail on the reasoning behind the selection will be provided in the methodology section:

Subpackage	Selection for this study
Grey-BOX or Black-Box	BBO
Optimization	real valued
Fitness	BBOFitnessFunctionRealValued
Linkage	Full

Table 4.1: Design selection for the GOMEA algorithm

4.2.3 Solution Space

The solution space for our portfolio optimization problem that we will explore has some particular characteristics. We expect multiple local optima to exist due to the presence of a non-normal distribution followed by the cryptocurrency returns, making it a non-convex, high-dimensional, and dynamic space. As mentioned, the search space is likely to contain many local optima and suboptimal regions.

5. Methodology

5.1 Experimental Design

We adopt the methodology presented by Cui et al. [10], which involves constructing a CVaR (Conditional Value at Risk) efficient portfolio of cryptocurrency assets through two main components: the computation of CVaR and a portfolio optimization algorithm. In our study, the optimization algorithm that we use is the Black-Box Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) from Bouter and Bosman's study [25], and compare it to a Genetic Algorithm constructed from pseudocode and a Particle Swarm optimization constructed using the `pyswarm` python package. Additionally, to assess the performance in terms of the risk/return trade-off, we construct efficient frontiers for each algorithm and compare them. After constructing and testing the model with simulated data, we will make use of historical data to backtest our algorithm. This step is critical to determine whether the patterns and behaviors observed in the simulated data are consistent with those in historical data.

For our asset selection, we chose the cryptocurrencies with some of the highest overall market cap in the last six years. We use the information available in Yahoo Finance to retrieve the historical returns for the following cryptocurrencies: Bitcoin, Ethereum, Litecoin, Ripple, and Monero. We use the daily returns of the mentioned asset to keep track of the performance of the crypto-assets by calculating the percentage change in the closing price from one day to the next. Additionally, to make sure that we are making a balanced comparison of the returns of the assets, we transform our data to logarithmic returns by taking the natural logarithm of the ratio of each day's closing price over the previous day's closing price. The returns collected range from the periods of January 2018 to November 2023, providing a window of approximately 5 years of returns. An important point to consider is that we sought to include periods of high volatility, such as those seen in the collapse of Bitcoin in 2018 and the later surge in 2021. Within our selected time window, we can also find the COVID-19 pandemic, posterior international conflicts such as the Russio-Ukranian war, Brexit, and the elections in the United States of 2020. This sample is representative of various types of distress periods, which could have a more substantial impact on the volatility of the prices.

To construct the scenarios, we use Vine copulas that allow the simulation of volatile scenarios (closer to how cryptocurrency markets actually behave), which will be employed as vector y_i the loss function $f(x, y_i)$. We use the `pyvinecopula` package in Python for a process in which we initially transform the array of log returns into pseudo observations. This transformation is applied to generate a set of values between 0 and 1, which represent the original data in terms of their cumulative distribution, effectively making them uniform marginals. The guiding principle behind this is that for copula models, any multivariate distribution can be dissected into its marginal distributions and a copula that delineates the dependency structure among the variables. Therefore, by converting our log returns into uniform marginals, we enable the copula model to extract the dependency structure of our variables accurately. This allows for the proper simulation of further scenarios that preserve the identified dependency structures, providing a robust framework for analyzing potential market movements. In the next step of our copula modeling, we select the potential copula families that the algorithm should consider for fitting. We opt to select as candidates the Gaussian, Clayton, and Student T. Once these two steps are defined, we make use of the `Vinecop` function, which will automatically fit the best possible structure from the options we provided to our data. For our study, we make use of our constructed model to simulate 25000 scenarios, which account for approximately 68 years of financial data. This is reflected in the code:

```
u = pv.to_pseudo_obs(log_returns_array)
controls2 = pv.FitControlsVinecop(family_set=
[pv.BicopFamily.gaussian, pv.BicopFamily.clayton,
```

```
pv.BicopFamily.student])
cop2 = pv.Vinecop(u, controls=controls2)
simulated_data = cop2.simulate(n=25000)
```

Once we have created the simulated scenarios, we can use the GOMEA algorithm. This approach allows the GOMEA algorithm to explore a more refined search space, potentially leading to a more focused and effective search for optimal solutions. We will use the CVaR as our objective function for our GOMEA algorithm. In particular, we will make use of Cui et al. formulation of the CVaR and optimize it:

$$
\begin{aligned}
& \min \left[\alpha + (1-\beta)^{-1} \sum_{i=1}^{N} p_i (f(x, y_i) - \alpha)^+ \right] \\
& \text{s.t.} \\
& z_i = \left(f\left(x, y_i\right) - \alpha\right)^+, \quad i = 1, \ldots, N \\
& z_i \geq f\left(x, y_i\right) - \alpha, \quad i = 1, \ldots, N \\
& z_i \geq 0, \quad i = 1, \ldots, N \\
& \sum_{i=1}^{N} p_i R^i \geq R^* \\
& \alpha \in \mathbb{R} \\
& x \in \mathbb{R}^n
\end{aligned}
\tag{5.1}
$$

In terms of the data that we will use as input for equation 5.1, we make use of the following:

1. For α, the Value-at-Risk (VaR), we will use a single value for the VaR that will be estimated based on the sorted scenario returns, the confidence level, and the corresponding percentile index.

2. For the parameter β, the confidence level, we will use a value of 0.95 as the value for our confidence level.

3. The parameter N, the total number of scenarios, will be set to 25,000 to provide statistical stability given our probabilistic approach.

4. Parameter p_i, the probability of the i-th scenario, will take on the value of $1/25000$ for each scenario in order to provide each scenario a realistic chance of consideration.

5. Within the loss function $f(x, y_i)$, we will initially select for the weight vector x an initial vector of random weights (a list of lists in Python terms): [[w11,w12,w13,w14,w15,w16], [w21,w22,w23,w24,w25,w26], ...]. For the parameter y_i, the vector of scenario returns, will be will use the scenarios generated from the copulas, and it will be a list of tuples (in Python terms) in which each tuple encapsulates a specific scenario [[s11,s12,s13,s14,s15,s16], [s21,s22,s23,s24,s25,s26], ...].

For the GOMEA algorithm make use of the `consval` branch from the `gomea` python library presented by Bouter and Bosman [25]. As explained in their paper, we customize the objective function for the Black Box GOMEA and add the corresponding constraint to prevent the solutions from being negative or not adding to one. By adding this constraints, we ensure that no shortselling is allowed and that the total capital is invested in the portfolio. The objective function that we employ in our code, along with its respective constraint function, is the following:

```
def objective_function(self, objective_index, variables):
    normalized_weights = bicom_normalize_weights_obj_funct(variables,
    min_target=0.01,max_target=0.99)
    var = estimate_var(normalized_weights, self.scenarios,
    self.confidence_level)
    portfolio_return = np.dot(self.scenarios, normalized_weights)
    probabilities = np.full(self.scenarios.shape[0], 1 /
    self.scenarios.shape[0])
    excess_loss = np.maximum(portfolio_return - var, 0)
    cvar_component = np.sum(excess_loss * probabilities)
    denominator = (1 - self.beta) * self.scenarios.shape[0]
    z_i = cvar_component / denominator
    cvar = var + z_i
    return cvar

def constraint_function(self, variables):
    if np.all([(0.0001 <= x <= 1) for x in variables]):
```

```
        sum_variables = np.sum(variables)
        if 0.99 <= sum_variables <= 1:
            return 0
    return 1
```

As part of our parameter selection, we decided to set the value for the parameter `value_to_reach` close to zero. For the linkage model, we opted for the **Full Linkage Model** due to its theoretical compatibility with our study and the assumption that there is interdependence among our assets. We established the lower and upper initial ranges between 0 and 1 to give the initial solutions a constrained random starting point to prevent any bias. Additionally, we decided to exclude the use of the IMS subfunction when establishing the parameters for the `RealValuedGOMEA` class. This decision was made during the hyperparameter tuning phase, during which we noticed that the algorithm performed poorly in terms of convergence and optimization whenever the IMS was paired with the constrained version of the GOMEA. This issue was consistent across various settings of the parameters `max_number_of_populations`, `base_population_size` or `max_number_of_seconds`. We suspect that while the `consval` branch of the library functions adequately with a fixed population, there may be incompatible elements when applying the IMS subfunction within the constrained use of the GOMEA library, especially since this branch is still under development. Given that we excluded the IMS subfunction, then by default, the `max_number_of_populations`, is set to 1. Additionally, and following the instructions presented in Bouter and Bosman's paper, we set the `max_number_of_generations` and `max_number_of_evaluations` to -1 to establish no limit to them, allowing the algorithm to explore the solution space freely [25]. Ultimately, our parameter selection was:

```
scenarios = simulated_data
confidence_level = 95
probabilities = np.full(scenarios.shape[0], 1 / scenarios.shape[0])
beta = 0.95
value_to_reach = 1e-6
frv = BBOforCVaROptimization(scenarios, probabilities, beta, confidence_level,
value_to_reach)
```

```
lm = gomea.linkage.Full
rvgom = gomea.RealValuedGOMEA(fitness=frv,
                              linkage_model=lm,
                              lower_init_range=0,
                              upper_init_range=1,
                              max_number_of_populations=1,
                              max_number_of_generations=-1,
                              base_population_size=250,
                              max_number_of_evaluations=-1,
                              max_number_of_seconds=900)
```

Regarding the data collection employed for constructing our efficient frontier, we conducted 270 runs to gather data on portfolio performances, using the Conditional Value at Risk (CVaR) as our objective function for single-objective optimization with the GOMEA. Once we obtained the optimized weights per run, we compute the corresponding expected returns by multiplying the optimized weights with the simulated/historical returns. These returns(y-axis) were then plotted against their respective CVaR values (x-axis), which were derived from the GOMEA optimization. It is important to emphasize that calculating the expected returns was a simple computation, separate from the optimization techniques used to determine CVaR values. We then select the top 10 portfolios from these data points, representing the efficient frontier for our chosen optimization algorithm. Our approach allows us to construct our efficient frontier systematically, selecting portfolios with the highest expected returns. Depending on the outcomes from our optimization algorithms, we arranged the CVaR values in ascending order to address minimum expected gains and in descending order to handle potential losses.

The financial metrics presented in the result section for both the simulated and historical data are constructed using the mean value of the 270 runs mentioned. For the performance graphs, we use 30 separate runs for the convergence graph and 20 runs for the construction of the hyperparameter tuning performance graphs. Specifically, the base population graph is constructed in increments of 25, and the time termination criteria graph in steps of 30 seconds.

5.2 Data

As data samples for our cryptocurrency, we use the asset's historic prices covering January 2018 to November 2023. We transform our normal returns into logarithmic returns as this is standard practice when analyzing the returns of volatile assets with wide varying returns. The first thing we examine is descriptive statistics. As seen in Table 5.1, we can draw some interesting insights regarding our assets. The first concerns risk. We see the XRP (Ripple) is the riskiest asset in the portfolio, as its standard deviation suggests. Ripple is also the asset that shows the most extreme price movements, as shown by its wide range of gains and losses (-0.5505 and 0.5486). In terms of the quartiles, we see that both Litecoin (LTC) and Monero (XMR) have a wider interquartile range in comparison to Bitcoin (BTC), which invites us to consider that there is a greater spread of daily returns around the median leading to a higher level of volatility and risk. In terms of expected returns, we notice that only Ethereum(ETH) and Bitcoin(BTC) exhibit positive expected returns with some low ceilings in their maximum returns. We have also noticed that Bitcoin appears to be the best asset to invest in as it holds the highest expected return under the lowest standard deviation, making it the most balanced asset in the portfolio.

(a) Count, Mean, Std, and Sharpe Ratio

Ticker	Count	Mean	Std	Sharpe Ratio
BTC-USD	2158	0.0005	0.0370	0.0128
ETH-USD	2158	0.0004	0.0477	0.0094
LTC-USD	2158	−0.0005	0.0507	−0.0108
XMR-USD	2158	−0.0004	0.0499	−0.0111
XRP-USD	2158	−0.0006	0.0572	−0.0072

(b) Interquartile Ranges with Min and Max

Ticker	Min	25%	50%	75%	Max
BTC-USD	−0.4647	−0.0142	0.0007	0.0161	0.1718
ETH-USD	−0.5507	−0.0199	0.0005	0.0235	0.2307
LTC-USD	−0.4491	−0.0237	0.0001	0.0242	0.2906
XMR-USD	−0.5342	−0.0210	0.0019	0.0243	0.3450
XRP-USD	−0.5505	−0.0218	−0.0010	0.0196	0.5486

Table 5.1: Descriptive Statistics of Log Returns for selected Cryptocurrencies

If we examine the historical returns of our assets as depicted in Figure 5.1, we can observe that certain periods of high volatility are shared by all the assets, namely right after 2020 (which can be attributed as the start of the Covid-19 Pandemic) and in the early stages of 2021. From Figure 5.1, we can also notice that each asset has different volatility clusters across different times, suggesting that they have different market dynamics.

In terms of the correlation between assets, we construct a Pearson correlation matrix in Figure 5.2 and find that most of the assets are highly correlated with each other. Only Ripple seems to have some low positive correlation with the other assets in the portfolio. This finding is not surprising as it is commonly known that cryptoassets tend to be highly influenced by each other movements.

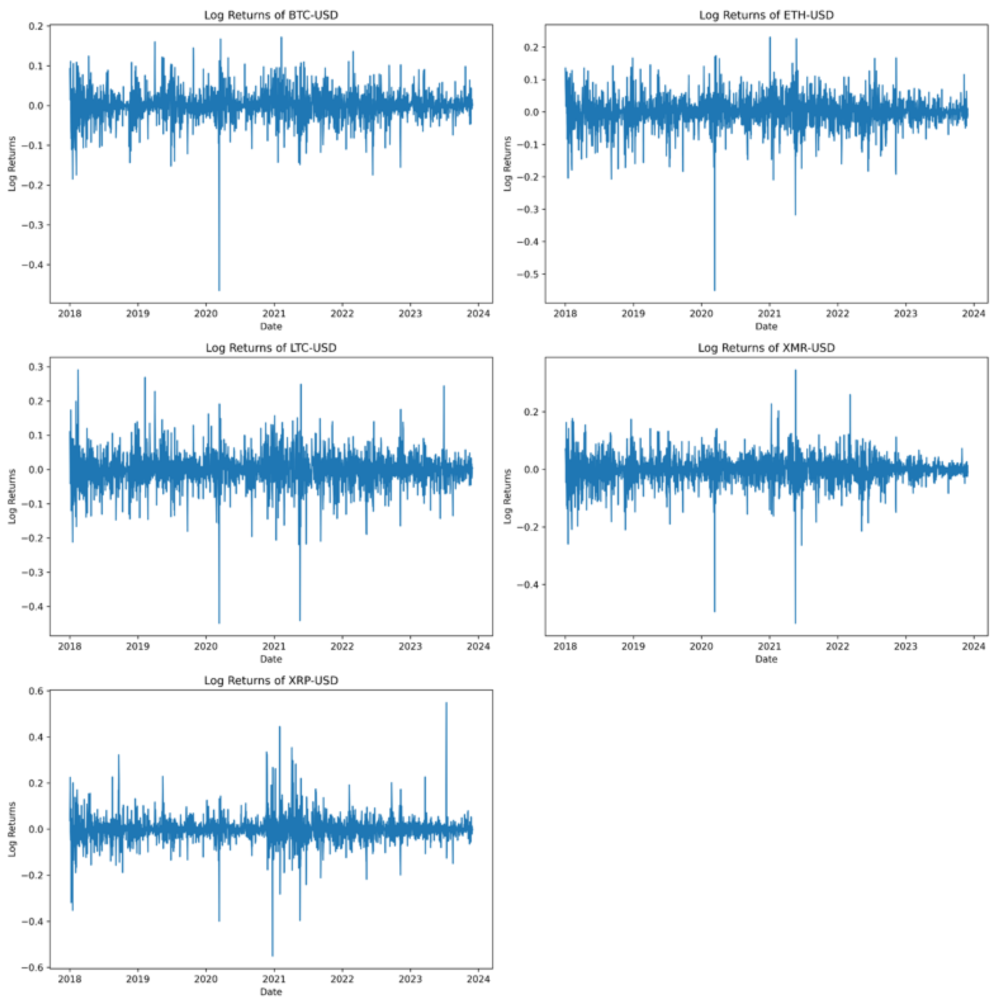


Figure 5.1: Volatility Clustering of the selected Cryptocurrencies

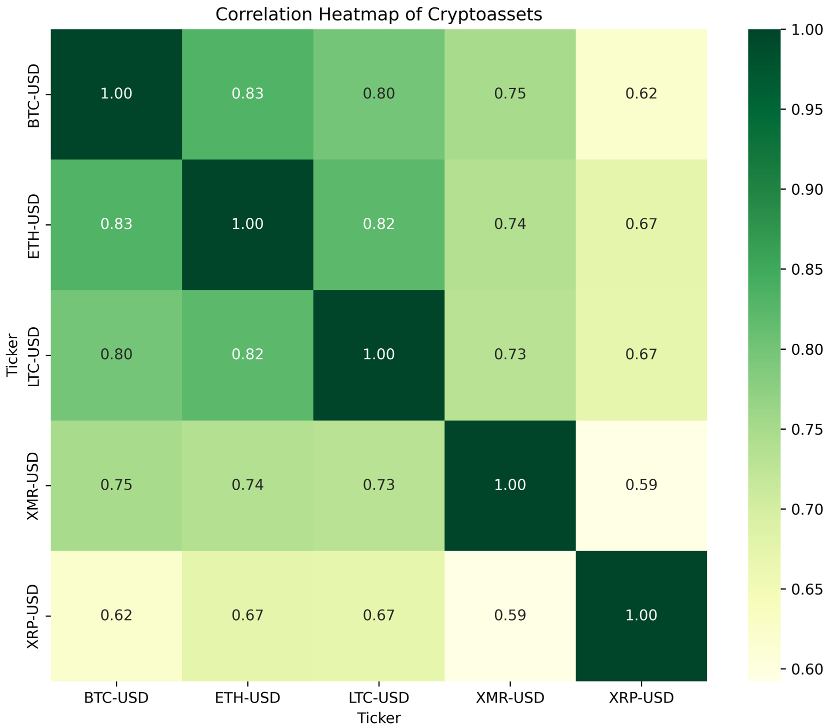


Figure 5.2: Correlation heatmap of the selected Cryptocurrencies

Cryptocurrencies returns tend to have varying distributions, most of them noticeably far from the normal distribution, as seen in Figure 5.3. In order to assess the correlation between the assets in our portfolio, we incorporate this fact of non-normal distributed returns into consideration and employ a non-parametric method through the Kendall Tau correlation matrix, which is suitable for non-normal distributions and provides a measure of the strength and direction of the pairwise association of our assets. Some key characteristics that we want to exploit from the nature of the Kendall Tau matrix are its low sensitivity to outliers, the focus it has on the ordinal association and the accounting it has for the ranks of data rather than the raw value themselves. In Table 5.2, we find the Kendall Tau correlation matrix, which provides some peculiar insights. The most noticeable one is that all cryptocurrencies exhibit a positive correlation, suggesting similar price movements. The strongest pairwise correlation appears to be between Bitcoin (BTC) and Ethereum (ETH), suggesting a strong rank movement association. The weakest pairwise association is found between Monero (XMR) and Ripple (XRP). Additionally, contrary to what the Pearson Correlation matrix in Figure 5.2 shows, Monero (XMR), not Ripple (XRP), exhibits the lowest pairwise correlation with other assets. We also observe that, despite the strong positive correlation among the assets, which limits straightforward diversification opportunities, nuanced dependence structures exist within these assets. These structures suggest the potential for more sophisticated diversification and risk management strategies. This insight reinforces our decision to employ Vine copulas and the GOMEA algorithm, which is designed to uncover and exploit complex dependence patterns for portfolio optimization.

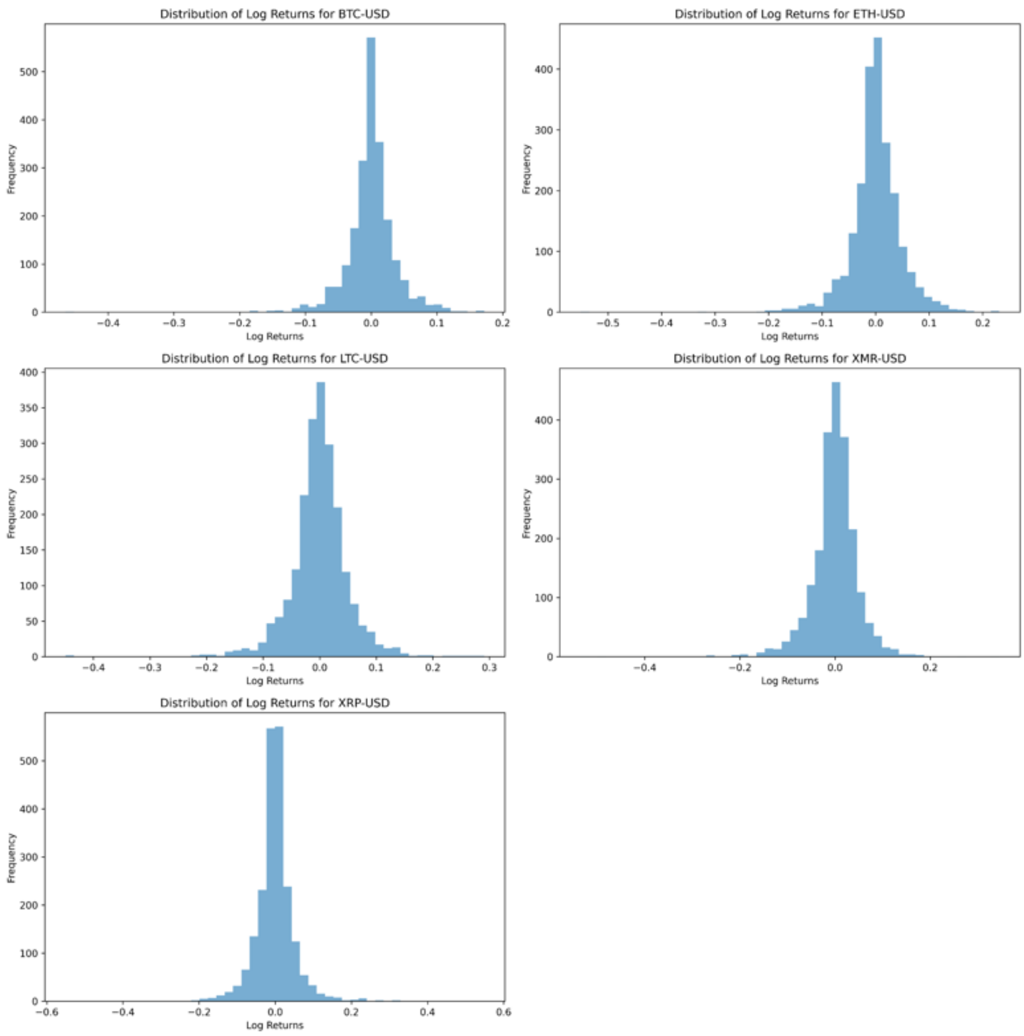


Figure 5.3: Histogram of Logarithmic Returns for selected cryptocurrencies

Ticker	BTC-USD	ETH-USD	LTC-USD	XMR-USD	XRP-USD
BTC-USD	1.000	0.647	0.599	0.526	0.527
ETH-USD	0.647	1.000	0.633	0.510	0.585
LTC-USD	0.599	0.633	1.000	0.502	0.561
XMR-USD	0.526	0.510	0.502	1.000	0.452
XRP-USD	0.527	0.585	0.561	0.452	1.000

Table 5.2: Kendall Tau Correlation Matrix for selected Cryptocurrency Returns

5.3 Hardware and computational cost

For our study, we ran our algorithms using an HP OMEN Laptop 15-ek1xxx, configured with hardware specifications suitable for data analysis. The hardware specifications are the following:

- **GPU**: NVIDIA GeForce RTX 3060 Laptop GPU, equipped with a DCH driver, featuring 3840 CUDA cores, and supporting DirectX Runtime version 12.
- **CPU**: Intel(R) Core(TM) i7-10870H CPU operating at a base frequency of 2.20GHz and capable of reaching up to 2.21 GHz under turbo boost.
- **RAM**: 15.8 GB available
- **Disk Storage**: 661 GB
- **Refresh Rate**: The display offers a 144Hz refresh rate

Additionally, we estimated the computational cost of each algorithm over 30 runs across our two datasets: the simulated and historical returns. In order to determine the computational cost of running our algorithms, we determined the runtime (in seconds), the memory usage (in Megabytes) and the CPU usage (in %) through the use of the functions `virtual_memory` and `cpu_percent` respectively from the `psutil` python library.

The result obtained are the following

Algorithm	Dataset	Runtime (s)	Memory Usage (MB)	CPU Usage (%)
GOMEA	Simulated	900	14249.02	7.02
	Historical	138.98	10493.40	6.68
GA	Simulated	86.75	11443.09	1.98
	Historical	53.43	11821.18	7.58
PSO	Simulated	29.68	11508.98	2.27
	Historical	15.23	11756.48	1.28

Table 5.3: Comparison the algorithms' of Computational Costs

From Table 5.3, we observe the computational costs of the algorithms. For the simulated data, we notice that the PSO is the fastest algorithm among the three, while the GA is the one with the lowest memory and CPU usage. However, the GOMEA algorithm observations must be put into context as we set the termination criteria in 900 seconds; thus, the runtime metric should be interpreted with caution. In terms of memory usage, while GOMEA employs the most, it is not far from the other two algorithms. Additionally, although GOMEA shows a higher CPU usage percentage, it is still quite low when put into context with other optimization algorithms.

Regarding the insights that we can take from the historical data, PSO remains the fastest algorithm, but we notice a significant improvement in GOMEA's runtime, suggesting that the algorithm performs better with real-world data. In terms of memory usage, GOMEA also shows improvement in the historical data as it records the lowest memory usage at 10493.40 MB, while GA increases its usage. Moving to the CPU usage, we observe a slight improvement in GOMEA's performance and an increase in GA's CPU usage while PSO maintains its performance.

6. Results

6.1 Simulated Data

6.1.1 Algorithm performance

As a start to our result section, we examine the performance of the GOMEA algorithm. We focus on two particular aspects: convergence and hyperparameter tuning. We examine the performance of our GOMEA algorithm in terms of the convergence towards an optimized CVaR minimized value. In regards to the hyperparameter tuning, we focus on the two main hyperparameters that exhibited the highest influence on the algorithm's performance: `base_population_size` and `max_number_of_seconds`. Other relevant parameters did not affect the algorithm's performance to the degree seen by the two hyperparameters selected. We proceed to examine them.

6.1.1.1 Convergence of the GOMEA algorithm

In Figure 6.1 the mean objective value for each generation is plotted. In order to provide some visualization of the variability of the data, we plotted the standard deviation bands, indicating one standard deviation from the mean in both directions. Such representation provides insight into the variability of the data points. The graph was constructed based on 30 different runs with the base parameter setup. In terms of the resulting graph, we observe how, as the number of generations increases, the algorithm progresses toward a lower objective value. However, the mentioned progression is not entirely smooth downwards and is faced with certain plateaus, as seen from the 100th generation onwards. Additionally, these plateaus appear to become more frequent and prolonged in subsequent generations. Further examination of the standard deviation bands suggests that the data's behavior maintains a degree of consistency despite the inherent variability. We conducted a one-way ANOVA test in order to determine whether the difference between the group means of the generations is statistically significant. The ANOVA test pointed to a statistically significant difference in the objective values across generations, with an F-statistic of 43.600 and a p-value less than 0.001.

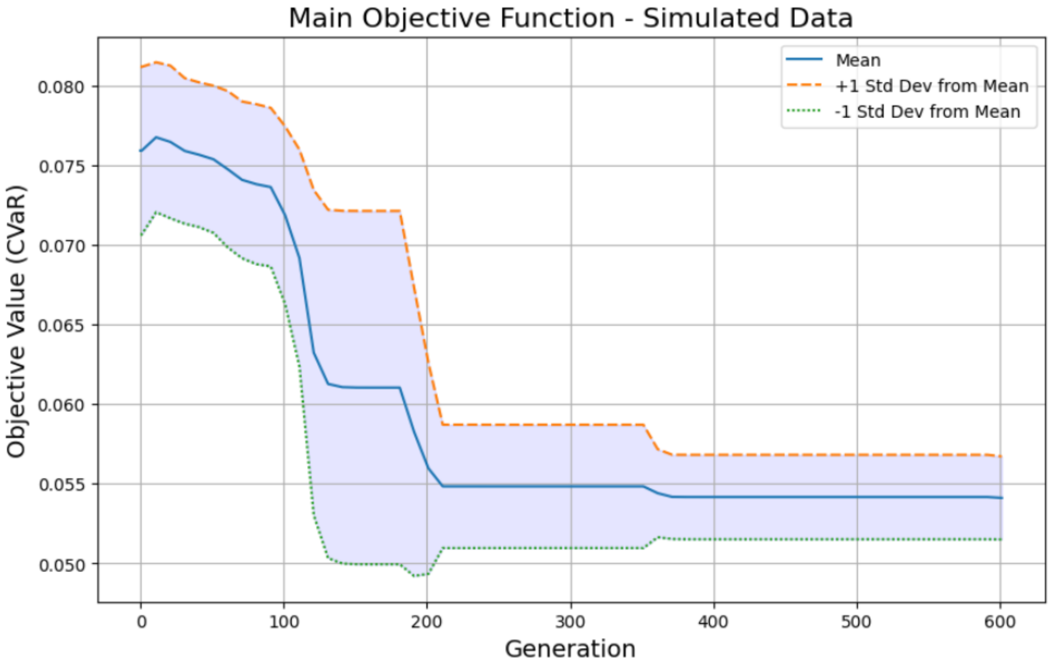


Figure 6.1: Convergence plot of the GOMEA for Simulated Data

6.1.1.2 Hyperparameter tunning

Throughout the experimental trials with the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA), various parameters underwent testing with different values across more than 700 runs with increasing interval steps. Parameters included initial upper and lower ranges, a predetermined maximum number of generations, evaluation limits, the Interleaved Multi-start Scheme (IMS) subgeneration factor, and specific (`value_to_reach`) benchmarks. We discovered that for portfolio optimization, the GOMEA algorithm's performance does not improve with the application of the IMS subgeneration factor, regardless of the base population size being large (10,000) or small (10). We found that consistently, when implementing the IMS subgeneration factor, the algorithm stalled between 6 to 50 generations, showing no improvement in the objective value. Additionally, this approach resulted in homogenous solutions, with resource allocation concentrated on a single asset. As testing advanced, it became clear that the base population size and the time termination criterion (`max_number_of_seconds`) significantly influence the exploration of the solution space and the distribution of asset allocations. In Figure 6.2, we observe the impact of varying the parameter `base_population_size` on the objective value through a graph constructed from 20 runs. The values for `base_population_size` range from 50 to 501, increasing in steps of 25. Initially, the objective value decreases as the base population size increases from 50 to 100. However, there is a sharp increase in the objective value around a base population of 100. The minimum CVaR obtained by the GOMEA algorithm occurs at a base population of 250. Beyond this point, the fitness value rises markedly, with a minor decrease between populations 350 and 400, a trend that continues even as the base population size reaches 500. This graph illustrates the initial 500 values for the parameter, but the observed pattern persists for larger population sizes.

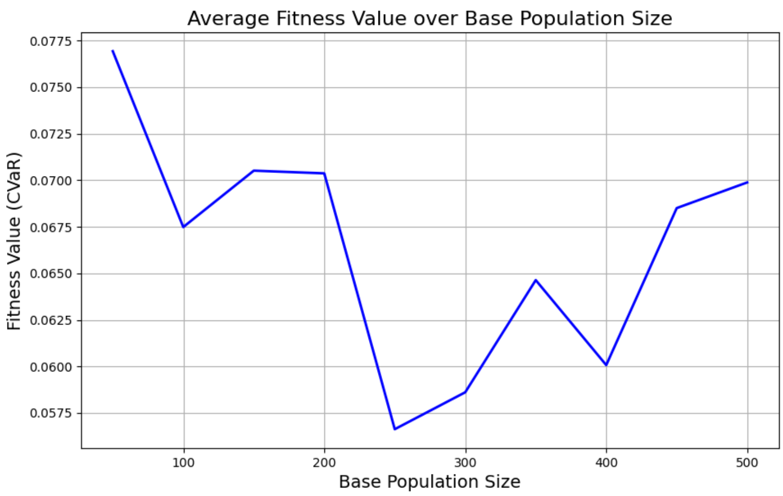


Figure 6.2: Base_population_size - Hyperparameter performance plot for GOMEA

A more challenging hyperparameter to experiment with was the termination parameter `max_number_of_seconds` as seen in Figure 6.3. This parameter required balancing an adequate time for the algorithm to explore the solution space against the risk of premature convergence or remaining trapped in local optima. We primarily experimented with values at the 300, 600, 900, 1200, 1500, 1800 and 2100-seconds marks in time steps of 30 seconds . Notably, the 900-second threshold exhibited a pattern of consistently decreasing lower fitness values. For higher time marks the algorithms did not improved the objective value. It is critical to note, despite occasional signs of the algorithm entrapment in local optima, in most instances it navigated the solution space effectively, avoiding such traps in the search space.

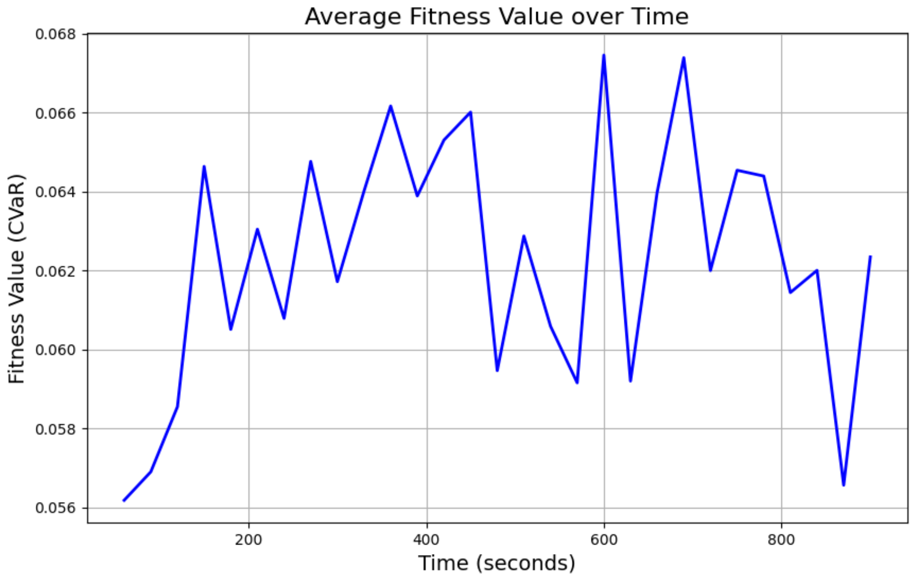


Figure 6.3: Time Termination criteria (seconds) - Hyperparameter performance plot for GOMEA

6.1.2 Results from the simulated data

6.1.2.1 Financial Metrics

The data in Table 6.1, presents the mean weight distribution of each asset per algorithm based on 270 runs. We observe that the three algorithms are inclined to generate diversified portfolios favoring a particular pair of assets, as is seen for GOMEA, which slightly favors Ethereum and Litecoin; GA, which favors Monero and Ripple; and PSO, which has a predominant stake in Ethereum and Litecoin, as well. The table also reflects how we have two core assets that seem to prevail in their weight allocation for all the algorithms, namely Ethereum and Litecoin. In terms of risk appetite, both GOMEA and PSO seem to take a rather conservative stance on highly volatile assets such as Monero and Ripple, while the GA has a higher risk appetite given the allocation of almost 60% of its capital into Monero and Ripple.

Algorithm	BTC	ETH	LTC	XRP	XMR
Gomea	0.2057	0.2112	0.2340	0.1814	0.1677
Genetic Algorithm	0.0896	0.1530	0.1640	0.2958	0.2977
PSO	0.2091	0.2736	0.2890	0.0971	0.1312

Table 6.1: Portfolio mean weight distribution per algorithm - Simulated data

In Table 6.2, we observe the mean portfolio financial metrics obtained from the mentioned 270 runs. Notably, the CVaR values are positive, indicating an optimistic outcome of our optimization process where the worst-case scenarios are seen as minimum expected gains rather than potential losses. This interpretation of CVaR, while uncommon, has been observed in various studies on optimization algorithms that employ copulas for the simulation of returns [10, 12]. Under this less traditional interpretation of the CVaR, the Genetic Algorithm (GA) exhibits the highest minimum potential gain in terms of Conditional Value at Risk (CVaR), suggesting the lowest exposure to tail risk. This is particularly interesting given its focus on highly volatile assets such as Monero and Ripple. Conversely, the GOMEA algorithm demonstrates the lowest minimum potential gain (lowest CVaR value) while achieving a slightly higher expected return than the other algorithms. Given GOMEA's low diversification and high concentration ratios, this could be considered the riskiest strategy among the three. Both GA and PSO are characterized by more diversified portfolios, with GA showing a particularly diverse allocation, yet they appear to have lower risk exposure due to their higher CVaR values. It is also noteworthy that the expected returns are peculiarly similar despite different weight allocations to each asset and varied risk preference profiles among the algorithms.

Another interesting metric to look at is the Sharpe Ratio, which tells us the returns per unit of risk taken. A portfolio with a Sharpe Ratio above 1 is considered good, and below 0 is considered too risky for the expected return. In Table 6.2, despite their different strategies, we observe that the three algorithms obtain high returns by efficiently managing intrinsic volatility, given their high Sharpe Ratio values. There is a caveat that needs to be considered when looking at the Sharpe Ratio in cryptocurrencies, as it is a traditional financial metric that was developed to compare the returns in perspective of the risk-free asset. Given the extremely volatile nature of the cryptocurrencies, regardless of whether they are stablecoins or alt-coins, and the mismeasured high returns seen in the cryptocurrency prices, it is likely to be a financial metric that should not be taken at face value and interpreted with caution.

Metrics	GOMEA	GA	PSOP
CVaR	0.0626	0.0840	0.0753
Expected Return	0.5000	0.4995	0.4995
Diversification Ratio	1.8557	3.7991	2.8466
Concentration Ratio	0.6837	0.2637	0.3628
Sharpe Ratio	1.9402	1.9611	1.9125

Table 6.2: Comparison of Main Financial Metrics - Simulated Data

As the final financial metric, we observe each asset's Marginal Risk contribution (MRC) to its corresponding portfolio. The MRC measures each asset's incremental risk to the portfolio, conditional on its weight and correlation with the other assets. It is a non-linear relationship, which is why the sum of the MRCs does not add up to 1. Analyzing the data presented in Table 6.3, we observe that no distinctive asset across the three portfolios has a major risk contribution, leading to a balanced distribution of risk across the algorithms' strategies. In terms of the highest risk contributor, we notice that Ethereum(ETH) stands out slightly as such. This is not surprising given its role as a core asset in terms of the weight it has in the portfolio, as seen in Table 6.1. We also recognize that Ripple (XRP) has the lowest MRC for all three algorithms. This is consistent with the weight distribution seen in GOMEA and PSOP weight distribution, but not so much with the GA weight assignment of 0.1640 (3rd largest weight assignment in that portfolio). Overall, we recognize that the three algorithms seem to employ a balanced strategy that does not generate any major disparities in risk contribution.

Cryptoasset	GOMEA	GA	PSOP
BTC	0.2629	0.2503	0.2635
ETH	0.2672	0.2595	0.2709
LTC	0.2643	0.2560	0.2681
XRP	0.2375	0.2486	0.2265
XMR	0.2455	0.2553	0.2426

Table 6.3: Marginal Risk Contribution per Asset and Algorithm - Simulated Data

6.1.2.2 Efficient frontier

We then proceed to construct the corresponding efficient frontier for each algorithm and then compare them. The efficient frontier is a set of 10 selected portfolios that provide the highest expected return for a specific level of risk (CVaR). We select the highest-performing portfolios that will allow us to construct an efficient frontier.

First, we examine the efficient frontier of the genetic algorithm and the Particle Swarm Optimization. From Figure 6.4, we observe that the GA constitutes a rather narrow frontier in terms of its search space with its x-axis (representative of the CVaR risk measure) ranging from 0.08378 to 0.08418 and its y-axis (representative of the expected return) ranging from 0.50168 to 0.50193. We observe that the expected return achieves the lowest expected return in its first portfolio with an expected return of 0.50168 for a CVaR level of 0.08377. The highest point in the efficient frontier for the GA is reached through the third portfolio, in which it has an expected return of 0.50194 for a CVaR value of 0.08388, and then it seems to plateau for the remaining 7 portfolios around the expected return value of 0.50186.

The PSO seems to encompass a broader yet still limited area of the solution space, with its expected return ranging from 0.49964 to 0.50055 and its CVaR ranging from 0.06819 to 0.07065. The Efficient Frontier of the PSO seems to exhibit a similar shape as that of the GA, in the sense that it has its most efficient portfolio in the third portfolio, with an expected return of 0.50055 and a CVaR of 0.07065. The remaining seven portfolios show some variability between them rather than a flatter plateau, as seen for the GA.

When we compare the three algorithms, we will examine the weight distribution of the portfolios that confirm the efficient frontier.

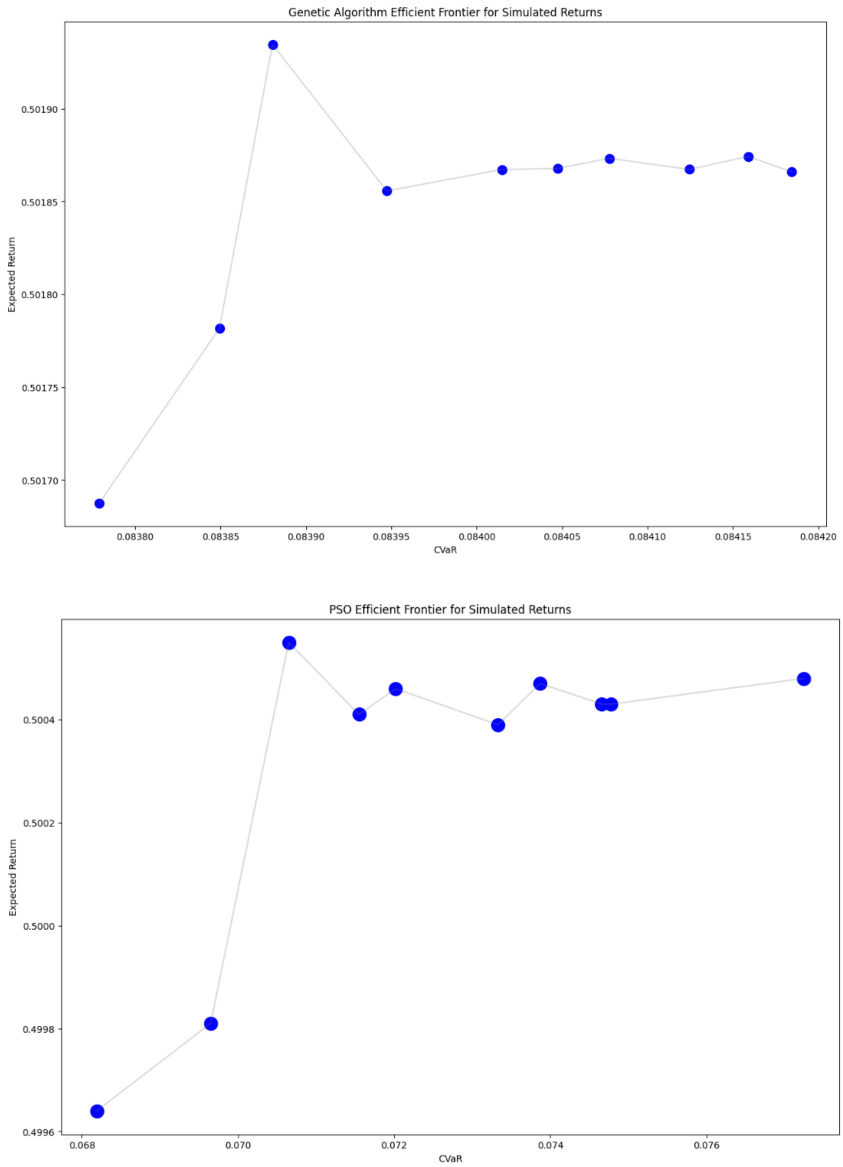


Figure 6.4: Efficient Frontiers for Genetic Algorithm and Particle Swarm Optimization - Simulated Data

In Figure 6.5, we examine the efficient frontier constructed for the GOMEA algorithm. The 10 selected portfolios reveal some intriguing characteristics distinct from those observed with the GA and PSO algorithms. Notably, the efficient frontier for GOMEA covers a broader range in terms of CVaR values and extends higher in the expected returns axis, with the portfolios' expected returns ranging from 0.49611 to 0.50305 and CVaR from 0.05081 to

0.07612. A closer look into the portfolios of the frontier shows an ascending pattern in expected returns up to portfolio 6 (expected return of 0.50263 and CVaR of 0.07612), followed by a slight decline in portfolio 7, and then an increase in both CVaR and expected returns for the last three portfolios. This pattern suggests fundamental differences in the algorithms' internal mechanisms, with portfolio 10 presenting the best performance in terms of the risk-reward trade-off with an expected return of 0.50263 (the second largest) and a CVaR value of 0.07612.

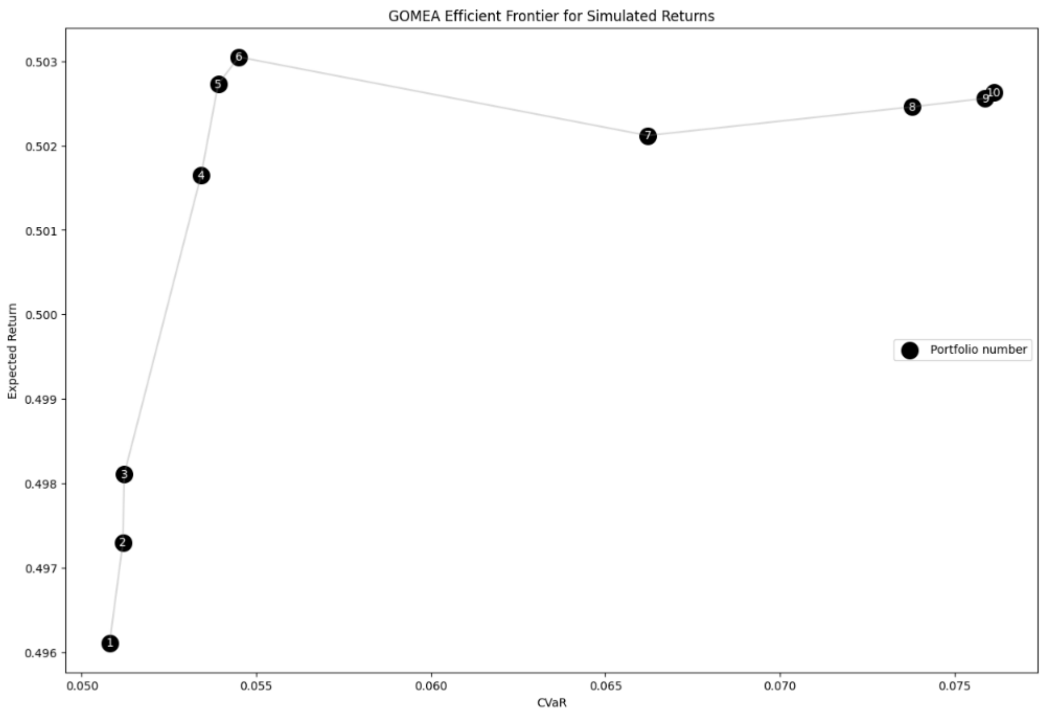


Figure 6.5: GOMEA Efficient Frontier - Simulated Data

To finalize our analysis of the simulated data, we take a look at Figure 6.6, which captures the efficient frontiers of the three algorithms together. In this figure, we can appreciate the different proportions of each frontier and the scalable differences that we find between the GOMEA algorithm and the other narrower frontiers of the PSO and the GA. While the shapes seem to share some resemblance, the dimensions are widely contrasting.

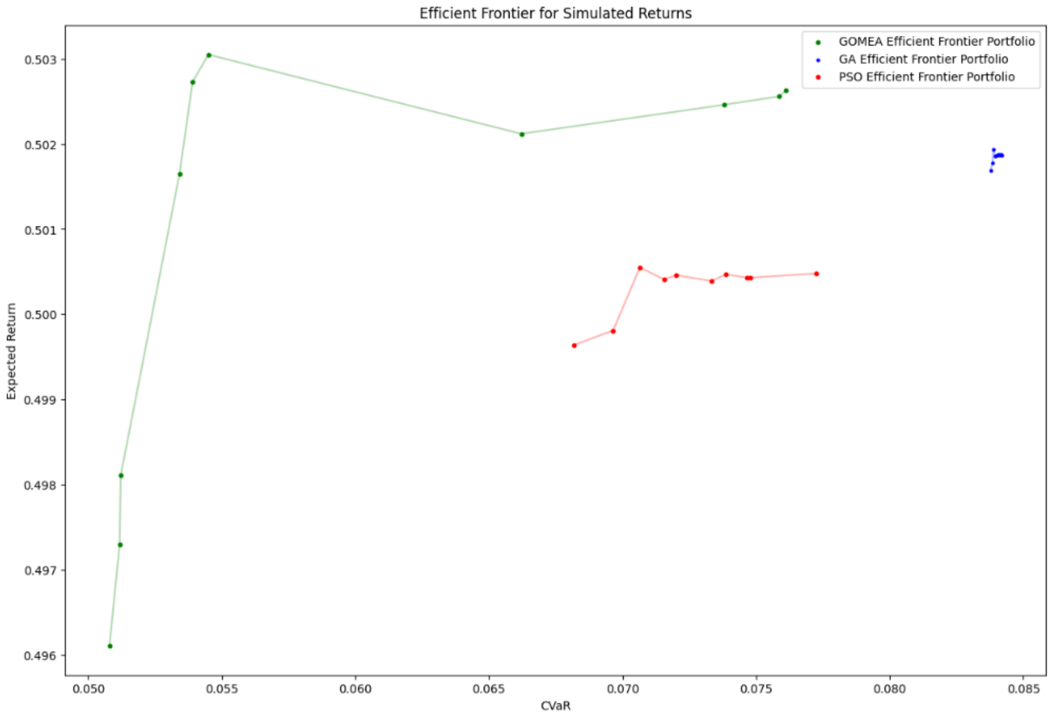
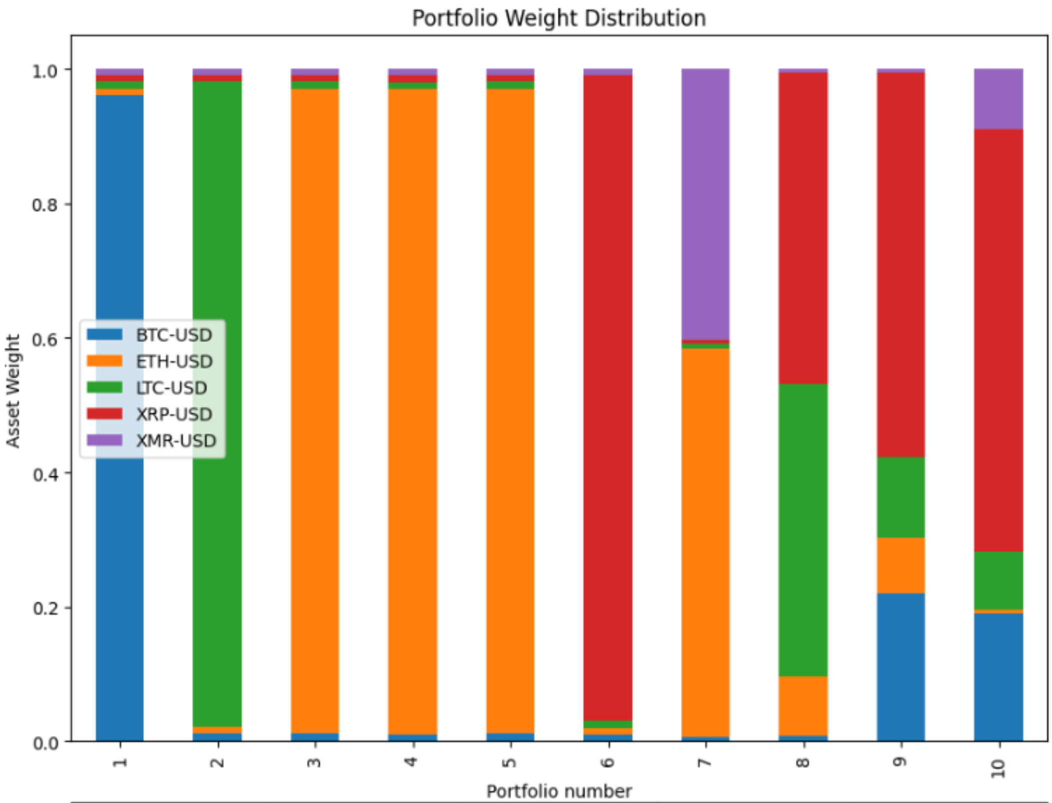


Figure 6.6: Efficient Frontiers Comparison - Simulated Data

In order to obtain a better understanding of the composition of the portfolios that construct the efficient frontiers, we examine the weight distributions of the 30 portfolios found in the 3 efficient frontiers. In Figure 6.8, we visualize the proportional weight distribution of each asset for each algorithm. We notice how the Genetic Algorithm has a balanced distribution of the assets across the different portfolios of its frontier, maintaining a diversified approach. We do see, however, a consistent preference for assets such as Bitcoin (BTC), Ethereum (ETH), and Monero (XMR) and a prevalent low position for Litecoin. On the other hand, the PSO shows a different approach to asset selection. We see portfolios in which there is a high allocation toward single assets, as seen in Portfolio 1 with a prevalent stake in Ethereum (ETH) or Portfolio 4 with a dominant preference for Monero (XMR). Such an approach to focus on single assets that might yield higher expected returns seems to expand the efficient frontier in terms of the CVaR values for each portfolio but does not necessarily provide a significantly wider range of values in the expected return axis. Lastly, we observe the weight distribution of the portfolios for the GOMEA, which takes the approach initially seen in the PSO and amplifies it, which is the preference for strong capital allocation towards a single or pair of assets, and pushes these boundaries further. In some portfolios, we seem to find almost an entire allocation of the capital in pairs of assets, such as is the case in portfolio 1, which is dominated by Bitcoin and Litecoin, respectively, or portfolio 8, which displays this strategy with assets, such as Litecoin and Ripple. Such strategy is detailed further in Figure 6.7, where we observe the weight distribution's granular detail.



	BTC-USD	ETH-USD	LTC-USD	XRP-USD	XMR-USD
1	0.96086	0.00971	0.00992	0.00982	0.00971
2	0.01209	0.00977	0.95877	0.00968	0.00968
3	0.01108	0.95901	0.0105	0.00969	0.00972
4	0.0097	0.96072	0.0097	0.01017	0.0097
5	0.01158	0.95935	0.00969	0.00969	0.00969
6	0.0097	0.01029	0.0097	0.9606	0.0097
7	0.00584	0.57854	0.00623	0.00584	0.40354
8	0.00781	0.08854	0.4344	0.46455	0.00469
9	0.21992	0.08279	0.12056	0.57097	0.00577
10	0.19019	0.00634	0.08478	0.62792	0.09077

Figure 6.7: GOMEA Efficient Frontier Portfolio's Weight distribution detailed - Simulated Data

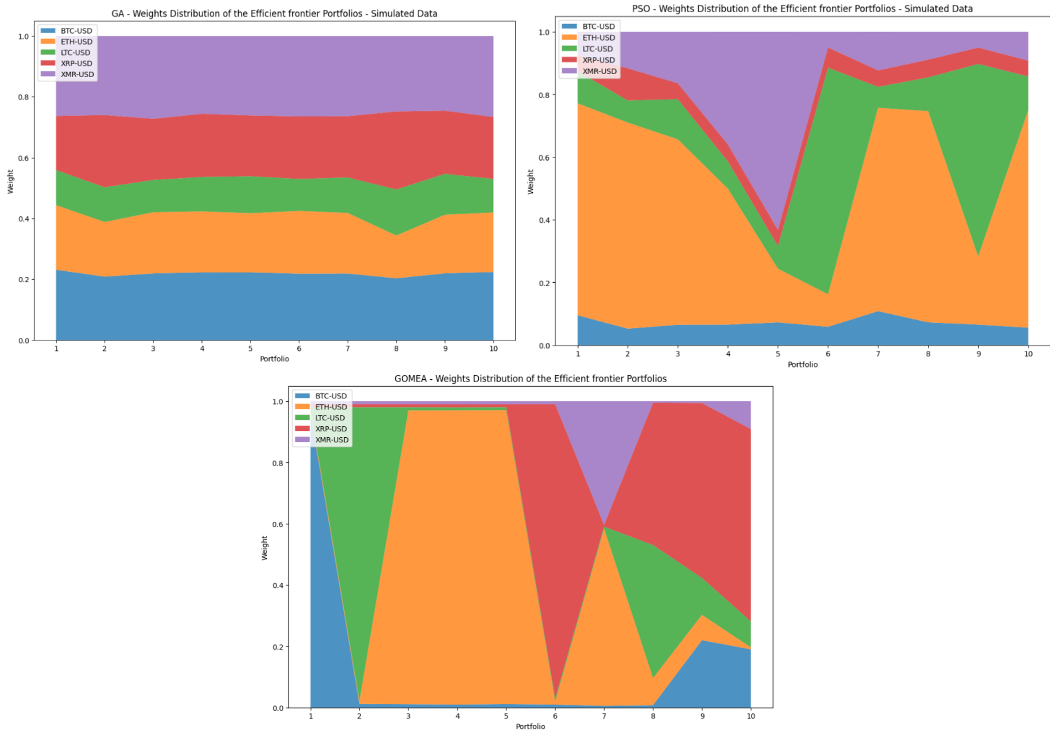


Figure 6.8: Weight Distribution of the Portfolios in the Efficient Frontier for GA, PSO, and GOMEA - Simulated Data

6.2 Out of sample test - Historical Data

As part of the backtesting process, we applied our three algorithms to historical data, which was used to generate the simulated data through Copulas. This approach allowed us to present the algorithms with out-of-sample data and observe whether the asset allocation strategies observed in the simulated data remain consistent.

6.2.1 Financial Metrics

We start analyzing the results of our algorithms on historical data by observing the mean values obtained from the new 270 runs of each algorithm across different parameters.

In terms of the mean weight value obtained for the weights of the assets across the three different algorithms, we observe some changes in the strategies in Table 6.4. GOMEA and GA seem to have shifted their strategy towards focusing on Bitcoin (BTC), with GA allocating almost 64% of its capital to this cryptocurrency and GOMEA making BTC the main asset in its portfolio. On the other hand, PSO maintains its focus on Ethereum and Litecoin, while shifting the weight previously assigned to Bitcoin towards more volatile cryptocurrencies like Ripple and Monero. Similar to what we found with the simulated data, GOMEA and PSO maintain the same strategy and exhibit a rather diversified approach focusing on asset pairs. The allocation towards Monero (XMR) is particularly interesting; it maintains almost the same capital allocation from all the algorithms except GA, which reduces its stake from 29% to 14%. PSO also departs in its allocation of stablecoins (BTC,ETH and LTC) and provides a more prominent role to the alt-coins by investing more in Ripple (XRP). Nonetheless, GOMEA maintains some similarities in its allocation strategy but with a clear shift from Ethereum towards Bitcoin.

Algorithm	BTC	ETH	LTC	XRP	XMR
Gomea	0.24196	0.16337	0.23854	0.19558	0.16056
Genetic Algorithm	0.64387	0.05422	0.07079	0.08733	0.14379
PSO	0.07898	0.26029	0.30398	0.16715	0.18960

Table 6.4: Portfolio mean weight distribution per algorithm - Historical data

We examine then the main financial metrics in Table 6.5. The first thing that jumps to our attention is that the CVaR metric of the three algorithms has shifted toward negative values, a behavior expected for historical returns, given that we are now dealing with portfolio losses. We observe that the Genetic Algorithm maintains its role as the algorithm with the lowest risk exposure given with a CVaR value of -0.059165. PSO now becomes the algorithm with the highest risk exposure with a CVaR value of -0.071919 and GOMEA ranks in the middle with a CVaR value of -0.069259. The mean expected return of the GA and PSO diminishes significantly from 0.4995 in the simulated data to 0.000167 and -0.000193 in the historical data. The diversification and concentration ratios also change. We observe a more diversified approach from GOMEA and PSO and a more concentrated approach from GA, which tells us that the strategy has shifted mainly for the GA. Moving on to the last metric, we find the most relevant changes. The Sharpe Ratio becomes negative for both the GOMEA and PSO, signaling some interesting dynamics in the risk-return relationship that the strategies carry. The Sharpe ratio is considerably reduced for the GA but remains positive. Overall, we see a consistent performance for the GOMEA algorithm, but we can notice some indication of an average approach towards a more diversified approach.

Metrics	GOMEA	GA	PSOP
CVaR	-0.069259	-0.059165	-0.071919
Expected Return	0.060180	0.000167	-0.000193
Diversification Ratio	2.854477	2.208908	3.355676
Concentration Ratio	0.500901	0.461796	0.307673
Sharpe Ratio	-0.002731	0.004353	-0.004355

Table 6.5: Comparison of Main Financial Metrics - Historical Data

The final metric that we examine is presented in Table 6.6. This table suggests that each cryptocurrency contributes less risk to the portfolio than was initially anticipated based on simulated data. These results are not surprising, considering the increase in diversification seen in Table 6.5, which leads us to extrapolate that diversification has been enforced to such a degree that the individual risk per asset has been significantly minimized. The most important takeaway from this table is the understanding that real-world market conditions may reduce the individual risk contributions of assets more effectively than what is observed in simulated data. Additionally, the low scores in Table 6.6 provide insight into the robustness of the optimization algorithms deployed.

Cryptoasset	GOMEA	GA	PSOP
BTC	0.033401	0.035808	0.032165
ETH	0.043643	0.042370	0.044142
LTC	0.046833	0.044144	0.047312
XRP	0.043198	0.041048	0.042135
XMR	0.045955	0.044309	0.046866

Table 6.6: Marginal Risk Contribution per Asset and Algorithm - Historical Data

6.2.2 Construction of the efficient frontier

Figure 6.9 displays the efficient frontiers for each of the three algorithms constructed from historical returns. We observe some similarities with the frontier obtained from simulated data, as shown in Figure 6.6. Firstly, we notice that the efficient frontier exhibits a closer scale or range, suggesting a particular convergence in their dimensions. The GA features a narrow, efficient frontier, mirroring its representation in the simulated data. Consistent with the simulated data, the efficient frontier of the GA also demonstrates a higher expected return and a higher CVaR score than that of the PSO. Nonetheless, the PSO presents a broader dimension than observed in the simulated data, to the extent of closely approaching the first four portfolios of the GOMEA algorithm. The efficient frontier of GOMEA remains the largest and broadest among the three, albeit with fewer portfolios that outperform those of the other two algorithms

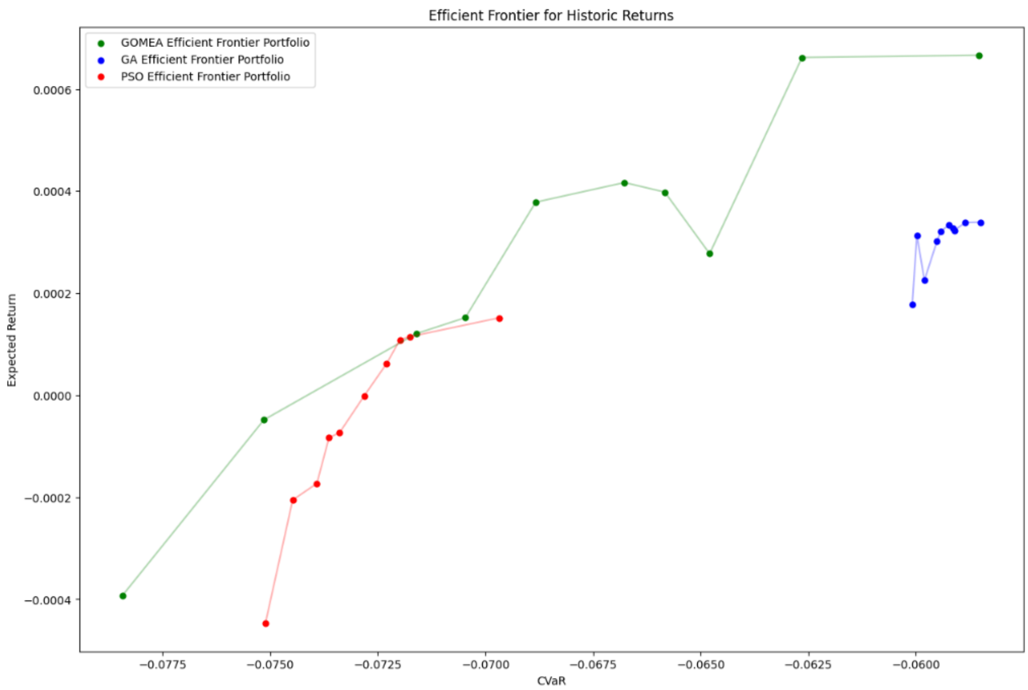


Figure 6.9: Efficient Frontiers Comparison - Historical Data

6.2.3 Weight distribution of the efficient frontier

The efficient frontier presented in Figure 6.9 is constructed by the portfolios presented in Figure 6.10. From Figure 6.10 we notice that the weight distribution patterns remain the same for PSO and GOMEA in comparison to what we found in Figure 6.8. For both algorithms, we see a prevailing tendency to select a single or a pair of predominant assets for which most of the capital is allocated. The GA is the only algorithm that seems to have undergone a completely different weight allocation strategy. Even though the parameters were kept the same, in terms of mutation rate and elitism, the GA algorithm allocated most of its capital into Bitcoin as its main asset to maintain an overarching distributed allocation of capital between the remaining 4 assets, shifting away from the clear diversified strategy seen in Figure 6.8. If we examine the GOMEA weight distribution in detail, as Figure 6.11 shows, we notice three prevalent cryptocurrencies across the three portfolios: Bitcoin, Litecoin, and Ethereum. Monero has the lowest allocation across the portfolios except for portfolio 4 in which it has an allocation of almost 21%.

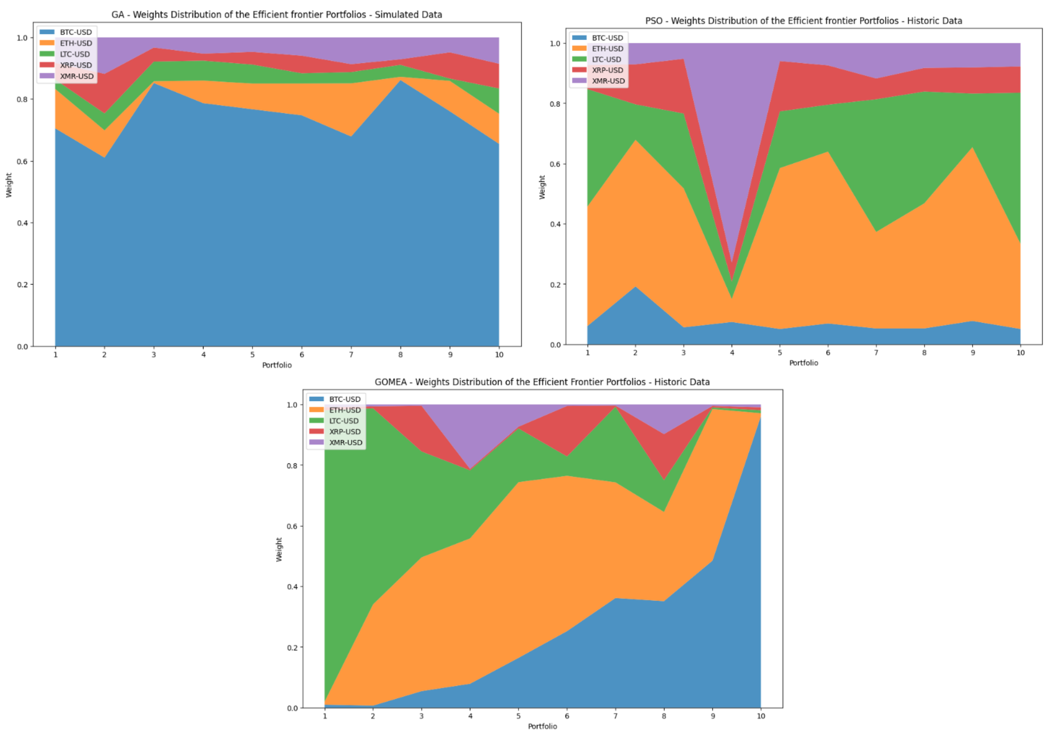
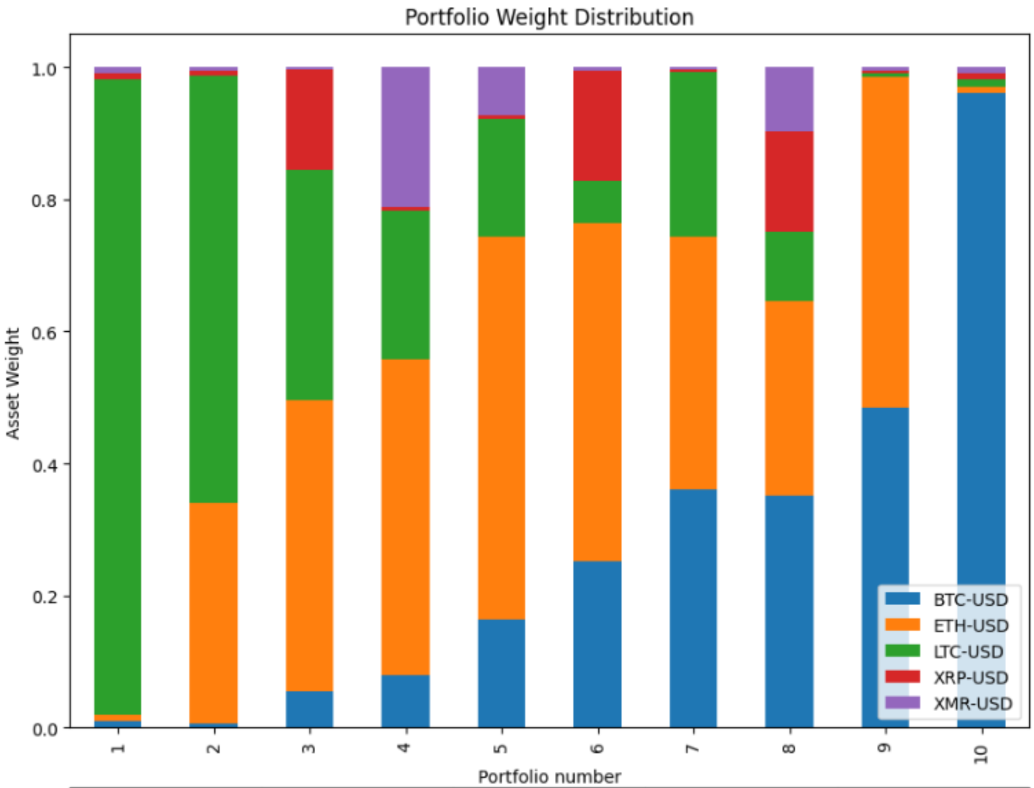


Figure 6.10: Weight Distribution of the Portfolios in the Efficient Frontier for GA, PSO, and GOMEA - Historical Data

In contrast to what we saw in the simulated data, Bitcoin seems to have gained a more relevant role in the construction of efficient frontier portfolios across all three algorithms. Ethereum maintains its influential presence in all portfolios, akin to the simulated scenario. Litecoin sees an increased prominence in the GOMEA algorithm and sustains its allocation for GA and PSO. Ripple gains traction within the GOMEA frontier but experiences a reduced share in GA, with funds reallocated towards Bitcoin. Lastly, Monero holds a minor yet consistent position across all algorithm portfolios, with PSO assigning a notable allocation to this asset in certain portfolios.



	BTC-USD	ETH-USD	LTC-USD	XRP-USD	XMR-USD
1	0.00971	0.00971	0.96117	0.00971	0.00971
2	0.00656	0.334	0.64637	0.00653	0.00653
3	0.05434	0.44117	0.34925	0.15077	0.00446
4	0.07866	0.47928	0.22458	0.00484	0.21264
5	0.16407	0.57947	0.17708	0.00585	0.07352
6	0.25216	0.51227	0.06382	0.16657	0.00517
7	0.36144	0.38153	0.24931	0.00385	0.00386
8	0.35109	0.29417	0.10491	0.15188	0.09795
9	0.48451	0.50027	0.00508	0.00505	0.00509
10	0.96117	0.00971	0.00971	0.00971	0.00971

Figure 6.11: Efficient Frontiers Comparison - Historical Data

7. Discussion

In this section, we will discuss what we consider the most relevant findings from the results section and provide an answer to our research question.

7.1 On the Algorithm performance

7.1.1 On the algorithm convergence

As seen in our result section, we notice that our GOMEA algorithms converge towards a good solution as the generations advance. Of particular interest for our discussion is to analyze two particular behaviors that we observe in Figure 6.1. The first is a rapid decrease in the objective function value within the first 120 generations and the second is a plateauing behavior that becomes more pronounced as the generations go by. We first need to consider our algorithm's parameter selection as it is influential to the interpretation. We run our GOMEA algorithm with no IMS scheme, a base population size of 250 and a full linkage model. This selection implies that we are assuming all variables are treated as interdependent, as indicated by the full linkage model, and that there is a more homogenous exploration process due to the lack of IMS. Additionally, a diverse genetic material is available to be used as individual solutions in the population, with a base population of 250.

An interpretation of why we observe a quick optimization in the first 120 generations can be attributed to the significant role played by the initial setup of the base parameters. Given the available 250 solutions, the GOMEA algorithm efficiently navigates the solution space, capitalizing on the interdependencies between variables to generate improved solutions. A key factor during the initial generations is the vast and varied genetic pool from which the GOM operator selects the beneficial traits of the variables found in the solution pool [23]. This genetic diversity enables the GOMEA algorithm to broadly explore different regions of the solution space, benefiting from the many potential initial combinations at its disposal. Consequently, the algorithm can rapidly identify and integrate advantageous variable combinations, facilitating swift initial optimization behavior.

The plateaus observed beyond the first 120 generations suggest that the algorithm progressively struggles to find new and better solutions, leading to the exhaustion of the available genetic variation. Given our current parameter setup, the absence of IMS intervention in generating variation might hinder the Gene-pool Optimal Mixing (GOM) operator's ability to create superior offspring in subsequent generations. Furthermore, as the generations advance, it is likely that we are dealing with a trend in which only the fitter individuals are selected and combined due to the homogeneous approach to exploration. Additionally, the decision to employ the full linkage model implies that any change to a variable needs to be considered in light of its effect on all other variables, given their assumed interdependencies. This comprehensive approach to treating the entire genome as a single, interconnected block (due to the interdependencies) may cause the GOM operator to overlook improvements in isolated variables. Ultimately, the operator focuses on its collective impact on the genome, limiting the search space exploration.

7.1.2 On the performance with the hyperparameters tunning

The design decision to forego the Interleaved Multi-start Scheme (IMS) in our GOMEA implementation significantly impacts the functionality of the `base_population_size` parameter. As explained by Bouter and Bosman [25], in their paper on the GOMEA library, `base_population_size` becomes the sole determinant of the algorithm's initial search space size when IMS is inactive. This makes it a critical factor in how widely and diversely GOMEA explores the search space. Moreover, Bouter and Bosman highlight that the use of IMS mitigates the complex and often laborious process of fine-tuning the population size factor, which has a massive impact on the performance of the algorithm. Dushatskiy et al. detail this by labeling it as problem-dependent and highly complex, especially for evolutionary algorithms such as GOMEA, whose linkage model requires enough solutions in its population to effectively learn the linkages between data points in an accurate enough manner that allows the algorithm to function properly [97].

This impact is evident in Figure 6.2, where the sensitivity of the objective function to various base population sizes is portrayed. The graph demonstrates that a `base_population_size` of 250 is the most efficient for minimizing the objective function, with both higher and lower values for the parameter yielding less favorable results. Ultimately, we are dealing with a trade-off in which we want to make use of a value that does not take too many computational resources, as is the case with high values for the base population, which, in our case, performs worse than the selected value, and a value that gives us the best performance for our problem structure. The selected value of 250 proves to be the optimal choice that balances our problem's structure and computational efficiency considerations

7.2 Interpretation of the results

7.2.1 Uncovering the insights from the financial metrics

When examining Table 6.5 for the financial metrics of the historical returns, we get an opportunity to discover how the use of historical data tests our algorithms. We first notice an increase in exposure to extreme adverse market conditions (tail risk), which is revealed by the negative CVaR values. Within the same table, we examine the expected returns and notice that our results are very different than those generated from the simulated set. However, these results are closer to what we expect from volatile assets such as cryptocurrencies.

In terms of what the Sharpe ratio tells us, we see an interesting dynamic taking place. While we see an increase in the Diversification Ratio and a reduction in the Concentration Ratio for both the GOMEA and the PSO, their Sharpe Ratio takes a noticeable decrease, which seems counterintuitive to the common financial practices that state that higher diversification tends to mitigate risk [98]. Nonetheless, in the context of cryptocurrencies, diversification may not always lead to reduced risk due to the high volatility and correlation dynamics within the cryptocurrencies [99]. For the mentioned type of assets, it is likely that diversification may not mitigate the inherent risk as effectively as it might in more traditional asset classes.

Upon further examination of Table 6.5 we acknowledge particular dynamics across the three optimization algorithms. The GA seems to provide an efficient risk management strategy with minimal returns, which, accompanied by its low-risk exposure to tail risk (low CVaR value), points towards a risk-averse optimization strategy. The PSO displays the opposite. The negative expected return paired with a negative Sharpe Ratio suggests that the strategy employed by this algorithm generates a portfolio that is not only expected to lose value but also does so by taking an unnecessary risk position on it. Such interpretation is reinforced by the fact that the PSO has the greatest exposure to tail risk, as its low CVaR value suggests. It is particularly interesting to look at GOMEA's case, in which we see the highest positive expected return across the three algorithms but with a negative Sharpe ratio. This dynamic suggests that the portfolio's risk levels are disproportionally high given its risk-adjusted returns, i.e., the returns do not compensate for the amount of risk that is being taken. It is worth mentioning that we also need to consider the possibility that the Sharpe Ratio is not a proper measurement of the risk/return tradeoff for cryptocurrencies, given its theoretical construction based on traditional assets' dynamics and behaviors. This is a potential avenue of investigation for further studies.

7.2.2 Interpretation from the efficient frontiers portfolios

We begin by analyzing the shape of our efficient frontiers, with a particular focus on GOMEA's performance. The spread of an efficient frontier across the CVaR values (x-axis) indicates a broad range of portfolio choices, providing investors with diverse options that can align with their risk profiles. GOMEA's ability to offer a wide set of choices in terms of CVaR levels suggests that the algorithm can successfully accommodate different risk levels while maintaining efficient portfolios in terms of returns. Conversely, a narrower spread across the x-axis, as seen by the GA and the PSO, implies a greater concentration at certain risk levels, ultimately limiting the availability of options to investors in terms of risk profiles. The extent of the spread on the y-axis (expected returns) is also worth analyzing. An algorithm that stretches vertically is capable of achieving varying performance potentials across different risk profiles, whereas a limited stretch indicates constrained potential returns. GOMEA's efficient frontier exhibits such extensiveness in the y-axis that, paired with the wideness of its efficient frontier, speaks of the algorithms' capacity to encompass various risk/return spectra.

An ascending efficient frontier, characterized by initial portfolios that exhibit low risk and low returns, with subsequent portfolios that progressively increase their values along both axes, highlights the capabilities of an algorithm to decrease risk while increasing expected returns effectively. Such a shape suggests a positive risk/return trade-off where less risky portfolios yield higher expected returns. We observe such shape in GOMEA's efficient frontier, which tells us that it is not only a well-performing portfolio risk management tool but also aligns with traditional investment goals. In the context of simulated data, in which CVaR is treated as a measure of minimum expected gains, portfolios positioned further to the right on the efficient frontier show higher minimum gains in their worst-case scenarios. This progression suggests that as portfolios move towards more optimistic scenarios (further right), they tend to generate higher overall returns. GOMEA's efficient frontier exhibits such behavior, which pinpoints the algorithm's ability to leverage optimistic forecasts and translate them into higher financial returns. This behavior also stresses GOMEA's adaptability to various market conditions, maintaining efficient allocation strategies through effective risk/return management.

The fact that we observe similar shapes in both the simulated (where CVaR is interpreted as minimum expected gain) and historical data (where CVaR represents potential losses), suggests that GOMEA is a robust algorithm for this optimization problem. Moreover, such robustness can be interpreted as indicative of an algorithm that can adapt its strategies to different applications of risk, proving its utility across a spectrum of investment scenarios and financial optimization problems.

Delving into the composition of the portfolios that constitute the efficient frontiers of the three algorithms, as seen in Figures 6.8 and 6.10, we note that both GOMEA and PSO maintain their asset allocation strategies, while the GA introduces a change. When presented with historical data, the GA opts for a higher allocation of capital towards BTC. The GA's strategy could be seen as a logical consequence of its intrinsic preference for stability and an optimized risk/return balance, as evidenced by BTC's superior historical performance with the highest expected return, lowest volatility, and most favorable Sharpe Ratio among the assets in the portfolio as seen in Table 5.1. The GOMEA algorithm retains its allocation strategy when presented with historical data due to its inherent linkage learning mechanism, which preserves and exploits beneficial genetic structures. This leads to a gradual improvement of solutions rather than abrupt strategic shifts, ensuring consistency in asset allocation regardless of the underlying dataset. The stability of the PSO algorithm is influenced by mechanisms that ensure the boundedness of errors, suggestive of a social learning dynamic where the collective behavior of the swarm tempers any individual particle's tendency to change the overall solution direction of the swarm drastically [100]. In a sense, the PSO algorithm's strategy is to evolve in a measured and stable manner, regardless of the data presented.

7.3 Addressing the research question

Up to this point, we have constructed the frameworks and conducted the corresponding analysis on our data in order to answer our research question: Is GOMEA a better algorithm than GA and PSO for portfolio risk management through cryptocurrency portfolio optimization? We answer this question from theoretical, empirical, and interpretational perspectives.

We believe that GOMEA has a theoretical advantage over the other two EAs. This advantage arises from its linkage learning ability, which allows it to identify and exploit the underlying intricate relationships between the crypto assets, which ultimately allows it to optimize the lower tail of the portfolio loss distribution represented through the CVaR metric. The dynamic nature of linkage learning enables GOMEA to granularly decipher and exploit characteristics existent in cryptocurrencies, such as inverse leverage [39] or positive asymmetric behaviours [37]. Moreover, the GOM operator has an advantage over other EAs mechanisms that employ crossover and mutation. Algorithms like GA tend to disrupt promising genetic structures as part of their random process of splitting and recombining chromosomes. The GOM operator prioritizes enhancing beneficial genetic material during the mixing process, preventing any disruption of effective found structures. So, from a theoretical perspective, we expect the GOMEA algorithm to perform better as a portfolio risk management algorithm.

The evidence found in our result section shows that the GA generates better CVaR and Sharpe Ratio values for both the historical and simulated data. Nonetheless, GOMEA does produce strong values for both metrics across the different datasets, even outperforming the PSO in both datasets. Regarding the efficient frontiers, GOMEA consistently exhibits wider and taller efficient frontiers than those generated by GA and PSO. This indicates that it can accommodate different risk profiles and, in most cases, offer higher expected returns for the same levels of risk.

This leads us to our interpretation. Although GOMEA did not perform as well as GA in minimizing CVaR in both datasets, it outperformed both GA and PSO regarding overall portfolio risk management. This is evidenced by the fact that GOMEA not only has robust portfolio optimization performance but also exhibits certain traits expected from an effective portfolio risk management tool. These particular traits are GOMEA's capacity to achieve higher returns across different risk levels, incorporate a variety of risk profiles, effectively decrease risk while ensuring higher expected returns, maintain consistency in results, and adapt strategies across different interpretations of risk. Particularly, such adaptability suggests a strong capacity for adaptability to different risk applications.

Therefore, we conclude that GOMEA offers a greater alignment with the core principles of what can be considered a more effective portfolio risk management tool. It exhibits potent capabilities in risk mitigation through the targeted optimization of the CVaR metric and surpasses the other algorithms in balancing the risk/return trade-off. For these reasons, we believe that GOMEA is a superior algorithm in portfolio risk management through cryptocurrency portfolio optimization of the CVaR metric.

7.4 Limitations of our study

Our study has certain limitations that require further exploration. The first limitation concerns asset selection. While the justification for the cryptocurrencies chosen for our portfolio is provided in Chapter 2, it is crucial to test whether the results concerning GOMEA's performance, compared to the other two algorithms, remain consistent across portfolios formed from different assets across different time windows than those selected within our study. This is a potential area for future research as it could provide additional insights into the performance of GOMEA.

Another limitation relates to the impact of market anomalies, often referred to as "black swans" in financial terminology, on our algorithm's performance. Although we simulate extreme market conditions using copulas to generate synthetic data, it is important to acknowledge that while copulas are useful, they do not provide a comprehensive solution to address all possible anomalies. In this sense, extensive real-world data is always the best source. Moreover, our study incorporates external economic factors such as regulatory hearings, the COVID-19 pandemic, Brexit, and the onset of the Russo-Ukrainian War into our historical dataset. However, we suggest that future research should also examine the influence of additional external factors, such as new regulatory measures on cryptocurrencies.

One additional limitation of our study that could affect the generalizability of our findings is the selection and tuning of parameters. Although employing copulas for model evaluation and historical data for out-of-sample backtesting provides valuable insights into the GOMEA algorithm's performance under unexpected market conditions, it is crucial to acknowledge potential unexplored parameter configurations that might enhance the outcomes. Specifically, the utilization of the IMS function and a different linkage model could enable GOMEA to discover more effective exploration and exploitation mechanisms, thereby improving its performance. Moreover, while CVaR is a commonly used metric for evaluating portfolio performance, exploring additional portfolio optimization problems, as discussed in section 3.2, could further clarify GOMEA's broader applicability.

Although our analysis is based on static data, providing a theoretical foundation and initial insights into GOMEA's performance, we recognize the importance of incorporating operational dynamics into our analysis. These dynamics include transaction costs, short selling, liquidity constraints, portfolio rebalancing frequency, and other relevant elements of actual trading environments. These elements are crucial for testing GOMEA's potential as an algorithmic trading alternative. Exploring these factors in future research would enhance its relevance for real-world portfolio risk management, further narrowing the gap between theoretical research and practical application of GOMEA in the dynamic cryptocurrency markets.

8. Conclusion

We initiated this study to determine whether the GOMEA algorithm outperforms other evolutionary algorithms, specifically the Genetic Algorithm and Particle Swarm Optimization, in portfolio risk management through cryptocurrency portfolio optimization. The theoretical framework constructed in our study suggests that GOMEA is better suited for this task through its variation operator and linkage learning capability, given its capacity to exploit certain inherent characteristics of cryptocurrencies, such as their high volatility and complex interdependence structures. Our empirical findings support our assumption, showing that the GOMEA algorithm not only performs strongly in the optimization process and during the construction of the efficient frontier but also exhibits additional beneficial traits that align closely with what is expected from effective portfolio risk management tools.

However, it is important to acknowledge that our study has certain limitations that should be addressed in future studies in order to enhance the robustness of our findings and promote the integration of GOMEA into mainstream financial practices and literature. Ultimately, our study establishes a foundation for the broader application of the GOMEA algorithm in financial practices, particularly in portfolio optimization and portfolio risk management as it demonstrates how GOMEA's capabilities can effectively address key challenges faced by portfolio managers.

This study contributes significantly to the evolving literature on cryptocurrency portfolio optimization using evolutionary algorithms. It underscores the need for enhanced algorithmic tools capable of navigating the complex dynamics of cryptocurrency markets.

Bibliography

[1] Satoshi Nakamoto. *Bitcoin. A peer-to-peer electronic cash system*. 2008.

[2] Harry Markowitz. "Portfolio Selection". In: *The Journal of Finance* 7 (1 1952), pp. 77–91.

[3] P. A. Samuelson. "The Fundamental Approximation Theorem of Portfolio Analysis in terms of Means, Variances and Higher Moments". In: *The Review of Economic Studies* 37 (4 Oct. 1970), pp. 537–542. ISSN: 0034-6527. DOI: 10.2307/2296483.

[4] Hiroshi Konno and Hiroaki Yamazaki. "Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market". In: *Management Science* 37 (5 May 1991), pp. 519–531. ISSN: 0025-1909. DOI: 10.1287/mnsc.37.5.519.

[5] Martin R. Young. "A Minimax Portfolio Selection Rule with Linear Programming Solution". In: *Management Science* 44 (5 May 1998), pp. 673–683. ISSN: 0025-1909. DOI: 10.1287/mnsc.44.5.673.

[6] Jorian. *Value at risk : the new benchmark for controlling market risk*. McGraw-Hill, 1997.

[7] R Tyrrell Rockafellar and Stanislav Uryasev. "Optimization of Conditional Value-at-Risk". In: *Journal of risk* 2 (2000), pp. 21–42.

[8] Farzan Soleymani and Eric Paquet. "Deep graph convolutional reinforcement learning for financial portfolio management – DeepPocket". In: *Expert Systems with Applications* 182 (Nov. 2021). DOI: https://doi.org/10.1016/j.eswa.2021.115127.

[9] Gah Yi Ban, Noureddine El Karoui, and Andrew E.B. Lim. "Machine learning and portfolio optimization". In: *Management Science* 64 (3 Mar. 2018), pp. 1136–1154. ISSN: 15265501. DOI: 10.1287/mnsc.2016.2644.

[10] Tianxiang Cui et al. "Portfolio constructions in cryptocurrency market: A CVaR-based deep reinforcement learning approach". In: *Economic Modelling* 119 (Feb. 2023). ISSN: 02649993. DOI: 10.1016/j.econmod.2022.106078.

[11] Richard S. and Andrew G. Barto Sutton. *Reinforcement learning: An introduction*. MIT press, 2018.

[12] Jules Clément Mba and Magdaline Mbong Mai. "A Particle Swarm Optimization Copula-Based Approach with Application to Cryptocurrency Portfolio Optimisation". In: *Journal of Risk Financial Manag* 15 (7 June 2022). DOI: https://doi.org/10.3390/jrfm15070285.

[13] Z. Guennoun, M. El Hachloufi, and F. Hamza. "Stocks Portfolio Optimization Using Classification and Genetic Algorithms". In: *Applied Mathematical Sciences* 6.94 (2012), pp. 4673–4684.

[14] Yongjae Lee et al. "Sparse and robust portfolio selection via semi-definite relaxation". In: *Journal of the Operational Research Society* 71 (5 May 2020), pp. 687–699. ISSN: 0160-5682. DOI: 10.1080/01605682.2019.1581408.

[15] Bo Li et al. "Uncertain portfolio optimization problem under a minimax risk measure". In: *Applied Mathematical Modelling* 76 (Dec. 2019), pp. 274–281. ISSN: 0307904X. DOI: 10.1016/j.apm.2019.06.019.

[16] Duan Li, Xiaoling Sun, and Jun Wang. "Optimal lot solution to cardinality constrained mean-variance formulation for portfolio selection". In: *Mathematical Finance* 16 (1 Jan. 2006), pp. 83–101. ISSN: 0960-1627. DOI: 10.1111/j.1467-9965.2006.00262.x.

[17] Justus Shunza, Mary Akinyemi, and Chika Yinka-Banjo. "Application of quantum computing in discrete portfolio optimization". In: *Journal of Management Science and Engineering* 8 (4 Dec. 2023), pp. 453–464. ISSN: 25895532. DOI: 10.1016/j.jmse.2023.02.001.

[18] Giuseppe Buonaiuto et al. "Best practices for portfolio optimization by quantum computing, experimented on real quantum devices". In: *Scientific Reports* 13 (1 Dec. 2023). ISSN: 20452322. DOI: 10.1038/s41598-023-45392-w.

[19] Shaen Corbet et al. "Cryptocurrencies as a financial asset: A systematic analysis". In: *International Review of Financial Analysis* 62 (Mar. 2019), pp. 182–199. ISSN: 10575219. DOI: 10.1016/j.irfa.2018.09.003.

[20] Dirk G. Baur, Ki Hoon Hong, and Adrian D. Lee. "Bitcoin: Medium of exchange or speculative assets?" In: *Journal of International Financial Markets, Institutions and Money* 54 (May 2018), pp. 177–189. ISSN: 10424431. DOI: 10.1016/j.intfin.2017.12.004.

[21] ladislav Kristoufek. "What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis". In: *PLOS ONE* 10 (4 Apr. 2015), pp. 1–15. DOI: 10.1371/journal.pone.0123923.

[22] Dan Simon. *Evolutionary Optimization Algorithms*. John Wiley & Sons, 2013.

[23] Dirk Thierens and Peter A.N. Bosman. "Optimal Mixing Evolutionary Algorithms". In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA, July 2011, pp. 617–624. ISBN: 9781450305570. DOI: 10.1145/2001576.2001661.

[24] E. P. Setiawan. "Comparing bio-inspired heuristic algorithm for the mean-CVaR portfolio optimization". In: vol. 1581. Institute of Physics Publishing, July 2020. DOI: 10.1088/1742-6596/1581/1/012014.

[25] Anton Bouter and Peter A. N. Bosman. "A Joint Python/C++ Library for Efficient yet Accessible Black-Box and Gray-Box Optimization with GOMEA". In: *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. Association for Com-

puting Machinery, July 2023, pp. 1864–1872. DOI: 10.1145/3583133.3596361.

[26] Aurelio F. Bariviera. "The inefficiency of Bitcoin revisited: A dynamic approach". In: *Economic Letters* 161 (5 Dec. 2017), pp. 1–4. DOI: https://doi.org/10.1016/j.econlet.2017.09.013.

[27] Shermin Voshmgir. *Token Economy: How the Web3 Reinvents the Internet*. Vol. 2. Token Kitchen, 2020.

[28] Rowan van Pelt et al. "Defining blockchain governance: A framework for analysis and comparison". In: *Information Systems Management* 38.1 (Jan. 2021), pp. 21–41. DOI: 10.1080/10580530.2020.1720046.

[29] Eng-Tuck Cheah and John Fry. "Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin". In: *Economic Letters* 130 (130 May 2015), pp. 32–36.

[30] Shaen Corbet et al. "Exploring the Dynamic Relationships between Cryptocurrencies and Other Financial Assets". In: *Economics Letters* 165 (Apr. 2018), pp. 28–34. ISSN: 0165-1765. DOI: 10.1016/j.econlet.2018.01.004.

[31] Stephen Chan et al. "A Statistical Analysis of Cryptocurrencies". In: *Journal of Risk and Financial Management* 10 (2 May 2017), p. 12. DOI: 10.3390/jrfm10020012.

[32] Marie Brière, Kim Oosterlinck, and Ariane Szafarz. "Virtual currency, tangible return: Portfolio diversification with bitcoin". In: *Journal of Asset Management* 16 (6 Nov. 2015), pp. 365–373. ISSN: 1470-8272. DOI: 10.1057/jam.2015.5.

[33] George Selgin. "Synthetic commodity money". In: *Journal of Financial Stability* 17 (Apr. 2015), pp. 92–99. DOI: https://doi.org/10.1016/j.jfs.2014.07.002.

[34] Ilyes Abid andZied Ftiti Khaled Guesmi Samir Saadi. "Portfolio diversification with virtual currency: Evidence from bitcoin". In: *International Review of Financial Analysis* 63 (May 2019), pp. 431–437. DOI: https://doi.org/10.1016/j.irfa.2018.03.004.

[35] Sakshi Vasudeva. "Cryptocurrency as an investment or speculation: a bibliometric review study". In: *Business Analyst Journal* 44 (1 Oct. 2023), pp. 34–50. ISSN: 0973-211X. DOI: 10.1108/baj-07-2022-0008.

[36] C. Baek and M. Elbeck. "Bitcoins as an investment or speculative vehicle? A first look". In: *Applied Economics Letters* 22 (1 Jan. 2015), pp. 30–34. ISSN: 1350-4851. DOI: 10.1080/13504851.2014.916379.

[37] Dirk G. Baur and Thomas Dimpfl. "Asymmetric volatility in cryptocurrencies". In: *Economics Letters* 173 (Dec. 2018), pp. 148–151. ISSN: 01651765. DOI: 10.1016/j.econlet.2018.10.008.

[38] Syed Jawad Hussain Shahzad Elie Bouri and David Roubaud. "Coexplosivity in the cryptocurrency market". In: *Economic Letters* 29 (June 2019), pp. 178–183.

[39] Theodore Panagiotidis, Georgios Papapanagiotou, and Thanasis Stengos. "On the volatility of cryptocurrencies". In: *Research in International Business and Finance* 62 (Dec. 2022). ISSN: 02755319. DOI: 10.1016/j.ribaf.2022.101724.

[40] Sven Thies and Peter Molnár. "Bayesian change point analysis of Bitcoin returns". In: *Finance Research Letters* 27 (Dec. 2018), pp. 223–227. DOI: https://doi.org/10.1016/j.frl.2018.03.018.

[41] Anamika, Madhumita Chakraborty, and Sowmya Subramaniam. "Does Sentiment Impact Cryptocurrency?" In: *Journal of Behavioral Finance* 24 (2 Apr. 2023), pp. 202–218. ISSN: 1542-7560. DOI: 10.1080/15427560.2021.1950723.

[42] Ahmad Chokor and Elise Alfieri. "Long and short-term impacts of regulation in the cryptocurrency market". In: *The Quarterly Review of Economics and Finance* 81 (Aug. 2021), pp. 157–173. DOI: https://doi.org/10.1016/j.qref.2021.05.005.

[43] Paraskevi Katsiampa. "Volatility estimation for Bitcoin: A comparison of GARCH models". In: *Economics Letters* 158 (Sept. 2017), pp. 3–6. ISSN: 01651765. DOI: 10.1016/j.econlet.2017.06.023.

[44] Stavros Stavroyiannis and Vassilios Babalos. "Dynamic Properties of the Bitcoin and the US Market". In: *SSRN Electronic Journal* (May 2017). DOI: 10.2139/ssrn.2966998. URL: https://ssrn.com/abstract=2966998.

[45] Shi Chen et al. *A First Econometric Analysis of the CRIX Family*. Available at SSRN: https://ssrn.com/abstract=2832099 or http://dx.doi.org/10.2139/ssrn.2832099. Aug. 2016. DOI: 10.2139/ssrn.2832099.

[46] Jeffrey Chu et al. "GARCH Modelling of Cryptocurrencies". In: *Journal of Risk and Financial Management* 10 (4 Oct. 2017), p. 17. DOI: 10.3390/jrfm10040017.

[47] Guglielmo Maria Caporale, Nikitas Pittis, and Nicola Spagnolo. "IGARCH models and structural breaks". In: *Applied Economics Letters* 10 (12 Oct. 2003), pp. 765–768. ISSN: 13504851. DOI: 10.1080/1350485032000138403.

[48] Razvan Pascalau, Christian Thomann, and Greg N. Gregoriou. "Unconditional Mean, Volatility, and the Fourier-GARCH Representation". In: *Financial Econometrics Modeling: Derivatives Pricing, Hedge Funds and Term Structure Models*. London: Palgrave Macmillan UK, 2011, pp. 90–106.

[49] Frederik Armknecht et al. "Ripple: Overview and Outlook". In: *Trust and Trustworthy Computing*. Ed. by M. Conti, M. Schunter, and I. Askoxylakis. Vol. 9229. Lecture Notes in Computer Science. Cham: Springer, 2015. DOI: 10.1007/978-3-319-22846-4_10.

[50] Yhlas Sovbetov. "Factors Influencing Cryptocurrency Prices: Evidence from Bitcoin, Ethereum, Dash, Litcoin, and Monero". In: *Journal of Economics and Financial Analysis* (Feb. 2018), pp. 1–27. DOI: 10.1991/jefa.v2i2.a16.

[51] Abhishek Gunjan and Siddhartha Bhattacharyya. “A brief review of portfolio optimization techniques”. In: *Artificial Intelligence Review* 56 (5 May 2023), pp. 3847–3886. ISSN: 15737462. DOI: 10.1007/s10462-022-10273-7.

[52] Gregory Connor, Lisa R. Goldberg, and Robert A. Korajczyk. *Portfolio risk analysis*. Princeton University Press, 2010.

[53] Yves Choueifaty and Yves Coignard. “Toward Maximum Diversification”. In: *The Journal of Portfolio Management* 35.1 (2008), pp. 40–51.

[54] Valeriy Zakamulin. “Optimal dynamic portfolio risk management”. In: *Journal of Portfolio Management* 43.1 (Oct. 2016), p. 85.

[55] Pierre Brugière. *Quantitative portfolio management*. Springer Texts in Business and Economics. Springer, 2020.

[56] William F. Sharpe. “Capital asset prices: a theory of market equilibrium under conditions of risk”. In: *The Journal of Finance* 19 (3 Sept. 1964), pp. 425–442. ISSN: 0022-1082. DOI: 10.1111/j.1540-6261.1964.tb02865.x.

[57] Robert C. Merton. “An Intertemporal Capital Asset Pricing Model”. In: *Econometrica* 41 (5 Sept. 1973), p. 867. ISSN: 00129682. DOI: 10.2307/1913811.

[58] Fischer Black and Robert Litterman. “Global Portfolio Optimization”. In: *Financial Analysts Journal* 48.5 (Sept. 1992), pp. 28–43.

[59] Andrew Lo. “The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective”. In: *Journal of Portfolio Management, Forthcoming* (Oct. 2004). ISSN: 15446123.

[60] Stoyan V. Stoyanov Svetlozar Todorov and Frank J. Fabozzi. Rachev. *Advanced stochastic models, risk assessment, and portfolio optimization : the ideal risk, uncertainty, and performance measures*. John Wiley & Sons, 2008.

[61] Wei Chen et al. “Mean–variance portfolio optimization using machine learning-based stock price prediction”. In: *Applied Soft Computing* 100 (2021). ISSN: 15684946. DOI: https://doi.org/10.1016/j.asoc.2020.106943.

[62] Mikkel. Rasmussen. *Quantitative Portfolio Optimisation, Asset Allocation and Risk Management: A Practical Guide to Implementing Quantitative Investment Theory*. Springer, 2002.

[63] Rüdiger Frey Alexander J. and Paul Embrechts McNeil. *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton university press, 2015.

[64] Eric Jondeau and Michael Rockinger. “Optimal Portfolio Allocation under Higher Moments”. In: *European Financial Management* 12 (1 Jan. 2006), pp. 29–55. ISSN: 1354-7798. DOI: 10.1111/j.1354-7798.2006.00309.x.

[65] Hersh Shefrin and Meir Statman. “Behavioral Portfolio Theory”. In: *The Journal of Financial and Quantitative Analysis* 35 (2 June 2000), p. 127. ISSN: 00221090. DOI: 10.2307/2676187.

[66] Tim Bollerslev. "Generalized autoregressive conditional heteroskedasticity". In: *Journal of Econometrics* 31 (3 1986), pp. 307–327. ISSN: 03044076. DOI: 10.1016/0304-4076(86)90063-1.

[67] Francis X. Diebold and Kamil Yilmaz. *Macroeconomic Volatility and Stock Market Volatility, Worldwide*. Working Paper w14269. National Bureau of Economic Research, Aug. 2008.

[68] Konstantinos Gkillas, Rangan Gupta, and Mark E. Wohar. "Volatility jumps: The role of geopolitical risks". In: *Finance Research Letters* 27 (Dec. 2018), pp. 247–258. ISSN: 15446123. DOI: 10.1016/j.frl.2018.03.014.

[69] Wayne Y Lee, Christine X Jiang, and Daniel C Indro. "Stock market volatility, excess returns, and the role of investor sentiment". In: *Journal of banking & Finance* 26 (12 2002), pp. 2277–2299. URL: www.elsevier.com/locate/econbase.

[70] Dirk Bleich, Ralf Fendel, and Jan-Christoph Rülke. "Monetary Policy and Stock Market Volatility". In: *Economics Bulletin* 33.3 (2013), pp. 1669–1680. DOI: 10.2139/ssrn.2796936.

[71] Daniel B Nelson. "Conditional Heteroskedasticity in Asset Returns: A New Approach". In: *Econometrica* 59 (2 Dec. 1991), p. 347. DOI: https://doi.org/10.2307/2938260.

[72] Golnoosh Babaei, Paolo Giudici, and Emanuela Raffinetti. "Explainable artificial intelligence for crypto asset allocation". In: *Finance Research Letters* 47 (June 2022). ISSN: 15446123. DOI: 10.1016/j.frl.2022.102941.

[73] Abdelouahed Hamdi et al. "Portfolio Selection Problem Using CVaR Risk Measures Equipped with DEA, PSO, and ICA Algorithms". In: *Mathematics* 10 (15 Aug. 2022), p. 2808. ISSN: 2227-7390. DOI: 10.3390/math10152808.

[74] Alexei A. Gaivoronski and Georg Pflug. "Value-at-risk in portfolio optimization: properties and computational approach". In: *Journal of risk* 7 (2 2005), pp. 1–31. DOI: 10.21314/JOR.2005.106.

[75] Giorgio Costa and Roy H. Kwon. "Risk parity portfolio optimization under a Markov regime-switching framework". In: *Quantitative Finance* 19 (3 Mar. 2019), pp. 453–471. ISSN: 1469-7688. DOI: 10.1080/14697688.2018.1486036.

[76] "Generalized risk parity portfolio optimization: an ADMM approach". In: *Journal of Global Optimization* 78 (1 Sept. 2020), pp. 207–238. ISSN: 15732916. DOI: 10.1007/s10898-020-00915-x.

[77] Vaughn Gambeta and Roy Kwon. "Risk Return Trade-Off in Relaxed Risk Parity Portfolio Optimization". In: *Journal Risk Financial Management* 13 (10 Oct. 2020), p. 237. DOI: https://doi.org/10.3390/jrfm13100237.

[78] James Ming Chen. "The Capital Asset Pricing Model". In: *Encyclopedia* 1 (3 Sept. 2021), pp. 915–933. DOI: 10.3390/encyclopedia1030070.

[79] Marc S. Paolella and Paweł Polak. "COBra: Copula-Based Portfolio Optimization". In: *International Conference of the Thailand Econometrics Society*. Springer International Publishing, Dec. 2017, pp. 36–77.

[80] Hengxin Cui, Ken Seng Tan, and Fan Yang. "Portfolio credit risk with Archimedean copulas: asymptotic analysis and efficient simulation". In: *Annals of Operations Research* (2022). ISSN: 15729338. DOI: 10.1007/s10479-022-04717-0.

[81] David A. Hennessy and Harvey E. Lapan. "The Use of Archimedean Copulas to Model Portfolio Allocations". In: *Mathematical Finance* 12 (2 Apr. 2002), pp. 143–154. ISSN: 0960-1627. DOI: 10.1111/1467-9965.00136.

[82] Paul Embrechts. "Copulas: A Personal View". In: *Journal of Risk and Insurance* 76 (3 Sept. 2009), pp. 639–650. ISSN: 0022-4367. DOI: 10.1111/j.1539-6975.2009.01310.x.

[83] Damian Zięba, Ryszard Kokoszczyński, and Katarzyna Śledziewska. "Shock transmission in the cryptocurrency market. Is Bitcoin the most influential?" In: *International Review of Financial Analysis* 64 (2019), pp. 102–125.

[84] Rishal Hurbans. *Grokking artificial intelligence algorithms*. Manning Publications, 2020. ISBN: 9781617296185.

[85] Antonin Ponsich, Antonio Lopez Jaimes, and Carlos A. Coello Coello. "A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications". In: *Institute of electric and Electronics Engineers Transactions on Evolutionary Computation* 17.3 (3 Apr. 2012), pp. 321–344. ISSN: 1089-778X. DOI: 10.1109/TEVC.2012.2196800.

[86] Ray Lim et al. "Non-linear Domain Adaptation in Transfer Evolutionary Optimization". In: *Cognitive Computation* 13 (2 Mar. 2021), pp. 290–307. ISSN: 18669964. DOI: 10.1007/s12559-020-09777-7.

[87] Maumita Bhattacharya, Rafiqul Islam, and Jemal Abawajy. "Evolutionary optimization: A big data perspective". In: *Journal of Network and Computer Applications* 59 (Jan. 2016), pp. 416–426. ISSN: 10958592. DOI: 10.1016/j.jnca.2014.07.032.

[88] Shatendra Singh and Aruna Tiwari. *Enhanced Opposition Differential Evolution Algorithm for Multimodal Optimization*. Aug. 2022. arXiv: 2208.11066 [cs.NE].

[89] B.H. Sumida et al. "Genetic algorithms and evolution". In: *Journal of Theoretical Biology* 147.1 (Nov. 1990), pp. 59–84. ISSN: 00225193. DOI: 10.1016/S0022-5193(05)80252-8.

[90] Salvatore Arnone, Andrea Loraschi, and Andrea Tettamanzi. "A Genetic Approach to Portfolio Selection". In: *Neural Network World* 3.6 (1993), pp. 597–604.

[91] E. P. Setiawan and D. Rosadi. "Solving Portfolio Selection Problems with Minimum Transaction Lots Based on Conditional-Value-

at-Risk". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 166. 1. IOP Publishing, Feb. 2017, p. 012005. DOI: 10.1088/1757-899X/166/1/012005.

[92] Vladimir Ranković et al. "The mean-Value at Risk static portfolio optimization using genetic algorithm". In: *Computer Science and Information Systems* 11 (1 Jan. 2014), pp. 89–109. ISSN: 18200214. DOI: 10.2298/CSIS121024017R.

[93] Tun Jen Chang, Sang Chin Yang, and Kuang Jung Chang. "Portfolio optimization problems in different risk measures using genetic algorithm". In: *Expert Systems with Applications* 36 (7 Sept. 2009), pp. 10529–10537. ISSN: 09574174. DOI: 10.1016/j.eswa.2009.02.062.

[94] James Kennedy and Russell Eberhart. "Particle Swarm Optimization". In: *Proceedings of International Conference on Neural Networks'95*. Vol. 4. Institute of Electric and Electronic Engineers, Nov. 1995, pp. 1942–1948.

[95] John Schulman et al. "Proximal Policy Optimization Algorithms". In: (July 2017). URL: http://arxiv.org/abs/1707.06347.

[96] Dirk Thierens and Peter A.N. Bosman. "Hierarchical Problem Solving with the Linkage Tree Genetic Algorithm". In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: Association for Computing Machinery, July 2013, pp. 877–884. ISBN: 9781450319638. DOI: 10.1145/2463372.2463477.

[97] Arkadiy Dushatskiy et al. "Parameterless Gene-pool Optimal Mixing Evolutionary Algorithms". In: *Evolutionary Computation* (June 2023), pp. 1–28. DOI: 10.1162/evco_a_00338.

[98] K. Kiani. "Relationship between portfolio diversification and value at risk: Empirical evidence". In: *Emerging Markets Review* 12.4 (2011), pp. 443–459. ISSN: 15660141. DOI: 10.1016/j.ememar.2010.12.004.

[99] Goodell JGoutte S. "Diversifying equity with cryptocurrencies during COVID-19". In: *International Review of Financial Analysis* 76 (July 2021). ISSN: 0022-1082. DOI: https://doi.org/10.1016/j.irfa.2021.101781.

[100] Anshul Gopal, Mohammad Mahdi Sultani, and Jagdish Chand Bansal. "On Stability Analysis of Particle Swarm Optimization Algorithm". In: *Arabian Journal for Science and Engineering* 45.4 (2020), pp. 2385–2394. ISSN: 2193-567X. DOI: 10.1007/s13369-019-03991-8.