BSc (Hons) Artificial Intelligence And

Data Science

Level 04

CM 4601

Computational Intelligence

**Optimized Asset Allocation Using Genetic Algorithms and Machine Learning**

**CW Report**

NADUN SHAMIKA SENARATHNE

IIT ID: 20210488

RGU ID: 2117538

Table of Contents

[Task 1: Portfolio Optimization Problem Identification 3](#_Toc185330327)

[1. Problem Identification: 3](#_Toc185330328)

[2. Why Traditional IT or Numerical Optimization Fails: 3](#_Toc185330329)

[3. Why Genetic Algorithms (GA) Are Suitable for This Problem: 3](#_Toc185330330)

[Task 2: Justification for Using Evolutionary Algorithms (GA) 4](#_Toc185330331)

[1. Complexity of Search Space: 4](#_Toc185330332)

[2. Handling Multiple Objectives: 4](#_Toc185330333)

[3. Non-Convex and Non-Smooth Objective Functions: 4](#_Toc185330334)

[4. Stochastic Nature and Risk Management: 4](#_Toc185330335)

[5. Constraint Handling: 4](#_Toc185330336)

[6. Robustness and Multi-Run Evaluation: 5](#_Toc185330337)

[7. Simulation of Future Scenarios: 5](#_Toc185330338)

[Task 3: Holistic View and Modularized Diagram 5](#_Toc185330339)

[1. Input Data Module: 5](#_Toc185330340)

[2. Machine Learning Prediction Module: 6](#_Toc185330341)

[3. Fitness Evaluation Module: 6](#_Toc185330342)

[4. GA Evolution Module: 7](#_Toc185330343)

[5. Constraint Handling Module: 7](#_Toc185330344)

[6. Termination Check Module: 8](#_Toc185330345)

[7. Output Module: 8](#_Toc185330346)

[Modularized Diagram: 9](#_Toc185330347)

[Logical Progression: 9](#_Toc185330348)

[Conclusion: 9](#_Toc185330349)

[Task 4: Define the Methodology 10](#_Toc185330350)

[1. Chromosome Structure 10](#_Toc185330351)

[2. Fitness Function 10](#_Toc185330352)

[3. Constraints 11](#_Toc185330353)

[4. Genetic Algorithm (GA) Operations 11](#_Toc185330354)

[5. Algorithm Flow 11](#_Toc185330355)

# Task 1: Portfolio Optimization Problem Identification

## 1. Problem Identification:

* **Industry:** Finance
* **Application:** Optimize asset allocation in a financial portfolio to maximize returns while minimizing risks.
* **Complexity:**
  + There are numerous asset combinations conceivable in a portfolio, especially with a high number of accessible assets.
  + Each asset's risk and return profile fluctuates over time owing to market circumstances, creating uncertainty.
  + Multiple competing objectives, such as maximising profits and minimising risk (mean-variance optimisation).
  + A Constraints such as budget, industry diversity, and risk tolerance further complicate the situation.

## 2. Why Traditional IT or Numerical Optimization Fails:

* **Exponential Growth of Combinations:** As the number of assets rises, the number of alternative portfolio configurations expands exponentially, making thorough searches impossible.
* **Non-Convex Objective Function:** Portfolio optimisation includes non-linear interactions between assets, creating a non-convex objective function with several local optima. This makes classic gradient-based optimisation approaches challenging to apply.
* **Uncertainty:** Traditional numerical optimisation methodologies are ineffective in capturing market uncertainty due to their deterministic nature and continuous change.

## 3. **Why Genetic Algorithms (GA) Are Suitable for This Problem:**

Evolutionary techniques, such as Genetic techniques (GA), are ideal for this sort of problem because they can effectively traverse a wide, complicated search domain. They are capable of managing several conflicting objectives and providing strong answers in the face of ambiguity. Unlike classic optimisation approaches, GAs may avoid being stuck in local optima and adapt effectively to the changing nature of financial markets.

# Task 2: Justification for Using Evolutionary Algorithms (GA)

## 1. Complexity of Search Space:

* Traditional search methods become ineffective when the quantity of assets increases dramatically.
* The present implementation of the GA utilizes tournament selection, uniform crossover, and random mutations to effectively traverse the huge space.

## 2. Handling Multiple Objectives:

* Optimizing a portfolio requires balancing many objectives, including maximising return, minimising risk, and ensuring diversification.
* The fitness function in the current implementation combines the Sharpe ratio, risk penalties, and incentives for balanced asset allocations to optimise several objectives.

## 3. Non-Convex and Non-Smooth Objective Functions:

* A portfolio's return-risk landscape is non-linear and influenced by complicated inter-asset correlations.
* The GA's stochastic evolution process avoids local optima in the present implementation of covariance matrix-based risk computation, which is not convex.

## 4. Stochastic Nature and Risk Management:

* Deterministic optimisation is ineffective in financial markets due to its uncertainty and volatility.
* In the current implementation:
  + GA's mutation and crossover operators use randomisation to prevent premature convergence.
  + The Random Forest regression model predicts future asset returns by using previous volatility to direct GA development.

## 5. Constraint Handling:

* Portfolio optimisation involves both hard limitations (e.g., budget limits, minimum asset weights) and soft constraints (e.g., risk tolerance, diversification).
* In the current implementation:
  + The current method enforces hard limitations through penalties in the fitness function for weight deviations or high risk.
  + To promote diversification, asset weights with significant variance are penalised while balanced allocations are rewarded.

## 6. Robustness and Multi-Run Evaluation:

* Financial optimization demands resilience in many circumstances.
* In the current implementation:
  + GA is performed numerous times to ensure convergence and consistency.
  + This multi-run technique optimises the GA without relying on a single solution, providing a comprehensive perspective of its performance.

## 7. Simulation of Future Scenarios:

* Optimizing portfolios requires considering future market situations.
* The present implementation uses the Random Forest model to estimate predicted returns based on current data, simulate future scenarios, and guide optimization.

The code improvements, such as machine learning integration for return prediction, multi-run evaluation for robustness, and sophisticated constraint handling in the fitness function, show the versatility and power of Genetic Algorithms in solving complex, multi-objective, and non-linear portfolio optimisation problems. These enhancements make GAs suited for real-world financial optimisation tasks.

# Task 3: Holistic View and Modularized Diagram

**Holistic View:** The **portfolio optimization solution using Genetic Algorithm (GA)** incorporates machine learning for return prediction and a modularized architecture to optimize asset allocation efficiently. Here’s a holistic view of the system:

**Modularized Diagram Flow (Adapted Version):**

1. **Data Preprocessing Stage**:
   * Fetch Data → Clean Data → Calculate Returns → Output Preprocessed Data.
2. **Machine Learning Prediction Module**:
   * Train Random Forest → Predict Asset Returns → Output Predicted Returns.
3. **Fitness Evaluation Module**:
   * Evaluate Fitness:
     + Portfolio Return (using predicted returns)
     + Risk (volatility from covariance matrix)
     + Sharpe Ratio
     + Penalties and Rewards (constraints, diversification).
4. **GA Evolution Module**:
   * Initial Population → Fitness Evaluation → Selection → Crossover → Mutation → Normalization → Offspring.
5. **Constraint Handling Module**:
   * Apply Constraints (budget, risk, minimum allocation).
6. **Termination Check Module**:
   * Check Early Stopping or Max Generations → Return Best Solution.
7. **Output Module**:
   * Present Results:
     + Optimized Asset Weights
     + Portfolio Return and Risk
     + Visualization of Fitness Evolution Across Runs.

# Task 4: Define the Methodology

## **1. Chromosome Structure**

* **Chromosome Representation**: Each chromosome represents a potential solution, such as a financial portfolio.
* **Representation in Code**: Each chromosome is a vector of real integers, with each value representing the capital allocation to an asset. For example, a chromosome representing five assets (AAPL, MSFT, GOOGL, AMZN, and TSLA) may look like [0.20, 0.25, 0.15, 0.30, 0.10].
* **Gene**: Each gene represents the weight of a certain asset in the portfolio.
* **Constraint**: All asset weights must be substantially equal to one to ensure 100% portfolio allocation. The fitness function penalises weights that stray from the 98% to 102% range.

## **2. Fitness Function**

The fitness function evaluates a prospective portfolio's performance by balancing risk, return, and diversification. The aim is to maximize the Sharpe Ratio, which is a measure of return adjusted for risk.

**Components of the Fitness Function**:

1. **Portfolio Return**: The portfolio's return is determined as the weighted sum of expected asset returns. The anticipated returns are calculated using a Random Forest Regressor trained on historical asset returns.
2. **Portfolio Risk**: The covariance matrix of asset returns is used to determine the portfolio's risk or volatility. This matrix tracks how the returns of several assets change together.
3. **Sharpe Ratio**: The Sharpe Ratio serves as the major fitness metric. It calculates the return-risk trade-off by dividing the difference between the portfolio's projected return and a risk-free rate by the volatility of the portfolio. The code assumes a risk-free rate of 2%.
4. **Diversification Reward**: Portfolios with non-zero asset weights receive an additional benefit. This fosters diversity rather than focussing just on a few items.
5. **Balance Penalty**: Portfolios with large volatility in asset weights are penalised to ensure a more balanced allocation. Lower weight standard deviations are rewarded, encouraging evenly dispersed portfolios.

**Constraint Handling in the Fitness Function**:

* A penalty is given if the total of weights is not between 98% and 102%.
* Penalties are enforced for asset weights below 5% to ensure meaningful allocations.
* If the portfolio's risk exceeds 0.35, the solution is penalised.

## **3. Constraints**

The following constraints ensure that the portfolio solutions are realistic and feasible:

1. **Budget Constraint**: Total portfolio allocation should be about 100%.
2. **Minimum Allocation**: Allocate at least 5% of each asset to avoid negligible contributions.
3. **Risk Constraint**: The portfolio risk (volatility) cannot exceed a specified threshold of 0.35.
4. **Diversification**: To decrease concentration risk, portfolios are rewarded for distributing allocations across numerous assets.

## **4. Genetic Algorithm (GA) Operations**

The Genetic Algorithm is used to iteratively evolve portfolios toward optimal solutions.

* **Selection**: The tournament selection technique is used to choose parent portfolios. Higher fitness portfolios have a better probability of being chosen for reproduction.
* **Crossover**: The algorithm employs uniform crossover, combining asset weights from two parent portfolios to produce child portfolios. This strategy provides a diverse population.
* **Mutation**: Random mutation is used to increase variety and prevent premature convergence. The mutation rate in the code is set to 25%, causing asset weights to be adjusted at random within legal ranges.
* **Normalization**: After mutation and crossover, portfolio weights are normalised to add up to 1 while adhering to the budget restriction.

## **5. Algorithm Flow**

1. **Initialize Population**: Generate a random population of portfolios with asset weights between 0 and 1.
2. **Evaluate Fitness**: The fitness function takes into account return, risk, diversity, and restrictions to determine each portfolio's fitness level.
3. **Selection**: Parent portfolios are chosen based on fitness scores through tournament selection.
4. **Crossover and Mutation**: New offspring portfolios are created by mixing the weights of parent portfolios (crossover) and making random modifications.
5. **Normalization**: Asset weights are normalised to add up to 100%.
6. **Constraint Handling**: Portfolios that violate limitations (e.g. risk or minimum allocation) receive poorer fitness ratings.
7. **Output the Best Portfolio**: Output the optimal portfolio, including predicted return, risk, and fitness score (Sharpe Ratio), at the end of the evolution process.

This methodology optimizes financial portfolios by combining machine learning forecasts with Genetic Algorithms. This technique enforces realistic portfolio limits, including total allocation, minimum asset weight, and risk constraints.

* The fitness function incorporates Sharpe Ratio, diversification incentives, and penalties to ensure robust evaluation.
* Genetic Algorithms use selection, uniform crossover, mutation, and normalisation to effectively explore solution space while balancing return and risk.

The end result is a strong, diverse, and optimized portfolio that satisfies predetermined goals and restrictions while reacting to market fluctuations.