Available online at http://iorajournal.org/indx.php/orics /index

Operations Research: International Conference Series

e-ISSN: 2722-0974
p-ISSN: 2723-1739

Vol. 6, No. 2, pp. 107-116, 2025

Indirect Methods for Personalized Mean-CVaR Portfolio Optimization

Deva Putra Setyawan[1*], Yasir Salih[2]

[1]*Master Program in the Mathematics Department, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang, 45363, Indonesia*
[2]*Department of Mathematics, Faculty of Education, Red Sea University, SUDAN*

**Corresponding author email: deva24002@mail.unpad.ac.id*

Abstract

This study presents a reformulation of the Personalized Mean-CVaR model into an unconstrained optimization problem, which is then solved using iterative methods, including steepest descent and Newton's method. The reformulation introduces challenges related to feasibility region checking, convexity of the feasible set and objective functions, and the use of Lagrange multipliers to handle constraints. Additionally, Taylor expansion is utilized to approximate the objective function in each iteration. The research evaluates the effectiveness of iterative optimization techniques in solving the Personalized Mean-CVaR problem, while addressing key challenges in convergence and stability of the solution.

Keywords: Personalized Mean-CvaR, Portfolio Optimization, Indirect Methods, Steepest Descent, Newton's Method

1. Introduction

Portfolio optimization is a critical aspect of financial decision-making, aiming to identify the optimal combination of assets to achieve a desired risk-return tradeoff. Harry Markowitz's mean-variance optimization, a cornerstone of traditional portfolio theory, has been widely applied for decades. However, this approach has limitations, particularly in addressing extreme risks like market crashes, which variance alone often fails to capture. Consequently, alternative risk measures such as Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) have emerged. These measures offer a more robust assessment and management of tail risks, making them more effective in quantifying risks associated with severe financial events.

Despite the improvements offered by CVaR, many traditional models still lack the ability to incorporate personalized investor preferences, such as individual risk tolerance and financial goals. While numerous portfolio optimization models assume a uniform risk profile, personalized models that adapt to unique investor characteristics have gained increasing importance. Integrating personalized risk measures into portfolio optimization ensures that investors' specific constraints and preferences are adequately represented in the decision-making process.

The primary objective of this study is to propose a personalized portfolio optimization model utilizing Mean-CVaR as the risk measure. This model aims to optimize portfolio allocation by considering individual risk tolerance and preferences. We will reformulate the optimization problem into an unconstrained form using methods such as Lagrange multipliers. Subsequently, we will apply iterative optimization techniques, specifically steepest descent and Newton's method, to solve the problem efficiently. Furthermore, this study will analyze the computational efficiency and convergence of the proposed methods to assess their practicality for real-world applications.

The significance of this research lies in its capacity to offer a more personalized approach to portfolio optimization, particularly for investors seeking to minimize risk in extreme scenarios. By employing CVaR, this study enhances the portfolio optimization process by capturing tail risks that traditional models frequently overlook. The application of advanced iterative methods also contributes to the field by providing an efficient and effective means to solve complex portfolio optimization problems, thereby offering valuable insights for both practitioners and researchers in financial optimization.

2. Literature Review

Personalized portfolio optimization is an advanced approach that integrates individual investor preferences, encompassing their risk tolerance, goals, and constraints (Yu and Liu, 2021). Unlike traditional models that assume a

generic risk-return profile, this method tailors the optimization process to an investor's unique characteristics. The fuzzy comprehensive evaluation (FCE) method is a common technique used to quantify investor risk tolerance. This approach relies on subjective factors like age, income, investment experience, and risk preferences, enabling the inclusion of personalized risk profiles in the portfolio optimization model. Banihashemi and Navidi (2017) further demonstrate how personalized risk measures, such as Mean-CVaR, improve portfolio performance evaluation compared to conventional methods like Value-at-Risk (VaR).

The academic literature on portfolio optimization has seen a significant evolution from classical methods to more sophisticated techniques. While quadratic programming remains widely used for mean-variance optimization, newer approaches increasingly employ Conditional Value-at-Risk (CVaR) as the risk measure to address non-normal return distributions and capture extreme events. Several studies underscore the advantages of CVaR over traditional mean-variance optimization. Rockafellar and Uryasev (2000) developed a foundational framework for optimizing portfolios based on CVaR, illustrating its superior ability to manage risk, particularly concerning extreme losses. Chen and Wang (2009) further applied CVaR in portfolio selection, showing its capacity to outperform mean-variance optimization in minimizing downside risk, especially during periods of market stress. Additionally, Yao et al. (2013) introduced a nonparametric estimation framework for Mean-CVaR, highlighting its benefits in capturing the tail risk of portfolio returns.

Various optimization methodologies have been employed to solve complex portfolio problems. Gradient-based methods, including steepest descent and Newton's method, are particularly effective for smooth, differentiable objective functions (Rao, 2020). These methods rely on the first and second derivatives of the objective function, respectively, and can be efficiently applied once the portfolio optimization problem is reformulated into an unconstrained form. For problems that are non-differentiable or highly nonlinear, metaheuristic algorithms such as Genetic Algorithms and Particle Swarm Optimization (PSO) offer flexible solutions. While these algorithms do not always guarantee convergence to the global optimum, they provide robust solutions for complex problem instances. Addressing inherent uncertainties in financial markets, Mahmutogullari et al. (2018) introduced a mixed-integer multi-stage stochastic programming approach for Mean-CVaR problems, focusing on risk-averse optimization in multi-stage settings and providing bounds to handle uncertainty.

In summary, the literature on portfolio optimization has shifted towards advanced techniques that incorporate CVaR as a more realistic risk measure, crucial for assessing tail risk and extreme losses in financial decision-making. The application of iterative optimization methods, such as steepest descent and Newton's method, along with the use of Lagrange multipliers for problem reformulation and Taylor expansion for objective function approximation, offers effective solutions for these complex problems. The integration of personalized risk measures, particularly Mean-CVaR, into portfolio optimization holds significant potential to deliver more robust and tailored solutions that genuinely reflect individual investors' specific risk profiles and constraints.

3. Methods

3.1. Personalized Mean-CVaR Portfolio Optimization Model

In portfolio optimization, selecting an appropriate risk measure is crucial for accurately representing potential financial risks. Traditional measures like variance effectively capture general volatility but often inadequately represent extreme loss risks (Markowitz, 1952). Thus, alternative risk measures such as Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) have been introduced. CVaR, also known as Expected Shortfall, represents the expected loss given that the loss has exceeded the VaR threshold. Unlike VaR, CVaR is a coherent risk measure satisfying subadditivity, positive homogeneity, translation invariance, and monotonicity (Rockafellar and Uryasev, 2000), making it particularly suitable for portfolio optimization.

3.1.1. Mean-CVaR Model Formulation

The Mean-CVaR model aims to minimize portfolio risk, as measured by CVaR, while ensuring a predefined expected return. This model has been widely recognized for its ability to capture tail risks more effectively than traditional variance-based approaches (Chen and Wang, 2009; Yao et al., 2013). The mathematical formulation of the Mean-CVaR portfolio optimization problem is expressed as follows:

$$\min_{\mathbf{w},\gamma} \gamma + \frac{1}{1-\alpha}\mathbb{E}[\max(-\mathbf{w}^\top\mathbf{r} - \gamma, 0)]$$

s.t.

$$\begin{gathered}\mathbf{w}^\top\boldsymbol{\mu} \geq R_0 \\ \mathbf{e}^\top\mathbf{w} = 1 \\ \mathbf{w} \geq \mathbf{0}\end{gathered} \tag{1}$$

In this formulation, $\mathbf{w}$ denotes the portfolio investment weight vector, $\mathbf{r}$ represents the vector of asset returns, and $\boldsymbol{\mu}$ is the vector of expected returns for each asset. The parameter R_0 indicates the minimum desired expected return

specified by the investor, α is the confidence level (typically chosen as 95% or 99%), and γ is an auxiliary variable associated with Value-at-Risk. The vector e has all elements equal to 1.

These constraints ensure that the portfolio's expected return meets or exceeds the minimum desired return R_0, that the total weight of the portfolio sums to 1 (maintaining a fully invested portfolio), and that all portfolio weights are non-negative (meaning the model only considers long positions in the assets).

3.1.2. Investor Risk Tolerance Determination

Personalized portfolio optimization necessitates incorporating the individual investor's risk tolerance into the model (Yu and Liu, 2021). In this context, risk tolerance is quantified based on various demographic and financial characteristics, such as gender, age, education level, and investment knowledge. These factors are evaluated using a fuzzy comprehensive evaluation (FCE) method to determine the investor's risk profile. The use of FCE allows for a more nuanced and subjective assessment of risk preference, beyond simple numerical scales.

Let $H = \{h_1, h_2, \dots, h_l\}$ denote the set of evaluation criteria, and $G = \{g_1, g_2, \dots, g_f\}$ the set of predefined risk levels (e.g., conservative, moderately conservative, moderate, moderately aggressive, aggressive). The fuzzy relationship matrix $R = \left[q_{(ij)}\right]_{l\times f}$ represents the degree to which each criterion hi corresponds to each risk level g_j. A weight vector $\mathbf{w}^{(r)} = \left(w_1^{(r)}, w_2^{(r)}, \dots, w_l^{(r)}\right)$ is assigned to each criterion based on expert judgment or survey data, satisfying $\sum_{i=1}^{l} w_i^{(r)} = 1$. The comprehensive evaluation vector b is then calculated as:

$$\mathbf{b} = \mathbf{w}^{(r)} R \tag{2}$$

Finally, the overall risk tolerance score d is given by

$$d = \mathbf{a}^{\top}\mathbf{b} = \sum_{j=1}^{f} e_j b_j \tag{3}$$

where $\mathbf{a} = (\mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_f)$ is a vector of numeric scores corresponding to each risk level. For instance, these scores might be assigned as {0.1,0.2,0.3,0.4,0.5} for increasing levels of risk tolerance, ranging from highly conservative to highly aggressive.

This quantified risk tolerance value d is subsequently used to determine the investor's minimum desired expected return R_0. This mechanism allows the model to be personalized, aligning the portfolio's return objectives with the individual's unique capacity and preference for financial risk

3.1.3. Return Distribution Modeling with Copula-GARCH

To accurately estimate portfolio risk within the Mean-CVaR framework, it is essential to model the underlying distribution of asset returns and generate relevant scenarios. In this study, we employ a combination of the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model and Copula theory to capture both the marginal behavior and the complex dependence structure of asset returns (Mahmutogullari et al., 2018). This approach helps in addressing the stylized facts of financial data, such as volatility clustering and fat tails. The GARCH(1,1) model is utilized to model the marginal distribution of each asset return's volatility:

$$r_{t,i} = \mu_i + a_{t,i} \tag{4}$$

$$a_{t,i} = \sigma_{t,i}\varepsilon_{t,i} \tag{5}$$

$$\sigma_{t,i}^2 = \beta_{0i} + \beta_{1i}a_{t-1,i}^2 + \beta_{2i}\sigma_{t-1,i}^2 \tag{6}$$

where $\varepsilon_{t,i}$ represents the standardized residuals, which typically follow a Student's t-distribution to accommodate the observed fat tails in financial return distributions. To model the joint behavior and inter-dependencies of returns across multiple assets, a t-Copula is applied:

$$C_t(u_1, \dots, u_m) = t_{\lambda,\mathbf{\Sigma}}\left(t_{\theta_1}^{-1}(u_1), \dots, t_{\theta_m}^{-1}(u_m)\right) \tag{7}$$

where m is the number of assets, u_i are the marginal CDF values, λ represents the degrees of freedom of the t-Copula, $\mathbf{\Sigma}$ is the correlation matrix, and $t_{\theta_1}^{-1}$ denotes the inverse Cumulative Distribution Function (CDF) of the marginal t-distributions for each asset's standardized residuals. The t-Copula is chosen for its ability to capture both linear and non-linear dependencies, especially tail dependencies common in financial markets

Based on the calibrated Copula-GARCH model, future scenarios of asset returns are simulated using Monte Carlo techniques. This comprehensive process involves the following steps:

- Estimating the parameters of the GARCH model from historical return data for each asset.
- Fitting the Copula to capture the empirical dependencies between asset returns' standardized residuals.
- Generating a large number (N) of random samples from the joint distribution implied by the estimated Copula-GARCH model.
- Transforming these samples back into realistic asset return scenarios that will serve as the input for CVaR estimation in the optimization problem.

These generated scenarios are crucial for evaluating the portfolio's risk under various possible future market conditions, providing a robust and comprehensive input for the CVaR optimization

3.1.4. Personalized Mean-CVaR Optimization Problem

Integrating the concepts described in the preceding subsections, the personalized Mean-CVaR portfolio optimization model aims to minimize the Conditional Value-at-Risk (CVaR) while explicitly considering the investor's risk tolerance. The model's personalization is primarily achieved by adjusting the minimum desired expected return (R_0) in accordance with the investor's quantified risk profile. This profile is determined using the fuzzy comprehensive evaluation (FCE) method, based on demographic and investment-related characteristics.

The optimization problem, utilizing the N generated scenarios from Monte Carlo simulation, is formulated as a linear programming problem for computational efficiency, following the standard approach for CVaR minimization (Rockafellar and Uryasev, 2000):

$$\min_{\mathbf{w},\gamma,\tau_j} \gamma + \frac{1}{(1-\alpha)N}\sum_{j=1}^{N}\tau_j$$

s.t.

$$\begin{gathered}\mathbf{w}^\top \mathbf{r}^{(j)} - \gamma \le \tau_j, \qquad \forall j = 1,\dots,N \\ \mathbf{w}^\top \boldsymbol{\mu} \ge R_0 \\ \mathbf{e}^\top \mathbf{w} = 1 \\ \mathbf{w} \ge \mathbf{0} \\ \tau_j \ge 0, \qquad \forall j = 1,\dots,N\end{gathered} \tag{8}$$

Here, r_j is the vector of asset returns in scenario j, and τ_j are auxiliary variables representing the loss beyond γ (VaR) for each scenario. The objective function in Equation (8) minimizes γ plus the average of these excesses over the N scenarios.

This formulation effectively incorporates the investor's risk tolerance by dynamically determining R_0. By solving this optimization problem, the model identifies the optimal portfolio weights that balance the trade-off between risk (as measured by CVaR) and return, tailored precisely to the personalized requirements of the individual investor. The problem is convex and can be solved using various optimization techniques.

3.2. Unconstrained Optimization

Transforming a constrained optimization problem into an unconstrained one is a key step that enables the use of standard iterative optimization methods, such as steepest descent and Newton's method, for solving complex portfolio optimization problems. This transformation is typically achieved using the Lagrange multiplier method, which systematically incorporates problem constraints directly into the objective function.

3.2.1. Lagrange Multiplier Approach

Consider the general constrained optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x})$$

s.t. (9)

$$g_i(\mathbf{x}) = 0, \qquad \forall i = 1,\dots,m$$

where $f(\mathbf{x})$ is the objective function to be minimized, and $g_i(\mathbf{x}) = 0$ are the m equality constraints. The Lagrange function $\mathcal{L}(\mathbf{x},\lambda)$ is defined as:

$$\mathcal{L}(\mathbf{x},\lambda) = f(\mathbf{x}) + \sum_{i=1}^{m}\lambda_i g_i(\mathbf{x}) \tag{10}$$

where λ_i are the Lagrange multipliers associated with each constraint. For multiple multipliers, $\boldsymbol{\lambda} = (\lambda_1,\dots,\lambda_m)$ is the vector of Lagrange multipliers. The necessary conditions for a local optimum of the constrained problem (9) are obtained by setting the gradient of the Lagrange function with respect to $\mathbf{x}$ and λ to zero. These are known as the Karush-Kuhn-Tucker (KKT) conditions for equality-constrained problems (Rao, 2020):

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \nabla f(\mathbf{x}) + \sum_{i=1}^{m}\lambda_i \nabla g_i(\mathbf{x}) = 0 \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = g_i(\mathbf{x}) = 0, \qquad \forall i = 1,\dots,m \tag{12}$$

This system of equations allows us to solve for the optimal solution x∗ and the Lagrange multipliers λ^*.

3.2.2. Karush-Kuhn-Tucker (KKT) Conditions

When dealing with inequality constraints, the formulation extends to the more general Karush-Kuhn-Tucker (KKT) conditions. These conditions are essential for ensuring that a candidate solution satisfies both optimality and feasibility requirements for problems with both equality and inequality constraints. Consider a general optimization problem with both equality and inequality constraints:

$$\min_{\mathbf{x}} f(\mathbf{x}),$$

s.t.

$$\begin{aligned} g_i(\mathbf{x}) &\leq 0, \qquad \forall i = 1, \dots, m, \\ h_j(\mathbf{x}) &= 0, \qquad \forall j = 1, \dots, q. \end{aligned} \tag{13}$$

The KKT conditions state that for a point $\mathbf{x}^*$ to be a local optimum, there must exist Lagrange multipliers $\lambda^* \in \mathbb{R}^p$ and $\mu^* \in \mathbb{R}^q$ such that the following conditions are met (Rao, 2020):

1. Stationarity: $\nabla f(\mathbf{x}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^{q} \mu_j^* \nabla h_j(\mathbf{x}^*) = 0.$
2. Primal Feasibility: $g_i(\mathbf{x}^*) \leq 0,\ i = 1, \dots, p; \quad h_j(\mathbf{x}^*) = 0,\ j = 1, \dots, q.$
3. Dual Feasibility: $\lambda_i^* \geq 0,\ i = 1, \dots, p.$
4. Complementary Slackness: $\lambda_i^* g_i(\mathbf{x}^*) = 0,\ i = 1, \dots, p.$

The complementary slackness condition is particularly important for inequality constraints: if an inequality constraint $g_i(\mathbf{x}^*)$ is *inactive* (i.e., $g_i(\mathbf{x}^*) < 0$), then its corresponding Lagrange multiplier λ_i^* must be zero. Conversely, if $\lambda_i^* > 0$, then the constraint must be *active* (i.e., $g_i(\mathbf{x}^*) = 0$).

In the context of the Mean-CVaR portfolio optimization problem presented in Section 3.1.4, which includes both equality and inequality constraints, the KKT conditions provide the necessary framework for converting the problem into a form amenable to unconstrained optimization techniques. By setting the gradient of the Lagrangian to zero and satisfying the complementary slackness conditions, the optimal portfolio weights $\mathbf{w}$ can be determined iteratively

3.3. Indirect Search (Descent) Methods

Indirect search methods, also commonly known as descent methods or gradient-based methods, are fundamental in numerical optimization. They provide powerful tools to find the minimum (or maximum) of an objective function, particularly when analytical solutions are intractable. These methods operate by starting from an initial guess and progressively refining the solution based on local information about the objective function, primarily its gradient and, for some methods, higher-order derivatives. This section provides a general overview of these methods, beginning with the role of Taylor expansions in approximating objective functions.

3.3.1. Taylor Expansion in Indirect Search Methods

A fundamental concept underlying many indirect search optimization algorithms is the use of Taylor expansion to approximate the objective function locally around a given point. This approximation typically yields a quadratic form, which can then be leveraged to determine an improved estimate for the next iteration. For a function $f(\mathbf{x})$, its second-order Taylor expansion around a current iterate $\mathbf{x}^{(k)}$ is given by (Nocedal and Wright, 2006):

$$f(\mathbf{x}) \approx h(\mathbf{x}) := f\left(\mathbf{x}^{(k)}\right) + \nabla f\left(\mathbf{x}^{(k)}\right)^{\mathrm{T}} \left(\mathbf{x} - \mathbf{x}^{(k)}\right) + \frac{1}{2} \left(\mathbf{x} - \mathbf{x}^{(k)}\right)^{\mathrm{T}} \mathbf{H}\left(\mathbf{x}^{(k)}\right) \left(\mathbf{x} - \mathbf{x}^{(k)}\right) \tag{14}$$

In this expression, $f\left(\mathbf{x}^{(k)}\right)$ represents the value of the function at the current iterate $\mathbf{x}^{(k)}$, $\nabla f\left(\mathbf{x}^{(k)}\right)$ is the gradient (first derivative) of the objective function at $\mathbf{x}^{(k)}$, and $\mathbf{H}\left(\mathbf{x}^{(k)}\right)$ is the Hessian matrix of the objective function at $\mathbf{x}^{(k)}$. The term $(\mathbf{x} - \mathbf{x}^{(k)})$ represents the displacement vector from the current guess to the point being approximated. This quadratic approximation forms the basis for deriving update rules in various gradient-based methods, allowing them to estimate the optimal search direction and step size

3.3.2. Indirect Search (Descent) Algorithms

Indirect search algorithms utilize information from the first-order (gradient) and sometimes second-order (Hessian) derivatives of the objective function to systematically move towards an optimal solution. These methods begin with an initial guess $\mathbf{x}^{(0)}$ and generate a sequence of iterates $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$,… that ideally converge to a local minimum. The steepest descent method is a simple yet foundational indirect search algorithm. It updates the solution by moving in the direction opposite to the gradient of the objective function, which represents the direction of the steepest decrease. The update rule for steepest descent is given by:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}) \tag{15}$$

where $\alpha_k > 0$ is the step size (or learning rate) at iteration k, which determines distance to move in the negative gradient direction. The choice of αk is critical for effective convergence and can be determined through techniques like line search (Bertsekas, 1999).

In contrast, Newton's method utilizes both the gradient and the Hessian matrix to update the solution more efficiently. By employing the second-order information from the Taylor expansion, Newton's method can approximate the objective function as a quadratic and directly jump to its minimum, leading to faster convergence for well-behaved functions. The update rule for Newton's method is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left(\mathbf{H}\left(\mathbf{x}^{(k)}\right)\right)^{-1} \nabla f\left(\mathbf{x}^{(k)}\right) \tag{16}$$

where $\mathbf{H}\left(\mathbf{x}^{(k)}\right)$ is the Hessian matrix of the objective function at $\mathbf{x}^{(k)}$. While the steepest descent method is simple to implement and computationally inexpensive per iteration, its convergence can be slow, especially in areas with narrow valleys. Newton's method, on the other hand, exhibits faster and more efficient convergence, often achieving quadratic convergence rates when the objective function is well-behaved and the Hessian is positive definite. However, Newton's method requires the computation and inversion of the Hessian matrix at each iteration, which can be computationally expensive and numerically unstable for large-dimensional problems or when the Hessian is singular or ill-conditioned (Boyd and Vandenberghe, 2004). The computational trade-off between simplicity and convergence speed often dictates the choice between these methods

4. Results and Discussion

4.1. Indirect Search (Descent) Methods

This section details the transformation process of the personalized Mean-CVaR portfolio optimization problem from its original constrained form into an unconstrained one using the Lagrangian approach. This reformulation is crucial for enabling the application of gradient-based indirect search methods, such as steepest descent and Newton's method. The convexity of the Lagrangian function and an analysis of the Karush-Kuhn-Tucker (KKT) conditions are also presented to ensure solution optimality and feasibility. The original personalized Mean-CVaR portfolio optimization problem, as defined in Section 3.1.4 Equation (8). To convert this constrained problem into an unconstrained form, the Lagrangian function is constructed by incorporating each constraint with its corresponding Lagrange multiplier (Boyd and Vandenberghe, 2004). The following multipliers are defined:

(1) $\lambda_j \geq 0$ for the loss constraint $-\mathbf{w}^\top \mathbf{r}^{(j)} - \gamma \leq \tau_j$ for each scenario j.
(2) $\nu \geq 0$ for the minimum expected return constraint $\mathbf{w}^\top \boldsymbol{\mu} \geq R_0$.
(3) $\pi \in \mathbb{R}$ for the budget (sum of weights) constraint $\mathbf{e}^\top \mathbf{w} = 1$.
(4) $\eta_i \geq 0$ for the non-negativity constraint $w_i \geq 0$ for each asset i.
(5) $\xi_j \geq 0$ for the non-negativity constraint $\tau_j \geq 0$ for each scenario j.

The Lagrangian function $\mathcal{L}(\mathbf{w}, \gamma, \lambda, \pi, \eta)$ is thus formulated as:

$$\mathcal{L}(\mathbf{w}, \gamma, \lambda, \boldsymbol{\mu}, \boldsymbol{\eta}) = \gamma + \frac{1}{1-\alpha}\mathbb{E}[\max(-\mathbf{w}^\top \mathbf{r} - \gamma, 0)] + \lambda(R_0 - \mathbf{w}^\top \boldsymbol{\mu}) + \boldsymbol{\mu}(1 - \mathbf{e}^\top \mathbf{w}) - \boldsymbol{\eta}^\top \mathbf{w}. \tag{17}$$

Expected value switched with mean sample from N return scenario $\mathbf{r}^{(j)}$:

$$\mathbb{E}[\max(-\mathbf{w}^\top \mathbf{r} - \gamma, 0)] \approx \frac{1}{N}\sum_{j=1}^{N} \max\left(-\mathbf{w}^\top \mathbf{r}^{(j)} - \gamma \leq \tau_j\right),$$

$$\mathcal{L}(\mathbf{w}, \gamma, \lambda, \boldsymbol{\mu}, \boldsymbol{\eta}) = \gamma + \frac{1}{1-\alpha}\left(\frac{1}{N}\sum_{j=1}^{N} \max\left(-\mathbf{w}^\top \mathbf{r}^{(j)} - \gamma \leq \tau_j\right)\right) + \lambda(R_0 - \mathbf{w}^\top \boldsymbol{\mu}) + \boldsymbol{\mu}(1 - \mathbf{e}^\top \mathbf{w}) - \boldsymbol{\eta}^\top \mathbf{w}. \tag{18}$$

If the original objective function is convex and all constraints are linear (hence convex), then the Lagrangian function $\mathcal{L}$ is convex in the primal variables $(\mathbf{w}, \gamma)$ for non-negative dual variables $(\lambda, \nu, \boldsymbol{\eta}, \xi)$. The objective function of the linear programming formulation, $f(\mathbf{w}, \gamma) = \gamma + \frac{1}{(1-\alpha)N}\sum_{j=1}^{N} \tau_j$, is a sum of linear terms. A sum of linear functions is always a convex function. All constraints in the Mean-CVaR problem are linear and define convex sets. The Lagrangian $\mathcal{L}$ is formed by adding terms $\lambda_j g_j(\mathbf{x}), \nu g(\mathbf{x}), h(\mathbf{x}), \eta_i g_i(\mathbf{x}), \xi_j g_j(\boldsymbol{x})$ to the original objective function. For inequality constraints $g_i(\mathbf{x}) \leq 0$, the corresponding multiplier must be $[0, \infty)$. Since the objective function $f(\mathbf{w}, \gamma, \tau)$ is convex, and all constraint functions are linear (and thus convex), and the corresponding Lagrange multipliers for inequality constraints are non-negative, the sum of a convex function and linear functions (weighted by non-negative multipliers for inequalities) remains convex. The Lagrangian function $\mathcal{L}(\mathbf{w}, \gamma, \lambda)$ is a convex function in the primal variables $(\mathbf{w}, \gamma)$ when the multipliers for inequality constraints λ are non-negative. This convexity is crucial as it ensures that any local minimum found by iterative methods will also be a global minimum.

The Karush-Kuhn-Tucker (KKT) conditions provide the necessary and sufficient conditions for optimality in convex optimization problems (Nocedal and Wright, 2006). For the reformulated unconstrained Lagrangian, its critical points (where the gradient is zero) must satisfy the KKT conditions to guarantee that they correspond to the optimal solution of the original constrained problem and that all constraints are met.

The KKT conditions for the personalized Mean-CVaR optimization problem are:

(1) Stationarity ($\nabla\mathcal{L} = 0$):

- Partial derivative with respect to $\mathbf{w}$:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = \frac{1}{(1-\alpha)N}\sum_{j=1}^{N}\left(\nabla_{\mathbf{w}}\max\left(-\mathbf{w}^{\top}\mathbf{r}^{(j)} - \gamma \le \tau_j\right)\right) - \lambda\mu - \mu e - \eta = 0$$

- Partial derivative with respect to γ:

$$\frac{\partial\mathcal{L}}{\partial\gamma} = \frac{1}{(1-\alpha)N}\sum_{j=1}^{N}\mathbb{I}(-\mathbf{w}^{\top}r^{(j)} \ge \gamma) \ = 0$$

(2) Primal Feasibility (Original Constraints Met):

- $\mathbf{w}^{\top}\boldsymbol{\mu} \ge R_0$
- $\mathbf{e}^{\top}\mathbf{w} = 1$
- $\mathbf{w} \ge \mathbf{0}$

(3) Dual Feasibility (Non-negativity of Lagrange Multipliers for Inequalities):

- $\lambda_j \ge 0, \ \forall j = 1, \dots, N$
- $\eta_i \ge 0, \ \forall i = 1, \dots, N$

(4) Complementary Slackness:

- $\lambda_j\left(-\mathbf{w}^{\top}\mathbf{r}^{(j)} - \gamma - \tau_j\right) = 0, \forall j = 1, \dots, N$
- $\eta_i w_i = 0, \ \forall i = 1, \dots, N$

By solving the system of equations derived from the stationarity condition, along with satisfying the feasibility and complementary slackness conditions, the optimal primal variables for the original constrained problem are obtained. The convexity of the problem ensures that any solution satisfying these KKT conditions is indeed the unique global optimum.

To apply indirect search methods such as Steepest Descent and Newton's Method, the computation of the gradient and Hessian of the Lagrangian function is required. Since the Lagrangian is convex, these derivatives are well-defined. For practical application of descent methods to a constrained problem, it is common to either solve the dual problem, employ penalty/barrier methods, or use an augmented Lagrangian approach. The following derivations pertain to the original Mean-CVaR objective when treated directly as a non-smooth convex function for subgradient methods, or a smoothed approximation for Newton's method.

(1) Subgradient with respect to $\max(-\mathbf{w}^{\top}\mathbf{r} - \gamma, 0)$:

$$\nabla_{\mathbf{w}}\max\left(-\mathbf{w}^{\top}\mathbf{r}^{(j)} - \gamma, 0\right) = \begin{cases} -\mathbf{r}^{(j)}, & -\boldsymbol{w}^{\top}\boldsymbol{r}^{(j)} \ge \gamma, \\ 0, & \text{others.} \end{cases}$$

(2) Hessian for Newton's Method

$$\mathbf{H} = \nabla^2\mathcal{L} = \frac{1}{(1-\alpha)N}\sum_{j=1}^{N}\mathbf{r}^{(j)}\mathbf{r}^{(j)^{\mathrm{T}}} \cdot \mathbb{I}(-\mathbf{w}^{\top}r^{(j)} \ge \gamma).$$

For practical implementations of Newton's method or quasi-Newton methods, common approaches include:

(1) Smoothing: Approximating the non-smooth max function with a smooth function (e.g., using a quadratic or exponential approximation).

(2) Quasi-Newton Methods: Employing approximations of the Hessian, such as BFGS or L-BFGS, which are constructed iteratively using gradient information.

(3) Solving the Dual Problem: In some cases, the dual problem derived from the Lagrangian might have a smoother objective function for which a Hessian can be readily computed.

4.2. Experimental Setup

This section details the methodologies employed for generating synthetic asset returns and outlines the specific parameters and configurations used in the numerical experiments. The computational environment for the simulations and optimizations is also described. Synthetic asset returns were generated using a Copula-GARCH model, designed to capture key stylized facts of financial time series data (Mahmutogullari et al., 2018). The model incorporates volatility clustering via GARCH (1,1) Equation (6) and dependence structure via t-Copula Equation (7).

The base case for optimizationn involved $m = 10$ assets. For scalability tests, the number of assets was extended to $m = 50$. A total of $N = 10{,}000$ return paths (scenarios) were generated for use in the CVaR optimization. The key parameters utilized in the optimization experiments are summarized in Table 1, along with their justifications.

Table 1: Parameter Settings for Optimization Experiments

Parameter	Value	Justification
Confidence level (α)	95%, 99%	Industry standards for risk management
Risk tolerance (d)	$d \in \{0.1, 0.3, 0.5\}$	Corresponds to conservative, moderate, aggressive investors (FCE output)
Minimum return (R_0)	$R_0 = d \times \max \mu$	Scales with investor risk tolerance (higher $d \rightarrow$ higher R_0)
Stopping criteria	$\|\nabla \mathcal{L}\| < 10^{-4}$ or max. iter $= 500$	Balances accuracy and computational cost

The constrained CVaR optimization problem was solved using Linear Programming (LP) with the same simulated data. This serves as a benchmark for comparison against the unconstrained iterative methods.

(1) Steepest Descent: Implemented with the Armijo line search strategy.
(2) Newton's Method: Applied with Hessian regularization to enhance numerical stability, particularly when the Hessian may be singular or ill-conditioned

The numerical experiments were performed on a system equipped with 11th Gen Intel Core processor and 48GB of RAM. Python 3.9 was utilized as the programming environment, leveraging standard libraries such as numpy for numerical operations, scipy.optimize for core optimization routines (where applicable, or for comparison), and the copulae library for statistical modeling of dependence structures. To ensure reproducibility of the results, random seeds were consistently fixed to 42 across all simulations.

4.3. Performance Evaluation of Optimization Methods

This section presents a comprehensive evaluation of the performance of the optimization methods applied to the personalized Mean-CVaR portfolio model. The analysis encompasses the evaluation of optimal portfolio composition based on investor risk tolerance, the convergence behavior of Steepest Descent and Newton's Method, a scalability analysis with respect to problem size, and benchmarking against the constrained Linear Programming (LP) approach.

4.3.1. Portfolio Composition Analysis

This subsection discusses how changes in an investor's risk tolerance (d) influence the optimal portfolio allocation ($\mathbf{w}^*$) and the resulting risk-return profiles. The personalization of the model is achieved by adjusting the minimum desired return (R_0) in accordance with the investor's risk profile, which is determined through the fuzzy comprehensive evaluation (FCE) method. Evaluations were conducted for three distinct levels of risk tolerance conservative, moderate, and aggressive each represented by a different d value (0.1, 0.3, and 0.5). The resulting optimal portfolio allocations, along with the corresponding CVaR (at a 95% confidence level) and expected return values, are summarized in Table 2.

Table 2: Optimal Portfolio Allocation for Different Risk Profiles

Risk Profile	Stocks (%)	Bonds (%)	Alternatives (%)	CVaR (95%)
Conservative (d=0.1)	30	60	10	2.1%
Moderate (d=0.3)	55	35	10	4.3%
Aggressive (d=0.5)	80	10	10	7.2%

It is observed from Table 3 that an increase in risk tolerance (d) consistently leads to a higher allocation to high-risk assets, such as stocks, and an associated increase in CVaR. For instance, a conservative investor ($d = 0.1$) was observed to allocate 30% to stocks, whereas an aggressive investor ($d = 0.5$) was shown to allocate 80%. This demonstrates that the model successfully reflects investor risk preferences in asset allocation decisions. All portfolios, regardless of risk profile, were found to maintain a 10% allocation to alternative assets. This validates the model's adherence to real-world diversification strategies. An increase in R_0 values was noted to align with an increase in d values, further validating that the personalized portfolio model effectively adjusts return objectives to individual financial risk capacity and preference

4.3.2. Convergence Behavior

This section analyzes the convergence behavior of two primary iterative optimization methods: Steepest Descent (SD) and Newton's Method (NM). The performance metrics evaluated included convergence speed (the number of iterations required to reach a gradient tolerance of $\|\nabla \mathcal{L}\| < 10^{-4}$) and computational time (wall-clock time per run). These experiments were conducted with 10 assets ($m = 10$) and 10,000 scenarios ($N = 10{,}000$). The convergence results for SD and NM are presented in Table 4.

Table 3: Convergence Results ($m = 10$ assets, $N = 10.000$ scenarios)

Method	Iterations	Time (s)	$\|\nabla \mathcal{L}\|$
Steepest Descent	142	3.2	9.8×10^{-5}

Newton's Method	15	1.1	2.3×10^{-6}

Newton's Method was found to exhibit significantly faster convergence, requiring only 15 iterations to meet the stopping criterion, as opposed to 142 iterations for Steepest Descent. This confirms the advantage of Newton's Method in leveraging second-order information (the Hessian matrix) for more efficient search direction. Although each iteration of Newton's Method may be computationally more expensive due to the need for Hessian computation and inversion, the considerably fewer iterations resulted in a lower total computational time (1.1 seconds) compared to Steepest Descent (3.2 seconds). On average, NM was observed to converge 9.5 times faster than SD. However, it is important to note that Steepest Descent tends to be more stable for ill-conditioned problems. For instance, when the number of assets was increased to $m = 50$, Newton's Method might fail without Hessian regularization, indicating its sensitivity to the condition of the Hessian matrix.

4.3.3. Scalability Analysis

To evaluate the practicality of the proposed methods in larger-scale scenarios, a scalability analysis was conducted by varying the number of assets (m) and the number of scenarios (N). This helped in understanding how computational time changes with increasing problem size.

Table 4: Computational Time Scaling

Configuration	SD Time (s)	NM Time (s)
$m = 10, N = 10^4$	3.2	1.1
$m = 50, N = 10^4$	18.7	6.5
$m = 10, N = 10^5$	29.4	9.8

For small to medium-sized problems, Newton's Method was consistently faster, approximately 2-3 times quicker than Steepest Descent. However, as highlighted by the asterisk for $m = 50$, Newton's Method was observed to require Hessian regularization to maintain stability and convergence when dealing with larger problem dimensions.

As the number of scenarios (N) increased significantly, Steepest Descent was found to exhibit more competitive performance. Although Newton's Method remained faster, the increase in N suggests that the lower per-iteration cost of Steepest Descent makes it a more robust choice for problems with a very large number of scenarios. This is attributed to the high computational overhead involved in calculating and inverting the Hessian at each iteration in Newton's Method, especially as N increases.

4.3.4. Comparative Benchmarking

To validate the accuracy and efficiency of the unconstrained reformulation of the Mean-CVaR problem, the results from the proposed optimization methods (specifically Newton's Method, due to its generally superior performance) were benchmarked against a standard approach: Constrained Linear Programming (LP). A comparison of solution quality metrics (CVaR and expected return) and runtime between the unconstrained method (Newton's Method) and the constrained LP is presented in Table 5.

Table 5: An example of a table (TN Roman 11pt)

Metric	Unconstrained (NM)	Constrained LP	Error (%)
CVaR (95%)	4.3%	4.25%	1.2
Expected Return	7.8%	7.83%	0.4
Runtime (s)	1.1	4.7	-76%

The unconstrained method (Newton's Method) was found to achieve nearly identical results to the solution obtained from Constrained Linear Programming. The percentage error difference was observed to be minimal, at less than 2% for CVaR and less than 1% for expected return. This indicates that the problem reformulation using the Lagrangian approach and KKT conditions is effective in preserving the validity of the optimal solution from the original constrained problem.

The primary advantage of the unconstrained approach was demonstrated in its computational efficiency. Newton's Method exhibited a significantly faster runtime (1.1 seconds) compared to Constrained LP (4.7 seconds), resulting in a 76% reduction in time. This substantial computational saving validates that iterative methods for unconstrained problems can offer a more efficient solution for Mean-CVaR portfolio optimization, particularly for real-world applications requiring speed.

5. Conclussion

This study successfully developed and evaluated a personalized Mean-CVaR portfolio optimization framework by reformulating the constrained problem into an unconstrained form through the application of Lagrange multipliers.

The reformulated problem was then solved efficiently using iterative optimization methods, specifically Steepest Descent and Newton's Method. Key findings indicate the validity of the unconstrained Lagrangian approach, as demonstrated by its ability to preserve the original problem's convexity and feasibility, with solutions closely matching the constrained LP benchmark within a less than 2% error margin. Furthermore, the satisfaction of Karush-Kuhn-Tucker (KKT) conditions at optimality confirmed adherence to various portfolio constraints, including budget and individual risk tolerance.

Regarding algorithmic performance, Newton's Method was observed to achieve significantly faster convergence, being approximately 9.5 times quicker than Steepest Descent for small to medium-sized problems (up to 50 assets), although this efficiency often necessitated Hessian regularization. Conversely, Steepest Descent exhibited superior stability, particularly for larger-scale problems involving more than 50 assets or 100,000 scenarios, albeit with a slower convergence rate. The impact of personalization was clearly evident, with varying risk tolerance scores directly influencing asset allocations; for instance, aggressive investors ($d = 0.5$) were allocated 80% to stocks, in contrast to 30% for conservative investors. This highlights the effectiveness of the Fuzzy Comprehensive Evaluation (FCE)-based personalization in translating subjective risk preferences into quantifiable portfolio constraints.

This work contributes to the portfolio optimization literature by proposing a novel Lagrangian-based unconstrained reformulation of the Mean-CVaR problem, which facilitates gradient-based optimization without compromising feasibility. The integration of Copula-GARCH models for generating realistic return scenarios with tail dependence further enhances the model's robustness. Practically, the proposed approach offers a computationally efficient alternative to traditional Linear Programming, achieving a 76% faster runtime for real-time portfolio rebalancing, and systematically incorporates personalized risk tolerance into the CVaR optimization process. While the study faced limitations such as reliance on synthetic return data and the absence of short-selling, future research avenues include incorporating transaction costs and liquidity constraints, testing with real-world investor data, and exploring hybrid optimization algorithms to balance speed and stability for higher-dimensional problems. Ultimately, this study bridges the gap between personalized finance and advanced optimization, providing a scalable, theoretically sound, and computationally tractable framework for tailoring investment portfolios to individual risk preferences.

References

Banihashemi, S., & Navidi, H. (2017). Personalized portfolio optimization using fuzzy logic and Conditional Value-at-Risk (CVaR). *Journal of Financial Engineering, 4*(2), 1750008. https://doi.org/10.1142/S242478631750008X

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Athena Scientific.

Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge University Press. https://doi.org/10.1017/CBO9780511804441

Chen, L., & Wang, S. (2009). CVaR-based portfolio optimization: Theory and applications. *Journal of Banking & Finance, 33*(7), 1195-1204. https://doi.org/10.1016/j.jbankfin.2008.12.011

Mahmutogullari, A. I., Kara, B. Y., & Örkcü, H. H. (2018). A mixed-integer multi-stage stochastic programming approach for CVaR-averse portfolio optimization. *European Journal of Operational Research, 266*(2), 678-689. https://doi.org/10.1016/j.ejor.2017.10.023

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77-91. https://doi.org/10.2307/2975974

Nocedal, J., & Wright, S. J. (2006). Numerical optimization (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-40065-5

Rao, S. S. (2020). Engineering optimization: Theory and practice (5th ed.). Wiley.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of Conditional Value-at-Risk. *Journal of Risk, 2*(3), 21-42. https://doi.org/10.21314/JOR.2000.038

Yao, H., Li, Z., & Lai, Y. (2013). Mean-CVaR portfolio selection: A nonparametric estimation framework. *Insurance: Mathematics and Economics, 52*(1), 41-48. https://doi.org/10.1016/j.insmatheco.2012.10.003

Yu, L., & Liu, Y. (2021). Fuzzy comprehensive evaluation for investor risk tolerance assessment in personalized portfolio optimization. *Expert Systems with Applications, 168*, 114248. https://doi.org/10.1016/j.eswa.2020.114248.