CONSTRAINED OPTIMIZATION WITH THE S&P 500 DATASET

Objective

In this project, you will formulate a constrained optimization problem using S&P 500 stock price data and solve it with at least three different optimization algorithms, then compare their performance. The focus is not financial forecasting; it is the behavior, efficiency, and comparison of optimization algorithms under realistic constraints.

Dataset

Kaggle Source: https://www.kaggle.com/datasets/camnugent/sandp500

Data Preparation Steps

1. Use the daily price data file (e.g., the file containing daily prices such as all_stocks_5yr.csv if present). Select one price field consistently (**Close** or **Adj Close**) and state clearly which one you use. If **Adj Close** is not available, use **Close**.

2. Choose a fixed analysis period and report it (start/end dates).

3. Define each asset by its ticker symbol. Filter out assets with incomplete histories over the selected period (remove any asset with missing price values). If duplicate tickers exist, keep the version with the longest complete history and state this rule. Report the final number of assets N.

4. Compute daily log-returns:

$$r_{i,t} = \ln\left(\frac{P_{i,t}}{P_{i,t-1}}\right)$$

Problem Definition

Decision Variables: Portfolio weights w_i

Constraints

- $\sum_i w_i = 1$(fully invested)
- $w_i \geq 0$(long-only)
- $w_i \leq 0.05$(per-asset upper bound to ensure diversification)

Objective Function

$$\min_{w} \quad \lambda \cdot \mathrm{Risk}(w) \; - \; (1-\lambda) \cdot \mathrm{Return}(w) \; + \; \gamma \cdot \mathrm{Turnover}(w)$$

Definitions (estimated on the training window)

- **Return:** $\mathrm{Return}(w) = w^{\top}\mu$, where $\mu$is the vector of mean returns.
- **Risk:** $\mathrm{Risk}(w) = w^{\top}\Sigma w$, where $\Sigma$is the covariance matrix.
- **Turnover:** $\mathrm{Turnover}(w) = \| w_t - w_{t-1} \|_1$, where $w_{t-1}$is the previous rebalance portfolio and $w_t$is the current one.
- **Initialization:** For the very first rebalance step, set $w_{t-1}$to the Equal Weight portfolio $w_i = 1/N$.

Parameters: $\lambda \in [0,1]$and $\gamma \geq 0$. Explain and justify your choices in the paper.

Algorithms

Use at least three optimization algorithms, including:

1. **Projected Gradient Descent (PGD):** Enforce constraints via projection.
   - Note: The L_1norm in the turnover penalty is non-differentiable. You must use a smoothing technique (e.g., $\sqrt{x^2 + \epsilon}$) or subgradients and explicitly explain your mathematical approach.
   - The projection/repair must satisfy all constraints simultaneously (sum-to-one, non-negativity, and per-asset upper bound). Explain your projection or repair method.
2. **Derivative-Free Method:** Choose either **Simulated Annealing** or **Random Search**.
3. **Population-Based Method:** Choose either **Genetic Algorithm** or **Differential Evolution**.

Important: For each algorithm, clearly describe how constraints are handled (exact projection, repair mechanisms, or penalty functions). Also state whether you implemented the algorithm yourself or used a library, and cite any library/tool you use.

Experimental Protocol

Rolling-Window Setup

- Training window: 252 trading days (~1 year)
- Test window: 21 trading days (~1 month)

Optimization Phase

- Optimization uses only the training window (estimate μ, Σ, and compute w_t).
- Report performance only on the subsequent test window.

Rebalancing

- Rebalance at the beginning of each test window.

Baseline & Robustness

- **Baseline:** Compare your optimized portfolios against an **Equal Weight** portfolio: $w_i = 1/N$.
- **Robustness:** Run each stochastic algorithm with at least 5 random seeds and report results as mean $\pm$standard deviation. Record and report the exact seeds used.

Budget (must be identical across algorithms)

Use either:

- **Time budget:** 60 seconds per rolling window per (λ, γ)setting, or
- **Iteration budget:** a fixed maximum number of iterations per rolling window per (λ, γ)setting.
  State clearly which budget type you used and the exact value (time or iterations).

Metrics and Annualization (state clearly in the paper)

- Use 252 trading days for annualization.
- For the Sharpe ratio, assume a risk-free rate of $r_f = 0$unless you explicitly state and justify another value.
- Report runtime as wall-clock time per rolling step per (λ, γ)scenario, averaged over seeds (for stochastic methods).

Primary Performance Metric

- Use out-of-sample (test-window) Sharpe ratio as the primary metric for comparison. Treat annualized return, volatility, max drawdown, turnover, and runtime as supporting metrics.

What the Paper Must Contain (6–10 pages, PDF)

Ensure your paper includes the following sections:

- **Abstract**
- **Introduction:** problem, motivation, and contributions
- **Related Work:** at least 5 relevant academic citations
- **Method:** mathematical formulation, constraints, objective terms, and turnover handling
- **Algorithms:** hyperparameters, stopping criteria, and constraint-handling logic
- **Experimental Setup:** rolling windows, budget, seeds, baseline, final N, chosen time period, and the file/price field used
- **Results (Tables/Plots):** annualized return, volatility, Sharpe ratio, max drawdown, turnover, and runtime
- **Discussion:** which algorithm performs better under which conditions, and why
- **Limitations:** include a note on dataset limitations or potential survivorship bias
- **Conclusion**
- **AI Assistance Statement:** (see rules below)

What the Code Submission Must Contain

Submit a single ZIP file (or a public repository link) containing:

- **Source Code:** Python or MATLAB
- **Documentation:** README.md with step-by-step instructions to reproduce results
- **Dependencies:** requirements.txt / environment.yml for Python, or toolbox requirements for MATLAB
- **Execution:** a single entry point script (e.g., run_experiment.py or main.m) that reproduces all experiments end-to-end and generates the exact figures/tables reported in the paper. Your code should run end-to-end with one command/script, and should reproduce the reported results when the same seeds are used.

Submission Guidelines

Submit **two items by email** to **seydikacmaz@gantep.edu.tr by the end of May** (**31 May 2026**)

- StudentID_Name_Surname_paper.pdf
- StudentID_Name_Surname_code.zip (or repository link)

Rules & Policies

Similarity (Plagiarism) Rule

- Your paper and code will be checked with a similarity tool (Turnitin/iThenticate or equivalent).
- The overall similarity score must be **≤ 20%**.
- Copying text/code from other students, websites, papers, or AI outputs without proper citation is strictly prohibited.
- If the similarity score is above 20% (or plagiarism is detected), the submission will receive **0 points** and will be handled according to university disciplinary rules.

AI Use Rule

- AI tools (e.g., ChatGPT/Copilot) may be used only for minor assistance such as grammar/spelling checks, rephrasing text you wrote, explaining concepts, or small debugging suggestions.
- AI tools must not be used to write the full paper, produce the full solution, generate full code blocks, or fabricate results.
- You must append the following statement at the end of your paper:

AI Assistance Statement (required):

"AI tools were used for: [specify usage, e.g., grammar checking, debugging].

All modeling, coding, experiments, results, and interpretation were done by the author."

Grading Breakdown (100 Points)

Category	**Points**
Data preparation & baseline definition	15
Model clarity (objective + constraints)	20
Implementation of 3 algorithms (incl. constraints)	30
Experimental design & comparison	20
Paper quality & discussion	15