

PRACTICAL: 1

Aim: Part A: Implement Hill Climbing Algorithm for function:

- Maximizing: $f(x) = -(x - 3)^2 + 9$
- This is a parabola with maximum at $x=3$
- Initial solution: $x_0=0$
- Neighbors: for simplicity, define them as $x-1$ and $x+1$

Part B: Hill climbing search of a one-dimensional objective function

Part C: Make a Case Study on Portfolio Management using Hill Climbing.

Case Study: Portfolio Management using Hill Climbing

1. Introduction

This case study demonstrates the use of the Hill Climbing optimization algorithm for portfolio management. The objective is to maximize a risk-adjusted performance measure that balances expected returns against risk.

2. Problem Definition

We consider a portfolio of five asset classes: two equities, one bond, one real estate investment trust (REIT), and one commodity. The objective function is defined as:

$$f(w) = w^T \mu - \lambda (w^T \Sigma w)$$

where μ is the expected returns vector, Σ is the covariance matrix of returns, and λ is the risk aversion parameter (here set to 3.0).

3. Methodology

The Hill Climbing algorithm starts with a random feasible portfolio (weights sum to 1, non-negative) and iteratively perturbs weights, projecting them back onto the simplex. Moves are accepted if they improve or maintain the objective, and the global best is tracked.

Steps:

1. Initialize a random feasible portfolio.
2. Compute its risk-adjusted objective.
3. Iteratively propose small random weight adjustments.
4. Re-project onto the feasible simplex.
5. Accept the new portfolio if the score improves or is equal.
6. Track the best solution found.

4. Results

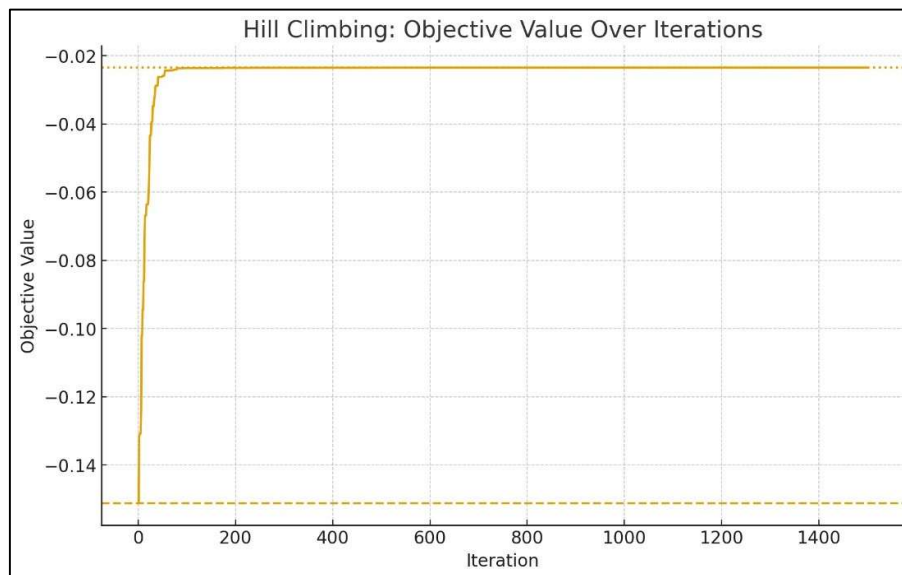
The optimization results are summarized below:

Portfolio	Objective	Expected Return	Risk (Std. Dev.)
Initial	- 0.1512574802977376	0.0831706742746892	0.2795401906538824
Best (Final)	- 0.0234908426940439	0.0577243547811294	0.1645348974890265

Portfolio Weights (Initial vs Final):

Asset	Initial Weight	Final Weight
Equity_A	0.1332	0.0
Equity_B	0.3381	0.0572
Bonds_C	0.2603	0.6151
REIT_D	0.2129	0.0793
Commodity_E	0.0555	0.2483

The following chart shows the improvement in objective value over iterations:



5. Discussion

The algorithm significantly improved the portfolio's objective value compared to the initial allocation. The Hill Climbing approach demonstrates how greedy local search can find better portfolios without requiring complex optimization solvers. However, it may get stuck in local optima, and performance depends on the initial portfolio and perturbation scale.

6. Limitations

1. Hill Climbing can converge to local optima.
2. Results are sensitive to initial portfolios.
3. Step size tuning affects convergence.
4. No guarantee of finding the global optimum.

7. Conclusion

Hill Climbing is a simple yet effective optimization method for portfolio management, capable of improving a portfolio's risk-return tradeoff. Future work could include multi-start strategies, simulated annealing, or genetic algorithms for global exploration.

Code:

PART: A

```
def F(x):  
    return - (x-3)**2 + 9  
  
def neigh(x):  
    return [x-1, x+1]  
  
def hill_Climbing(F, x0):  
    x = x0  
    while True:  
        neighbour = neigh(x)  
        best_neigh = max(neighbour, key=F)  
        if F(best_neigh) <= F(x):  
            return x  
        x = best_neigh  
  
Ans = hill_Climbing(F, 0)  
print("Optimal Solution", Ans)
```

PART: B

```
from numpy import arange
from matplotlib import pyplot

def objective(x):
    return x**2.0

r_min, r_max = -5.0, 5.0
inputs = arange(r_min, r_max, 0.1)
results = [objective(x) for x in inputs]

pyplot.plot(inputs, results)
x_optima = 0.0
pyplot.axvline(x=x_optima, ls='--', color='red')
pyplot.show()
```

Output:**PART: A**

Optimal Solution: 3

PART: B