# Portfolio Optimization using Differential Evolution

# 

-SUBMITTED By

RAUNAK KUMAR BINANI

Dr. MILLE PANT(Supervising faculty)

# Project Description:- The main motive behind the project is to obtain a portfolio optimization model that takes a real-life investment problem and gives a solution that results in the largest return possible with the lowest risk encountered by optimization of the capital proportion of the assets held

# Glossary

1. Portfolio:- A grouping of financial assets such as stocks, bonds and cash equivalents, as well as their mutual, exchange-traded and closed-fund counterparts. Portfolios are held directly by investors and/or managed by financial professionals.
2. Differential Evolution:- In evolutionary computation, **differential evolution** (DE) is a method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

3. Asset Return:-[(closing price for the current period)−(closing price for the previous period)+ (dividend(s) for the current period)]/(closing price for the previous period

4.Objective Function and Constraints:- The main problem that is needed to be maximized or minimized is the objective function. The various restrictions on the parameters that affect the objective function are the constraints.

## Mathematical Equivalent

The mathematical problem of portfolio optimization can be formulated in many ways but the principal problems can be summarized as follows:-

* Minimize risk for a specified expected return.
* Maximize the expected return for a specified risk.
* Minimize the risk and maximize the expected return using a specified risk aversion factor.
* Minimize the risk regardless of the expected return.
* Maximize the expected return regardless of the risk.

## Different Steps Involved in Solving in Differiantial evolution

* ***Initialization*** :-
* Allocate random values to the different affecting parameters.
* Create around 100-200 such initial solutions. These are called ***Target vectors(Xi).***
* Put them in the objective function and find the value.
* ***Mutation*** :-
* Chose a ***Scaling factor***  between 0-1.
* Create a new vector called ***Mutant vector*** using the following :-

**V(i) = X(r1) + F( X(r2) – X(r3))**

* **Crossover** :-
* Create ***Trial Vectors*** under the following condition :-

if( r > ***Cr***)

u[i][j]= x[i][j]

if(r < ***Cr*** )

u[i][j]= v[i][j]

* Find the value of objective function for these trial vectors.

***Cr***  is the ***Crossover Probability*** and r a random number between 0-1.

* ***Selection*** :-
* From the Target and Trial vectors chose the one which gives the better results.

## Methods of Solving

* **Mean-Variance Model**
* **Mean-Semi variance Model**
* **Mean-Absolute Deviation Model**

## Mean-Variance Model

* Min .((wt) X w))
* Σri xi = ro
* Σ xi = 1
* xi > 0 (i= 1,2,3….)

## Mean-Semi variance Model

* min. ΣΣ [(rit – ri)xi]Λ 2 ]/ co
* Σ [(rit – ri)xi]Λ 2 ]=0 (rit-ri)xi > 0
* Σri xi = ro
* Σ xi = 1
* xi > 0 (i= 1,2,3….)

## Mean-Absolute Deviation Model

* min. ΣΣ [(rit – ri)xi)/ co
* Σri xi = ro
* Σ xi = 1
* xi > 0 (i= 1,2,3….)

Code to do Portfolio Optimazation Problem:-

float a[10][20];

**for(i=0;i<co;i++)**

{

**for(j=0;j<n;j++)**

{

cin>>A[i][j];

avg[i]=avg[i] + A[i][j];

}

avg[i]= avg[i]/n;

}

For (i=0;i<co; i++)

{

**for (j=0;j<n;j++)**

A[i][j] = A[i][j] – avg[i];

}

void transpose( float w[10][20]; int n; float wt[1][10]; int j)

{

For (i=0; i<n; i++)

{

W[j][i]= wt[1][i];

}

}

Void transpose2 (float a[10][20]; float at[20][10];int co; int x; float x[20][10])

{

For (i=0;i<co;i++)

{

For(j=0;j<n;j++)

{

For(c=0;c<n;c++)

Void de(int a[100]; int co; int m; int count)

{

int sum=0; float at[10][20]; s[100]

int f = 0.5; int cp = 0.6;

if(count==0)

{

for(j=0; j<m; j++)

{

s[j]=0;

for(i=0;i<co;i++)

X[i][j]= X[i][j] + at[i][c]\* a[c][j];

}

}

Void de(int a[100]; int co; int m; int count,int r0)

{

int sum=0; float at[10][20]; s[100]

transpose(w,co,wt,j);

int f = 0.5; int cp = 0.6;

**for(j=0; j<m; j++)**

{

s[j]=0;

**for(i=0;i<co;i++)**

{

w[i][j]= math.random();

sum= sum + w[i][j];

}

}

**for(i=0;i<co;i++)**

{

wt[1][10];

w[j][i]=w[j][i]/sum;

transpose(w,co,wt,j);

float sa[1][10];

mm (x,wt,1,co,sa);

**for(int c=0;c<co;c++)**

{

s[j]=s[j]+ sa[1][c]\*w[c][1];

s[j]= s[j]/co;

}

}

}

float wm[10][20];

**for(int i=0; i<m;i++)**

{

**for(int j=0;j<co;j++)**

{

int x= math.random();

int y= math.ramdom();

int z= math.random();

wm[i][j]= w[x][j]+ f(w[y][j]-w[z][j]);

sum = sum + wm[i][j];

}

**for(int j=0;j<co;j++)**

wm[i][j]=wm[i][j]/sum;

}

**for(i=0;i<m;i++)**

{

**for(j=0;j<co;j++)**

{

float wt[1][10],sa[1][10],sc[100];

int x= math.random();

**if(x>cp)**

wc[i][j]=m[i][j];

**else**

wc[i][j]=w[i][j];

}

transpose(wc,co,wct,i);

mm(x,wct,1,co,sa);

**for(int c=0;c<co;c++)**

{

sc[i]sc[i]+ sa[1][c]\*wc[c][1];

s[i]=s[i]/n;

}

int sum1=0;

**if(s[i]>sc[i])**

{

**for(j=0;j<co;j++)**

{

w[i][j]=wc[i][j];

sum1= avg[j]\*w[j]+ sum1;

}

}

**else**

{

**for(j=0;j<co;j++)**

sum1= avg[j]\*w[j]+ sum1;

}

**if(sum1-r0==0.001|| r0-sum1==0.001)**

count==5000;

}

c++;

**if(count<5000)**

de(w,co,m,count,r0);

}

## Result:-

|  |  |
| --- | --- |
| **COMPANY** | **ALLOCATION** |
| **Alfa Laval Ltd.** | **0.00312** |
| **Siemens Ltd. (SIL)** | **0.01320** |
| **Bajaj Hindustan Ltd.(BHL)** | **0.13471** |
| **Hindustan Construction Co. Ltd.** | **0.02213** |
| **Star Paper Mill Ltd.** | **0.41578** |
| **ITC Ltd.** | **0.0** |
| **Hero Honda Motors Ltd.** | **0.00590** |
| **Trident Ltd.** | **0.22104** |
| **Crompton Greaves Ltd**. | 0.0 |
| **JK PAPERS Ltd.** | **0.18422** |

**Portfolio Return:- 0.25**

**Portfolio Risk:- 0.02681**