Portfolio Optimization Using Metaheuristic Algorithms

Artificial Intelligence Check Point

Carlos Sanches, up202107694
João Rebelo, up202107209
Gonçalo Sousa, up202207320

Problem Specification

Portfolio Optimization Tool
Objective: Develop a robust tool to optimize investment portfolios using metaheuristic algorithms.
Key Features:

- Determine optimal asset allocation to maximize returns while minimizing risk.
- Support multiple optimization algorithms (Hill Climbing, Simulated Annealing, Tabu Search, Genetic Algorithm).
- Allow customizable constraints (max/min weights, risk aversion, short selling).
- Provide interactive visualization of optimization results.

Expected Outcome: A user-friendly application that produces efficient portfolios based on historical data from Australian Stock Market (2000-2020).

Related Work

Literature Review

Modern Portfolio Theory (MPT):

- Markowitz, H. (1952). "Portfolio Selection." Journal of Finance, 7(1), 77-91.
- Introduced mean-variance optimization framework for portfolio construction.

Metaheuristic Approaches:

- Chang, T.J., et al. (2000). "Heuristics for cardinality constrained portfolio optimisation." Computers & Operations Research, 27(13), 1271-1302.
- Cura, T. (2009). "Particle swarm optimization approach to portfolio optimization." Nonlinear Analysis: Real World Applications, 10(4), 2396-2406.

Dataset and Software implementation

- Australian Historical Stock Prices Dataset and Notebooks.
- `https://www.kaggle.com/datasets/ashbellett/australian-historical-stock-prices/data`

Problem Formulation

Portfolio Optimization as an Optimization Problem

Solution Representation:

- Vector $w = (w_1, w_2, ..., w_n)$ representing portfolio weights for n assets.
- Constraint: $\sum_{i=1}^{n} w_i = 1$.

Objective Function:

$$U(w) = E[R_p] - \lambda \cdot \sigma_p^2$$

Where:

- $E[R_p] = \sum_{i=1}^{n} w_i \cdot \mu_i$ (expected portfolio return).
- $\sigma_p^2 = w^T \Sigma w$ (portfolio variance).
- λ is risk aversion parameter.

Maximize $U(w)$ or Sharpe ratio $\frac{E[R_p]}{\sigma_p}$.

Constraints:

- Weight bounds: $0 \leq w_i \leq max\_weight$ (without short selling).
- With short selling: $-1 \leq w_i \leq 1$.
- Minimum return: $E[R_p] \geq min\_return$.
- Maximum risk: $\sigma_p \leq max\_risk$.

Algorithms and Operators

Metaheuristic Approach

Hill Climbing:

- Neighborhood function: Random perturbation of weights with step size.
- Restart mechanism to avoid local optima.

Simulated Annealing:

- Temperature schedule: Initial_temp $\times$ Cooling_rate^{iteration}.
- Acceptance probability: $e^{\frac{\Delta U}{T}}$ for negative utility changes.

Tabu Search:

- Tabu list: Recent weight configurations to avoid cycling.
- Aspiration criterion: Accept tabu moves if they improve best solution.

Genetic Algorithm:

- Crossover: Weighted average of parent solutions.
- Mutation: Random perturbation with mutation rate.
- Selection: Tournament selection with elitism.

Implementation Details

Programming Language: Python 3.

Development Environment:

- GUI: Tkinter framework.
- Visualization: Matplotlib for interactive plots.

Key Data Structures:

- pandas.DataFrame for historical price data and returns calculation.
- NumPy arrays for efficient matrix operations.

Current Progress:

- Complete GUI implementation with data loading capabilities.
- Hill Climbing algorithm implementation.
- Performance visualization (portfolio pie chart, allocation details risk rate, annual return etc).
- Data preprocessing and portfolio metrics calculation.

Next Steps:

- Implement remaining optimization algorithms.
- Add comparative performance evaluation.
- Improve data visualization and insights about the portfolio.