TERM PAPER OF NTCC

ON

CARDINALITY CONSTRAINED PORTFOLIO OPTIMIZATION PROBLEM

AMITY UNIVERSITY UTTAR PRADESH

Guided by:

Mr. Dev Kumar Chaudhary

Assistant Professor

Submitted by:

Achal Sharma

Axxxxx15021

B.Tech CSE(E)

2015-19

DECLARATION BY THE STUDENT

I, **ACHAL SHARMA (Axxxxx15021),** student of **B.Tech. CSE (E)** hereby declare that the project titled "**CARDINALITY CONSTRAINED PORTFOLIO OPTIMIZATION PROBLEM**" which is submitted by me to Department of Amity School Of Engineering & Technology, Amity University, Noida, UP is in partial contentment of responsibility for the honour of the degree of B.Tech. in CSE, which may or may not have been previously performed to acquire any award such as a degree, or other similar title of recognition.

The Author has attested the permission for the use of any registered and copyrighted material in the following Project Report other than brief notes requiring only well acknowledgement in academic writing and such use is recognized.

(ACHAL SHARMA)

Name and Signature of Student

CERTIFICATE BY THE FACULTY GUIDE

This is to certify that **Mr. ACHAL SHARMA**, student of **B.Tech. CSE (Evening)** has carried out the work presented in the project of the Term paper entitled " **CARDINALITY CONSTRAINED PORTFOLIO OPTIMIZATION PROBLEM** " as a part of First Year Programme of Bachelor of Technology in CSE from Amity School of Engineering and Technology, Amity University, Noida, Uttar Pradesh under my supervision.

Name & signature of the Faculty Guide
Amity Institute of Engineering & Technology, AUUP

CONTENTS

ABSTRACT

This term paper implements the use of MOEA (Multiobjective Evolutionary Algorithm) to find the solution of CCPOP (Cardinality Constrained Portfolio Optimization Problem) by studying the Pseudo Code and using it to create a C Program. According to multiple sources, this MOEA is used in implementing an efficient encoding technique designed specifically to deal with CCPOP. There are many researches conducted that applied MOEAs to solve the constrained portfolio optimization problem. It may be used to predict the variation of the price of stocks of different companies.

INTRODUCTION

The portfolio optimization problem targets to find the ideal distribution of a restricted capital among a finite set of risky assets by concurrently maximizing the estimated return of the portfolio and minimalizing its related risk. The portfolio optimization problem can be resolved by quadratic programming. Nonetheless, if we apply some real-world nonlinear (deviating) constraints like Cardinality Constraint that apply upper and lower bound to number of assets stored in the portfolio, then the problem becomes NP-hard. The added deviating constraints made the problem more difficult to be solved with exact procedure. We present a different mutation operator specially designed to work well with the proposed representation scheme. The proposed genetic operators have been intended to satisfy the need of improving the already found good solutions and simultaneously to be able to explore other capable areas of the search landscape. This much asked balance between exploration & exploitation is achieved through the introduction of innovative mechanisms.

OBJECTIVE

The objective of this term paper is to create a C Program through the Pseudo Code of MOEA to find the solution of CCPOP.

DISCUSSION

Typically, the mutation operator takes responsibility for the preservation of population's diversity. In this paper, we use a two-phase mutation operator (TPMO) specially designed to work well with the proposed representation scheme for solving the cardinality constrained portfolio optimization problem (CCPOP).

We employ a variation of the polynomial mutation operator (PLM) for the real-valued solution vector specially adapted to fit the needs of the proposed representation scheme and binary mutation for the discrete (i.e. binary-valued) solution vector. We start the presentation of the proposed TPMO by analysing the mutation process of the real-valued solution vector. The real-valued solution vector contains information about the selected assets and the corresponding budget capitalised in every single one of these assets.

Also, the size of both the real and binary solution vectors is restricted by the upper limit of the cardinality constraint (Kmax). Given the fact that the available pool of stocks for all examined test instances is considerably larger than Kmax, assuming a Kmax value equal to 10, we should devise an updating mechanism that allows the replacement of some assets of the real-valued solution vector with other assets from the available pool of securities.

At the same time, the mutation operator should be able to probe limited areas of the search landscape with the hope of improving a promising solution that has already been found. Below, we analyse step-by-step how we incorporate the aforementioned principals to the proposed mutation operator.

```
Begin
ηm = distribution index;
  for i=0 to P; // where P is the population size
        for z=0 to K_max;  // where K_max is the upper limit of the cardinality constraint
          rand  →   [0, 1];
            if (rand <= mutation_probability) then

                r_p  = getValue(z);
                r_p  = w_p + a_p //where w_p is the assigned weight and α_p the corresponding asset of the parent solution
                w^L  = getLowerBound(z);
                w^U  = getUpperBound(z);

              rand_mut → [0, 1];

              if (0.0 ≤ rand_mut ≤ 0.2)   // Phase 1

                δ1 = (w_p − w^L)/(w^U − w^L)       δ2 = (w^U − w_p)/(w^U − w^L)

                 r → [0, 1];

                if (r<= 0.5) then
                     δq =  [2r + (1 − 2r)(1 − δ1)^(ηm+1)]^(1/(ηm+1))
                  else
                     δq =  1 − [2(1 − r) + 2(r − 0.5)(1 − δ2)^(ηm+1)]^(1/(ηm+1))
                 end if

                  w^child = w_p  + δq( w^U − w^L )
                  w^child' = normalized(w^child)  //w^child' satisfies the budget and floor and ceiling constraint
                  r_c   = w^child' + a_p
                  child_solution_vector =  parent_solution_vector.setValue(z ,  r_c  );

               else if (0.2 < rand_mut ≤ 1.0)  // Phase 2

                    rand_asset→ [1, N]; //where N is the available pool of assets, e.g. for port5, N = 225

                δ1 = (w_p − w^L)/(w^U − w^L)       δ2 = (w^U − w_p)/(w^U − w^L)

                 r → [0, 1];

                if (r<= 0.5) then
                     δq =  [2r + (1 − 2r)(1 − δ1)^(ηm+1)]^(1/(ηm+1))
                  else
                     δq =  1 − [2(1 − r) + 2(r − 0.5)(1 − δ2)^(ηm+1)]^(1/(ηm+1))
                 end if

                  w^child = w_p  + δq( w^U − w^L)
                  w^child' = normalized(w^child)  //w^child' satisfies the budget and floor and ceiling constraint
                  r_c   = w^child' + rand_asset
                  child_solution_vector =  parent_solution_vector.setValue(z ,  r_c  );
               endif
            endif
        endfor
  endfor
```

In particular, as seen in the pseudo code if rand ≤ P(m) then a decision variable is selected to be mutated. Suppose a hypothetical real-valued solution vector r = (r(1), r(2), . . . , r(Kmax)). Each real-valued decision variable r(i) , i = 1, 2, . . . , Kmax contains information about the selected asset a(j), j = 1, 2, . . . , N, where N is the available pool of assets and the corresponding budget w(i), i = 1, 2, . . . , Kmax invested in this particular asset. The budget w(i) invested in each asset, can take values in the interval: w(L)(i) ≤ w ≤ w(U)(i), i = 1, 2, . . . , Kmax where w(L)(i) and w(U)(i) stand respectively for the floor and ceiling constraints.

REFERENCES

1. K. Liagkouras, K. Metaxiotis. "A new efficiently encoded multiobjective algorithm for the solution of the cardinality constrained portfolio optimization problem", Annals of Operations Research, 2016.

2. Mishra, S. K., Panda, G.,& Majhi, R. (2014).Acomparative performance assessment of a set ofmultiobjective algorithms for constrained portfolio assets selection. Swarm and Evolutionary Computation, 16(2014)