

Decentralized Portfolio Optimization

Shayan Malekpour 810197661

December 2023

Problem Description

Consider a decentralized portfolio optimization problem with M investors and N assets. Each investor aims to maximize their expected portfolio return while considering risk and a global objective related to the average return across all portfolios.

The problem involves the following key components:

- **Investors:** There are M independent investors, each with its own risk profile, investment goals, and constraints.
- **Assets:** There are N financial assets available for investment.
- **Global Objective:** The global objective is to optimize the overall portfolio performance, considering both individual investor objectives and a collective goal related to the average return.
- **Local Decision Variables:** Each investor has a set of decision variables representing the weights assigned to different assets in their portfolio.
- **Local Cost Functions:** The local cost functions for each investor incorporate their individual objectives, risk tolerance, and a coupling term with the global variable representing the average return.

Centralized Portfolio Optimization Problem

Let:

- let $W \in \mathbb{R}^{N \times M}$ be the matrix of decision variables representing the portfolio weights, where W_{ij} is the weight of asset i in the portfolio of investor j .
- $R \in \mathbb{R}^{M \times N}$ be the matrix of simulated returns, where R_{ij} is the return of asset i for investor j .
- $\Sigma \in \mathbb{R}^{M \times N}$ be the matrix of simulated standard deviations (sigmas), where Σ_{ij} is the standard deviation of asset i for investor j .

- α be an adjustable parameter.

The centralized portfolio optimization problem can be formulated as follows:

$$\sum_{i=1}^M \left(-R_i^T W_i + 0.1 \sum_{j=1}^N W_{ij} \Sigma_{ij} W_{ij} + \alpha \sum_{j=1}^N \left| R_{ij}^T W_{ij} - \frac{1}{M} \sum_{k=1}^M R_{kj}^T W_{kj} \right| \right)$$

Subject to: $\sum_{i=1}^N W_{ij} = 1 \quad \text{for } j = 1, 2, \dots, M$

$W \succeq 0$ (element-wise non-negativity)

In this formulation, R_i denotes the vector of returns for investor i , and Σ_i denotes the vector of standard deviations for investor i . The objective function seeks to maximize the global objective, which incorporates the returns, sigmas, and a regularization term penalizing deviations from the average returns.

Decentralized Problem Description

Expected Portfolio Return ($R_i(W_i)$)

The expected portfolio return for investor i is a linear combination of the weights assigned to different assets in their portfolio:

$$R_i(W_i) = \sum_{j=1}^N W_{i,j} \cdot \text{Expected Return of Asset } j$$

Portfolio Risk ($\sigma_i(W_i)$)

The portfolio risk for investor i is represented by the standard deviation of the portfolio return, considering the covariance between assets:

$$\sigma_i(W_i) = \sqrt{\sum_{j=1}^N \sum_{k=1}^N W_{i,j} \cdot W_{i,k} \cdot \text{Covariance}(j, k)}$$

Coupling Term ($|R_i(W_i) - \bar{R}|$)

The coupling term represents the absolute difference between the expected portfolio return of investor i and the average return across all portfolios:

$$|R_i(W_i) - \bar{R}| = \left| R_i(W_i) - \frac{1}{M} \sum_{j=1}^M R_j(W_j) \right|$$

Global Variable (\bar{R})

The global variable represents the average portfolio return across all investors and is updated iteratively:

$$\bar{R} = \frac{1}{M} \sum_{i=1}^M R_i(W_i)$$

Primal Decomposition Method

The primal decomposition method is employed to address the decentralized portfolio optimization problem. The key steps are as follows:

1. **Decompose the Global Objective:** Express the global objective as the sum of individual local cost functions:

$$J_{\text{Global}} = \sum_{i=1}^M J_i(W_i)$$

2. **Decompose Individual Local Cost Functions:** Express each local cost function for investor i as a combination of their local decision variables and a coupling term involving the global variable:

$$J_i(W_i) = -R_i(W_i) + \lambda \cdot \sigma_i(W_i) + \alpha \cdot |R_i(W_i) - \bar{R}|$$

3. **Optimize Locally:** Each investor independently optimizes their local cost function, focusing on their individual objectives and constraints.
4. **Update Global Variable:** After each iteration, update the global variable (average return) based on individual portfolio returns:

$$\bar{R} = \frac{1}{M} \sum_{i=1}^M R_i(W_i)$$

5. **Calculate Coupling Term:** Calculate the coupling term in each local cost function using the updated global variable.
6. **Minimize Local Cost Functions with Respect to Global Variable:** Theoretical calculation to minimize each local cost function with respect to the global variable, ensuring alignment with the collective objective.
7. **Repeat:** Iterate the process until convergence, allowing each investor to refine their portfolio allocation while considering the global performance.

0.1 Results

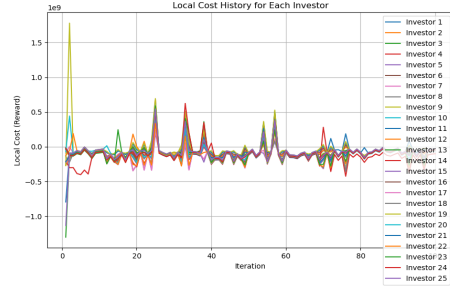
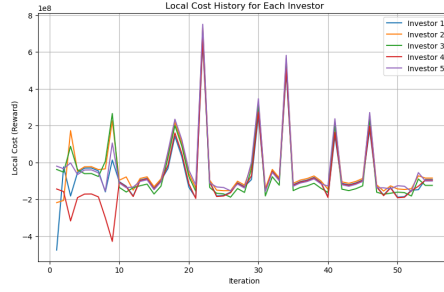


Figure 1: 5 agent investing in 100 as- Figure 2: 25 agent investing in 100 as-
sets. sets

Due to the Limitations of resource and computation power, and the larger scale of calculations needed for the *scipy-optimize* the problem was not solvable for more than 25 investors in this section only.

ADMM Algorithm for Decentralized Portfolio Optimization

The Alternating Direction Method of Multipliers (ADMM) is applied to solve the decentralized portfolio optimization problem. The algorithm iteratively updates the local decision variables and a consensus variable, enforcing consensus among the investors.

Problem Formulation

Consider the decentralized portfolio optimization problem with the following cost function for investor i :

$$J_i(W_i) = - \sum_{j=1}^N W_{i,j} \cdot \text{Expected Return of Asset } j + 0.1 \cdot \sqrt{\sum_{j=1}^N \sum_{k=1}^N W_{i,j} \cdot W_{i,k} \cdot \text{Covariance}(j,k)} + \alpha \cdot |R_i(W_i) - \bar{R}|$$

Where:

- W_i is the vector of decision variables representing the portfolio weights for investor i .
- The first term represents the expected portfolio return for investor i .
- The second term represents the portfolio risk for investor i .
- The third term is the coupling term involving the global variable \bar{R} , which is the average return across all portfolios.
- α is a parameter controlling the impact of the coupling term.

Steps

Step 1: Augmented Lagrangian Formulation

The augmented Lagrangian for the decentralized portfolio optimization problem is given by:

$$L_\rho(W, Z, U) = \sum_{i=1}^M J_i(W_i) + \frac{\rho}{2} \sum_{i=1}^M \|W_i - Z + U_i\|^2$$

Where:

- Z is the consensus variable.
- U is the vector of dual variables (Lagrange multipliers).
- ρ is the penalty parameter.

Step 2: ADMM Update Steps

- **Update W_i :**

$$W_i^{k+1} = \arg \min_{W_i} \left(J_i(W_i) + \frac{\rho}{2} \|W_i - Z^k + U_i^k\|^2 \right)$$

- **Update Z :**

$$Z^{k+1} = \frac{1}{M} \sum_{i=1}^M (W_i^{k+1} + U_i^k)$$

- **Update U_i :**

$$U_i^{k+1} = U_i^k + (W_i^{k+1} - Z^{k+1})$$

Step 3: Convergence Check

Check for convergence based on the primal and dual residuals:

$$r^k = \frac{1}{M} \sum_{i=1}^M \|W_i^{k+1} - Z^{k+1}\|, \quad s^k = \rho \|Z^{k+1} - Z^k\|$$

The algorithm converges when $\max(r^k, s^k) < \epsilon$ (where $\epsilon = \max(\epsilon_r, \epsilon_s)$).

Step 4: Iterative Updates

Iterate the ADMM update steps until convergence is achieved:

$$W_i^{k+1} = \arg \min_{W_i} \left(J_i(W_i) + \frac{\rho}{2} \|W_i - Z^k + U_i^k\|^2 \right)$$

$$Z^{k+1} = \frac{1}{M} \sum_{i=1}^M (W_i^{k+1} + U_i^k)$$

$$U_i^{k+1} = U_i^k + (W_i^{k+1} - Z^{k+1})$$

Step 5: Output

The solution W_i^{k+1} represents the optimized portfolio weights for each investor.

Notes

- Adjust the specific form of the local cost function $J_i(W_i)$ based on the characteristics of your portfolio optimization problem.
- The choice of penalty parameter ρ affects the convergence and solution accuracy. Experiment with different values.
- Convergence tolerances ϵ_r and ϵ_s control the convergence criterion. Smaller values lead to stricter convergence criteria.

0.2 Result

According to the parameters of the ADMM algorithm the convergence changes. Following results are for the $\rho = 1$ & $\alpha = 0.1$

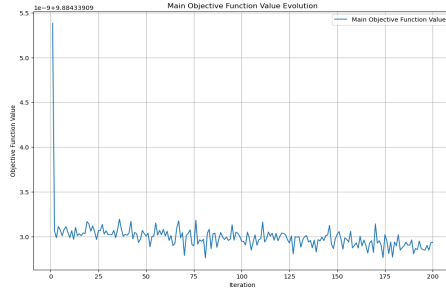


Figure 3: Convergence of ADMM algorithm for 5 investors

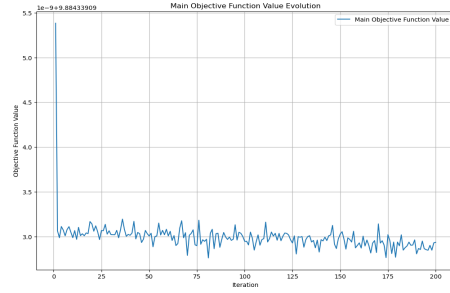


Figure 4: Convergence of ADMM algorithm for 15 investors

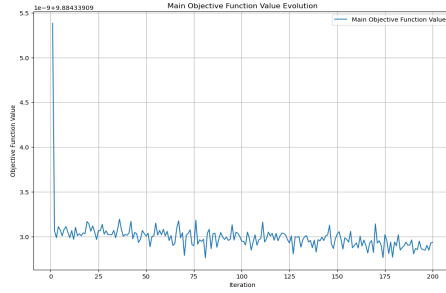


Figure 5: Convergence of ADMM algorithm for 25 investors

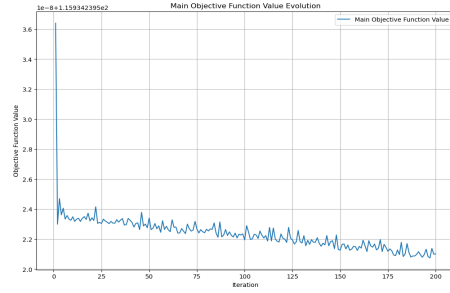


Figure 6: Convergence of ADMM algorithm for 50 investors

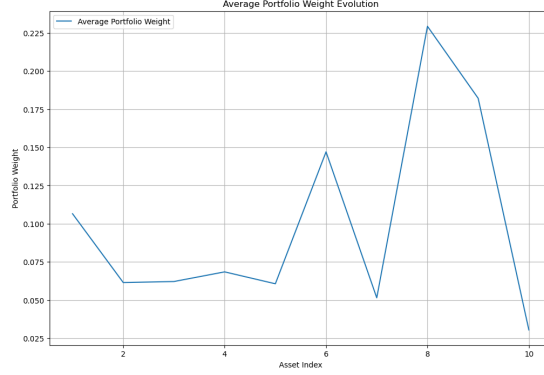


Figure 7: Average distribution of portfolios

Fast ADMM Algorithm for Decentralized Portfolio Optimization

The Fast Alternating Direction Method of Multipliers (Fast ADMM) is a modification of the standard ADMM algorithm designed to accelerate convergence. The key idea is to utilize variable splitting and apply specialized updating rules for improved efficiency.

Variable Splitting

The Fast ADMM introduces variable splitting, creating auxiliary variables for some of the original variables. This allows for faster updates and improved convergence rates.

Consider the decentralized portfolio optimization problem with variable splitting:

$$\begin{aligned}
J_i(W_i, P_i) = & - \sum_{j=1}^N W_{i,j} \cdot \text{Expected Return of Asset } j + \\
& 0.1 \cdot \sqrt{\sum_{j=1}^N \sum_{k=1}^N W_{i,j} \cdot W_{i,k} \cdot \text{Covariance}(j, k)} + \alpha \cdot |R_i(W_i) - \bar{R}| + \\
& \frac{\rho}{2} \sum_{j=1}^N (P_{i,j} - W_{i,j} + Z_j - U_{i,j})^2
\end{aligned}$$

Here, P_i is the auxiliary variable associated with the portfolio weights W_i .

Fast ADMM Update Steps

The Fast ADMM update steps involve specialized rules for updating the original variables and auxiliary variables efficiently.

- **Update W_i and P_i :**

$$\begin{aligned} W_i^{k+1} &= \arg \min_{W_i} J_i(W_i, P_i^k) \\ P_i^{k+1} &= \arg \min_{P_i} J_i(W_i^{k+1}, P_i) \end{aligned}$$

- **Update Z :**

$$Z^{k+1} = \frac{1}{M} \sum_{i=1}^M (W_i^{k+1} + U_i^k)$$

- **Update U_i :**

$$U_i^{k+1} = U_i^k + (W_i^{k+1} - Z^{k+1})$$

Convergence Check

The convergence check remains the same as in the standard ADMM:

$$r^k = \frac{1}{M} \sum_{i=1}^M \|W_i^{k+1} - Z^{k+1}\|, \quad s^k = \rho \|Z^{k+1} - Z^k\|$$

The algorithm converges when $\max(r^k, s^k) < \epsilon$ (where $\epsilon = \max(\epsilon_r, \epsilon_s)$).

Iterative Updates

Iterate the Fast ADMM update steps until convergence is achieved:

$$\begin{aligned} W_i^{k+1} &= \arg \min_{W_i} J_i(W_i, P_i^k) \\ P_i^{k+1} &= \arg \min_{P_i} J_i(W_i^{k+1}, P_i) \\ Z^{k+1} &= \frac{1}{M} \sum_{i=1}^M (W_i^{k+1} + U_i^k) \\ U_i^{k+1} &= U_i^k + (W_i^{k+1} - Z^{k+1}) \end{aligned}$$

0.3 Results

According to the parameters of the fast ADMM algorithm the convergence changes. Following results are for the $\rho = 0.2$ & $\alpha = 1$.

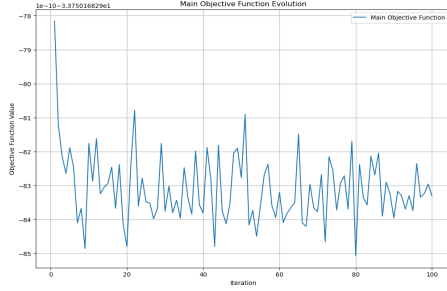


Figure 8: Convergence of fastADMM algorithm for 5 investors

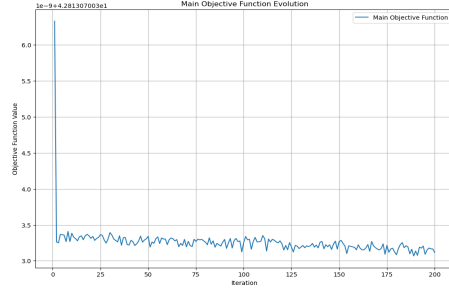


Figure 9: Convergence of fastADMM algorithm for 15 investors

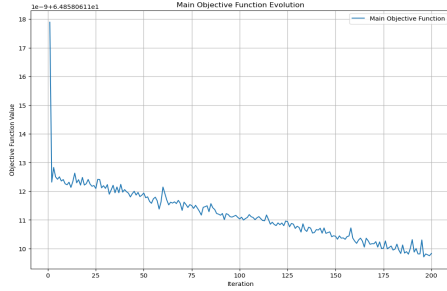


Figure 10: Convergence of fastADMM algorithm for 25 investors

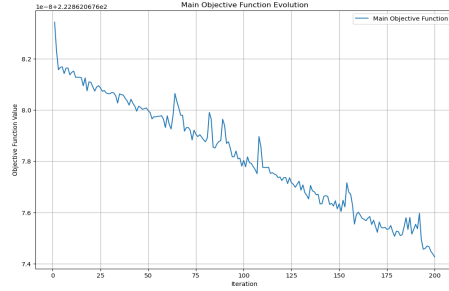


Figure 11: Convergence of fastADMM algorithm for 50 investors

Conclusion

In this study, we explored decentralized portfolio optimization problems with a focus on addressing the challenges posed by multiple independent investors. Two different optimization approaches, Primal Decomposition Method and Alternating Direction Method of Multipliers (ADMM), were employed to find solutions for the decentralized problem. Additionally, a variant of ADMM, the Fast ADMM, was introduced to enhance convergence speed.

Centralized Portfolio Optimization

The centralized portfolio optimization problem was formulated with the objective of maximizing the overall portfolio performance, considering both individual investor objectives and a collective goal related to the average return. The optimization problem involved decision variables representing the portfolio weights, simulated returns, simulated standard deviations, and an adjustable parameter.

The objective function incorporated the returns, standard deviations, and a regularization term penalizing deviations from the average returns. Constraints were imposed to ensure that the sum of portfolio weights for each investor equaled 1 and that the weights were non-negative.

Primal Decomposition Method

The Primal Decomposition Method decomposed the global objective into individual local cost functions for each investor. Each local cost function included terms related to the expected portfolio return, portfolio risk, and a coupling term involving the global variable (average return).

The optimization process involved iterative updates where each investor independently optimized their local cost function, and the global variable (average return) was updated based on individual portfolio returns. Theoretical calculations were performed to minimize each local cost function with respect to the global variable, ensuring alignment with the collective objective.

ADMM Algorithm

The ADMM algorithm was applied to solve the decentralized portfolio optimization problem. The algorithm iteratively updated local decision variables and a consensus variable, enforcing consensus among the investors. The decentralized problem was formulated with the cost function for each investor, considering expected portfolio return, portfolio risk, and a coupling term involving the global variable.

The ADMM algorithm involved augmented Lagrangian formulation, variable splitting, and specialized updating rules for improved efficiency. The iterative update steps included optimizing local decision variables, updating the consensus variable, and adjusting dual variables.

Fast ADMM Algorithm

The Fast ADMM algorithm, a modification of the standard ADMM, introduced variable splitting to accelerate convergence. It created auxiliary variables for some of the original variables, allowing for faster updates and improved convergence rates.

The update steps for the Fast ADMM algorithm involved optimizing both original variables and auxiliary variables, updating the consensus variable, and adjusting dual variables. The convergence check remained similar to the standard ADMM.