International Journal of Computer Applications (0975 – 8887)
Volume 58– No.13, November 2012

A New Multi-Objective Genetic Algorithm for Use in Investment Management

Simona Dinu
Sr. Lecturer
Constanta Maritime University
Mircea cel Batran Constanta
ROMANIA

Gabriela Andrei
Assistant
Constanta Maritime University
Mircea cel Batran Constanta
ROMANIA

ABSTRACT
The portfolio optimization problem is an important management issue in financial economics. Its aim is to calculate an optimal asset allocation that satisfy specific investment goals, out of a given investment plan.
In the past few years, more and more attention is given in applying Evolutionary Computation in solving complex optimization problems. The use of Multi-Objective Evolutionary Algorithms - MOEA in practical problems involving multi-objective optimizations is not restricted to a strict application of an existing algorithm described in literature. Oftenly, for a certain problem, one preferres an algorithm' design that includes strategies characterizing different important algorithms used in the MOEA field.
The main objective of this study was to develop an efficient and effective portfolio selection Multi-Objective Genetic Algorithm.
Experimental tests presented for five benchmark data sets are given to demonstrate significant advantages regarding the solution quality and the speed of the algorithm.

General Terms
Evolutionary Computation, Multi-Objective Evolutionary Algorithms, Optimization, Investment management

Keywords
Genetic Algorithms, Portfolio optimization, Efficient frontier, Mean-Variance

1. INTRODUCTION
The portfolio selection problem, initially proposed by Markowitz in 1952 applies mathematical programming methods to find the optimal investment portfolio, which can maximize the portfolio return and minimize the portfolio risk at the same time. "Markowitz' mean-variance model of stock portfolio selection and optimization is one of the best-known models in finance and was the bedrock of modern portfolio theory" [1].
In this model, it is assumed that asset returns follow a normal distribution. This involves that the return on a portfolio of assets can be perfectly characterized by the expected return and the associated risk measured by variance.
The model relies on several initial hypotheses regarding investors' behavior on the market:
- all investors have the same amount of time to make investment decisions;
- every stock asset return follows a normal distribution.
- investors permanently try to maximize their profit;
- investors face a decreasing marginal utility of their fortune;
- the expected return variability is used as a measure of the risk;
- investors always prefer bigger returns at a certain level of the risk and lower-risk investments at a given level of the expected return;
- all the portfolio securities are risky and characterized by a certain rate of return, dispersion and co-variation with other portfolio securities;
- the expected portfolio rate of return is an exogenous variable of the model;

An assets portfolio is efficient as long as no other portfolio with the same rate of return as the initial one but with a lower risk can be created.

2. PORTFOLIO OPTIMIZATION PROBLEM
An n stocks portfolio rate of return is given by the weighted average of the average returns of the constitutive stocks. This average is between the limits of the best and the worst return of the portfolio stocks in terms of the weights of the constitutive stocks (w_i).

$$E(R_p) = \sum_{i=1}^{n} w_i \cdot E(R_i) \tag{1}$$

where :
n – is the number of stocks in the portfolio
w_i , i=1,…,n – is the weight in the portfolio of stock i – which is the decision variable of the model
The portfolio risk depends on three factors:
- the risk of every stock included in the portfolio
- the covariance between the rates of return for assets in the portfolio
- weightings of its constituent stocks

Thus, the risk of a portfolio of n stocks is given by:

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i \cdot w_j \cdot \sigma_{ij} \tag{2}$$

and

$$\sigma_{ij} = \rho_{ij} \cdot \sigma_i \cdot \sigma_j \tag{3}$$

where:
σij = cov(i,j) – covariance between stock i and stock j
σi , σj – standard deviations of stock i and stock j
ρij – correlation coefficient between stock i and stock j
These quantities must be calculated statistically.
The covariance matrix derives from financial time series which contains stock returns tracked over a certain period of time.
Thus:

$$\sigma_{ij} = \frac{1}{k}\sum_{t=1}^{k}(((R_i^t - E(R_i)).((R_j^t - E(R_j))) \qquad (4)$$

where:

$$E(R_i) = \mu_i = \sum_{t=1}^{k} R_i^t / k \qquad (5)$$

and

$R_i^t =$ the return of stock i at time t

k = the dimension of time series: t = 1,…,k

The identification of efficient portfolios is realized by determining the portfolio structure with a given rate of return and the lowest risk. The efficient layout of portfolio must determine an weighted average of the expected returns of stocks E(Ri) equal to the expected portfolio rate of return (Rp*):

$$\sum_{i=1}^{n} w_i \cdot E(R_i) = R_p^* \qquad (6)$$

Thus, the efficient frontier (the curve that joins all the best possible combinations of the n-stocks portfolios) is generated by the solutions of the following optimization problem:

$$\min \sigma_p^2 = \min \sum_{i=1}^{n}\sum_{j=1}^{n} w_i \cdot w_j \cdot \sigma_{ij} \qquad (7)$$

$$\sum_{i=1}^{n} w_i \cdot E(R_i) = R_p^* \qquad (8)$$

$$\sum_{i=1}^{n} w_i = 1,\ 0 \le w_i \le 1,\ i=1,\ldots,n \qquad (9)$$

The portfolios on the efficient frontier represents the set of Pareto-optimal portfolios; each portfolio on this frontier has the maximum expected return for a given amount of risk, or alternatively, the minimum risk for any given level of return.

The quadratic programming approach to this problem requires method of Lagrange multipliers: the partial derivatives of variables are set to 0 and the corresponding linear equations are solved to determine the minimum variance portfolio structure.

In fact, the mathematical problem can be expressed in various ways:

1. Minimize risk for a specified expected return (as already shown)
2. Maximize the expected return for a specified risk
3. Minimize the risk and maximize the expected return:

$$\min \rho_w = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i \cdot w_j \cdot \sigma_{ij} \qquad (10)$$

and

$$\max \mu_w = \sum_{i=1}^{n} w_i \cdot \mu_i \qquad (11)$$

subject to:

$$\sum_{i=1}^{n} w_i = 1 \qquad (12)$$

$$0 \le w_i \le 1,\ i=1,\ldots,n \qquad (13)$$

In this mathematical model:

- parameters are:

μ_i= E(Ri): the expected return of stock i, i=1,…,n

σ_{ij}: the covariance between the returns of stock i and stock j , i,j=1,...,n

- decision variables are:

w_i: weight of stock i, i=1,…,n

This model represents a bi-objective optimization problem. A solution of this problem must simultaneously minimizes portfolio variance (Eq. 10) and maximizes expected return (Eq. 11) while satisfying the set of equalities (Eq. 12) and inequalities (Eq. 13) constraints.

The feasible space F of the problem is defined by (Eq. 12) and (Eq. 13):

$$F=\{w \in R^n \mid \sum_{i=1}^{n} w_i = 1,\ 0 \le w_i \le 1,\ i=1,\ldots,n\} \qquad (14)$$

Generally, the result of the problem consists of efficient portfolios. A portfolio is said to be efficient (or Pareto optimal or non-dominated) if and only if there is no other feasible portfolio that improves at least one of the two objective functions of the model without worsening the other. More specific, if X is the set of all feasible portfolios, we say that a portfolio x' ∈ X dominates another portfolio x'' ∈ X (x'≻x'') if μ_w (x')≥ μ_w (x'') and ρ_w(x')≤ ρ_w(x'') with at least one strict inequality.

Several methods have been proposed in the Operations Research literature for solving multi-objective optimization problems. According to [2], these methods can be grouped into following approaches:

- **A priori preference articulation:** the decision-maker is allowed to specify the preferences that are articulated in terms of goals or the relative importance of each objective function. Most of these methods aggregate the objective functions into a linear or nonlinear scalar cost function.

The most usual approach for evolutionary optimization is the weighted sum of objective functions:

$$F(x)= \sum_{i=1}^{n} \alpha_i \cdot F_i(x),\ \text{where } \sum_{i=1}^{n} \alpha_i = 1 \qquad (15)$$

Thus, the multi-objective problem is converted into a single objective one.

The bi-objective portfolio optimization problem is usually solved with such a method: a trade-off coefficient $\alpha \in [0,1]$ (which is also called a risk aversion parameter) is introduced to combine the two objective functions into a scalar to be minimized:

$$\min\ \lambda \cdot [\sum_{i=1}^{n}\sum_{j=1}^{n} w_i \cdot w_j \cdot \sigma_{ij}] - (1-\lambda) \cdot [\sum_{i=1}^{n} w_i \cdot \mu_i] \qquad (16)$$

subject to: $\sum_{i=1}^{n} w_i = 1$, (17)

$0 \le w_i \le 1$, i=1,…,n (18)

This approach generates non-dominated solutions by varying the λ coefficient: from the minimum variance portfolio (λ=1) to the maximum return portfolio (λ=0).

A known drawback of this method is the multitude of trials that must be performed, corresponding to different values of λ in the interval [0,1]. This requires substantial computing effort. Fig. 1 to Fig. 5 show the efficient frontiers computed taking 2000 different λ values for the five benchmark problems described in section 5 based on the data made available by Beasley in the OR library[3]. These are also called as standard efficient frontiers of global optimal Pareto set.

In literature there are described some other approaches of the a priori preference articulation. For example in [4] genetic algorithm with a combined fitness function based on the objective function of the portfolio variance and on the objective function of the portfolio return is used. The fitness function designed in [5] defines a balance between risk and return by adjustable constants. In [6] the designed genetic algorithm is based on a fitness function that combines the two objectives into a linear scalar cost function. Another approach is presented in [7]: the authors formulate a portfolio optimization problem involving multiple objectives and transform the multi-objective problem into a single-objective problem by weighting the objectives into the fitness function.

- **Progressive preference articulation:** during the optimization process, the decision-maker is confronted to different possible solutions and interacts with the optimization program.

- **A posteriori preference articulation:** the decision-maker is not capable to specify in advance the relative importance (by setting the weighting factors) of the objective functions. Once the set of non-dominated solutions has been found, the decision-maker can select the desired solution. Most Evolutionary Algorithms for multi-objective optimization can be viewed as a posteriori optimization techniques. Because they perform a search for multiple solutions in parallel, they try to discover the whole set of non-dominated solutions, or at least a well-distributed set of representatives [8]. A well-distributed set is obtained when the relative distance between the non-dominated solutions is as equal as possible.

According to the definition presented in [9], a good spread of solutions is obtained by the optimization problem:

$$\text{Maximize } \min_{\substack{i \neq j \\ x_i, x_j \in P, |P|=n}} \| x_i - x_j \| \quad (19)$$

The proposed approach belongs to this class of optimization techniques In our model, the two objective functions will be tracked at the same time, so that an approximation of the Pareto frontier is obtained in a single run, unlike the multiple runs needed in the case of converting the two objectives into a single objective one.

3. STOCK ALLOCATION OPTIMIZATION USING A MULTI-OBJECTIVE GENETIC ALGORITHM

To apply a genetic algorithm for this problem, an appropriate chromosome coding and a correct fitness function design are required. Solution for stock allocation should be a composition of the stock quantity to be held to minimize the risk on a given level of expected return that will represent the optimal solution.

Considering the admissibility condition: $\sum_{i=1}^{n} w_i = 1$, we propose a real coding, the k[th] chromosome of the current generation has the decisional values encoded in the following structure:

array w: w[i], i=1,...,m represents is the weight in the portfolio of stock i

Initial population P(0) is randomly generated: in every chromosome, a value of a gene (a weight) is randomly generated. We have also to rescale the weight to satisfy: $\sum_{i=1}^{n} w_i = 1$

As the result, we convert every weight into a normalized weight:

$$x_i = \frac{w_i}{\sum_{i=1}^{n} w_i}, \text{ i=1,…,n} \quad (20)$$

Let say PS is the population size. Therefore, the generation at time t is a set of PS strings:

P(t) = { $chrom[1]_t$, $chrom[2]_t$, ……, $chrom[PS]_t$ } and the length of each string is n.

In the portfolio optimization, the fitness function must produce a reasonable trade-off between minimizing risk and maximizing return. For the multi-objective optimization we've applied a technique of archiving the non-dominated solutions, combined with a strategy that characterizes the VEGA algorithm [10]: randomly dividing the population of chromosomes into two sub-populations of equal size (2= number of objectives), according to proportional selection, running consecutively for each objective:

procedure *Selection (P(t)) //P(t) is the current population*
begin
let q=PS/2;//PS is the population size
for *i=1,2 //for every objective*
for *j=1+(i-1)·q,...,i·q*
chrom[j].valf ← objective_function$_i$;
//chrom[j] is the j[th] chromosome in the current population P(t)
$f_{\max} \leftarrow \max_{j=1+(i-1)\cdot q}^{i\cdot q}$ {chrom[j].valf};
chrom[j].probability $\leftarrow \dfrac{f_{\max} - chrom[j].valf}{\sum_{k=1+(i-1)\cdot q}^{i\cdot q} (f_{\max} - chrom[k].valf)}$;
repeat
Based on these probabilities, one select q solutions by tournament selection;
// these solutions will form the sub-population $P_i(t)$
repeat
return P(t)=$\bigcup_{i=1}^{2} P_i(t)$ *// P(t) is the new population*
end

Subpopulations were then combined to form a population of size PS on which the crossover and mutation operators will be applied.

In the algorithm efficient solutions against one of the objectives are favored.

To obtain also intermediate solutions, the algorithm allows crossover between any two solutions in the population.

In this way, a crossover between two efficient solutions (each corresponding to one of the objectives) can result in offsprings that represent a good compromise between the two objectives.

To eliminate the disadvantage related to the fact that an individual which is not dominated in a generation can become dominated in a subsequent generation, we used two archives:

- an archive that saves the non-dominated individuals in the current generation;
- a second archive that saves non-dominated individuals identified until the current search.

In the new generation, a percentage of the population P(t +1) will be replaced randomly with solutions of this second external archive.

These procedures are described below:

procedure *Domination (P(t))*
//calculates the non-dominated individuals
begin
  for *i=1,PS*
    //initially all individuals are non-dominated
    chrom[i].domination ← 1;
  repeat
  for *i=1,PS*
    for *j=1,PS*
      if (j≠i)
        and (((chrom[j].objective1≤ chrom[i].objective1)
        and(chrom[j].objective2< chrom[i].objective2))
        or ((chrom[j].objective1<chrom[i].objective1)
        and(chrom[j].objective2≤ chrom[i].objective2))
        or ((chrom[j].objective1<chrom[i].objective1)
        and(chrom[j].objective2< chrom[i].objective2)))
          then chrom[i].domination ← 0;
            // the i^{th} chromosome is dominated
      endif
    repeat
  repeat
end

procedure *Current_generation_archiving (P(t))*
begin
  // „archive" will contain the non-dominated individuals from the current generation
  // initially the current position in this archive is 0
  position_archive_current$_{gen}$ ← 0;
  for *i=1,PS*
    if (chrom[i].domination=1) ***then***
      position_archive_current$_{gen}$ ←position_archive_current$_{gen}$+1;
      archive[position_archive_current$_{gen}$] ← chrom[i];
    endif
  repeat
end

procedure *Current_search_archiving (P(t))*
begin
//" current_gen_archive" will contain the non-dominated individuals identified until the current search
// initially the current position in this archive is 0
position_archive_current$_{search}$ ← 0;
  for *i*=1, position_archive_current$_{gen}$
    for *j*=1, position_archive_current$_{gen}$
      if (j≠i) and (((archive[j].objective1≤archive[i].objective1)
        and (archive [j].objective2<archive [i].objective2))
        or ((archive [j].objective1<archive [i].objective1)
        and (archive[j].objective2≤archive [i].objective2))
        or((archive[j].objective1<archive[i].objective1)
        and (archive [j].objective2<archive [i].objective2)))
        then archive [i].domination ← 0;
      endif
    repeat
  repeat
  for *i*=1, position_archive_current$_{gen}$
    if (archive[i].domination=1) ***then***
      position_archive_current$_{search}$ ←
      position_archive_current$_{search}$ +1;
      current_gen_archive[position_archive_current$_{search}$] ←
      archive[i];
    endif
  repeat
end

procedure *Random_replacing (position_archive_current$_{search}$)*
begin
  for i=1, position_archive_current$_{search}$
    chrom[1+rand()%PS] ← current_gen_archive[i];
  repeat
end

The algorithm displays solutions from external archive „current_gen_archive".

These form the set of non-dominant Pareto solutions (Pareto optimal solutions):

procedure *Pareto_optimality (P(t))*
begin
for *i*=1, position_archive_current$_{search}$
  write current_gen_archive[i].objective1;
  write current_gen_archive[i].objective2; *//best portfolio*
  for k=1,n
    write current_gen_archive[i].w[k];
      // the structure of the best portfolio
  repeat
repeat
end

Crossover:

This study uses the one-cut point crossover method: cutting two strings at a randomly chosen position and swapping the two tails.

This allows introduction of new genetic material and maintaining genetic diversity.

The probability of crossover is p_c , so that an average of p_c x 100% chromosomes undergo crossover.

Crossover is performed in the following way:

1. generate a random integer number r from the range {1, …, n-1}
2. $\left.\begin{array}{l} child_1[i] \leftarrow parent_1[i] \\ child_2[i] \leftarrow parent_2[i] \end{array}\right\}$ for i=1,…,r
3. $\left.\begin{array}{l} child_1[i] \leftarrow parent_2[i] \\ child_2[i] \leftarrow parent_1[i] \end{array}\right\}$ for i=r+1,…,n

Mutation:
This study uses the swap mutation technique: randomly pick two genes and swap their position in chromosome.
The probability of mutation is $p_{mut,}$ so that on average of p_{mut}x 100% of total genes undergo mutation.
Mutation probability is adjusted according to the scheme proposed in [11]:

$$p_m(t)=\left(2+(n-2)\frac{t}{G_{MAX}}\right)^{-1} \qquad (21)$$

$p_m \in [1/n, 1/2]$, where n is the length of the chromosome.
Thus, the algorithm for solving this optimization problem is:

t ← 0;
//*generate initial population*
P(0) ← {chrom[1]$_0$, chrom[2]$_0$, ……, chrom[PS]$_0$};
while (stopping condition not fulfilled)
  //*calculate the non-dominated individuals*
  Domination ((P(t)) ;
  //*calculate the non-dominated individuals from the current generation*
  Current_generation_archiving (P(t)) ;
  //*calculate the non-dominated individuals identified until the current search*
  Current_search_archiving (P(t)) ;
  Selection (P(t)) ; // *VEGA algorithm*
  Crossover ((P(t)) ;
  Adaptive_mutation ((P(t)) ;
  //*some individuals will be randomly replaced with nondominated ones*
  Random_replacing (position_archive_current$_{search}$) ;
  t ← t+1 ;
repeat
//*display the non-dominated solutions that shape the final Pareto frontier*
Pareto_optimality (P(t)) ;

4. PARETO FRONT QUALITY

In the multi-objective optimization problems, the proper evaluation of the performance of an evolutionary algorithm is based on two aspects:
- **Proximity:** the convergence of generated solutions to the Pareto-optimal set;
- **Diversity:** distribution of generated solutions

The heuristic efficient frontier generated by the proposed algorithm is evaluated by using some standard quality indicators of these aspects:
Proximity was measured using the Generational Distance (GD) metric [12]. It estimates how close the generated efficient frontier $PF_{generated}$ is from the known Pareto front, PF_{true}. Mathematically, GD is defined as:

$$GD=\frac{(\sum_{i=1}^{n} d_i)^{1/2}}{n} \qquad (22)$$

where:
- n is the number of solutions generated (the number of vectors in the obtained non-dominated set);
- d_i is the Euclidean phenotypic distance between each member i of the $PF_{generated}$ and the nearest member of the PF_{true}.

A lower GD signifies that $PF_{generated}$ is very close to the PF_{true}. A value of zero means that all the generated solutions are placed on the PF_{true}.
Diversity was measured using the Spacing (SP) metric [13]. It estimates the spread of generated solutions, how uniformly are these solutions distributed in the objective space.
Mathematically, S is defined as:

$$S=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2} \qquad (23)$$

where:
- n is the number of solutions generated (the number of vectors in the obtained non-dominated set);

$$d_i=\min_{j\neq i}(|\rho_w^i(w)-\rho_w^j(w)|+|\mu_w^i(w)-\mu_w^j(w)|)$$

, i,j=1,...,n

$$\bar{d}=\frac{\sum_{i=1}^{n} d_i}{n}$$

A lower S signifies that the solutions in $PF_{generated}$ are uniformly spread out. A value of zero means that all the generated solutions are equidistantly spaced.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to evaluate the efficiency of our algorithm and the superiority of the solutions obtained, the experiments have been conducted with a public available data set, obtained from the OR Library [3].
This data set offers input data for groups of assets in some stock market indices: Hang Seng with 31 assets (P1) - representative for the performance of the Hong Kong stock market, Dax 100 in Germany with 85 assets (P2), FTSE 100 in UK with 89 assets (P3), The US S&P 100 with 98 assets (P4) and Nikkei 225 in Japan with 225 assets (P5). These data correspond to weekly prices between March 1992 and September 1997.
The risk and the corresponding tradeoff return for standard efficient frontiers are available in portref1 to portref5 files [x]. The standard efficient frontiers and the obtained heuristic efficient frontiers are illustrated in Fig. 1 to Fig.5:

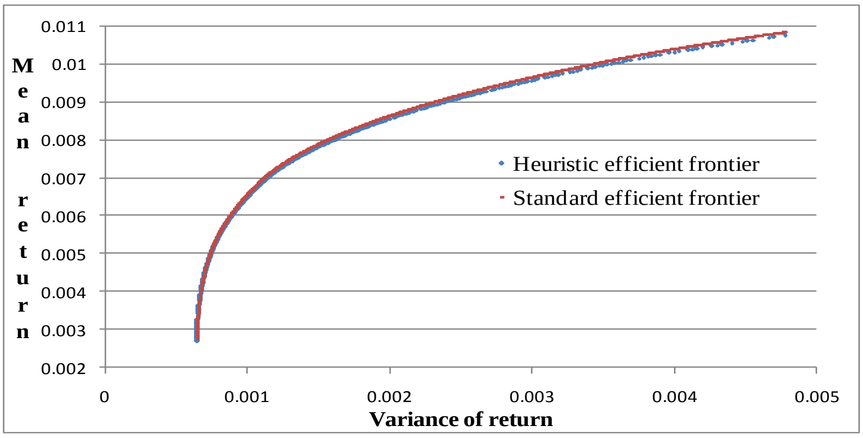

Fig 1: Standard and heuristic efficient frontiers for Hang Seng with 31 assets (P1)

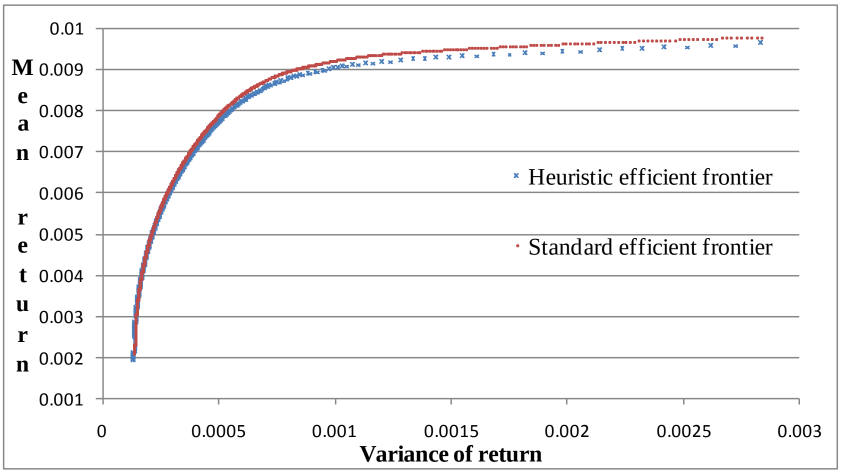

Fig 2: Standard and heuristic efficient frontiers for Dax 100 with 85 assets (P2)

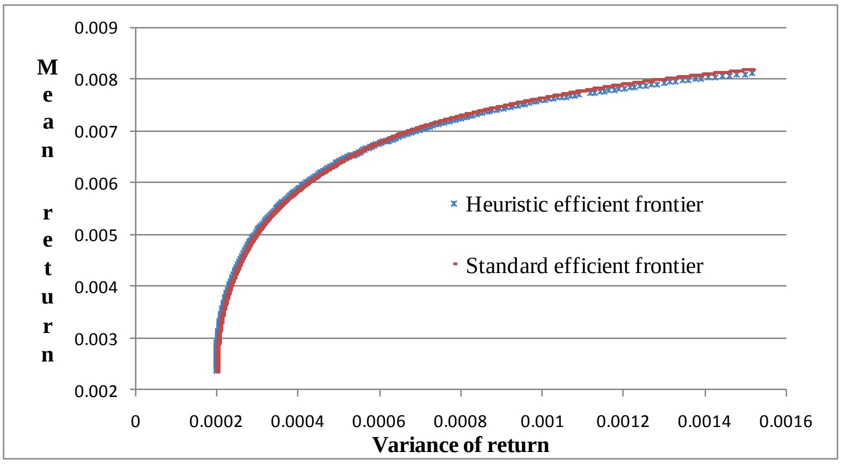

Fig 3: Standard and heuristic efficient frontiers for FTSE 100 with 89 assets (P3)

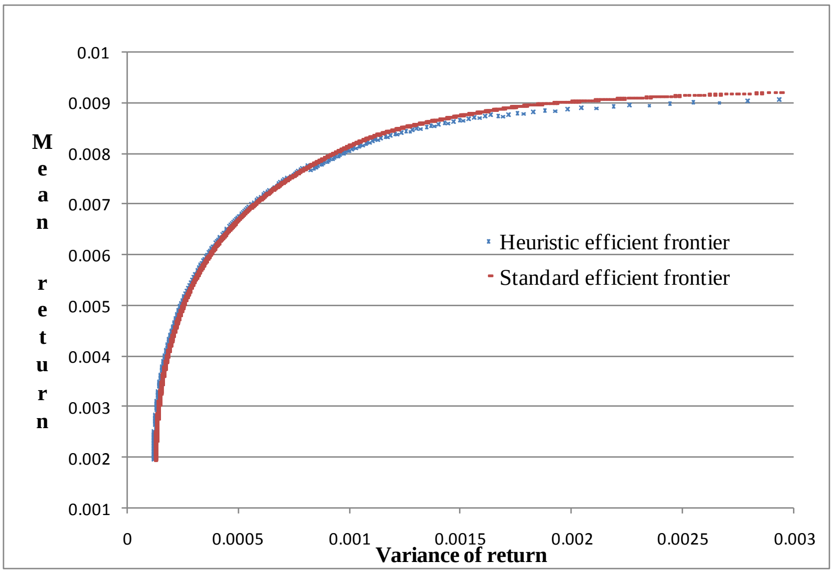

Fig 4: Standard and heuristic efficient frontiers for S&P 100 with 98 assets (P4)

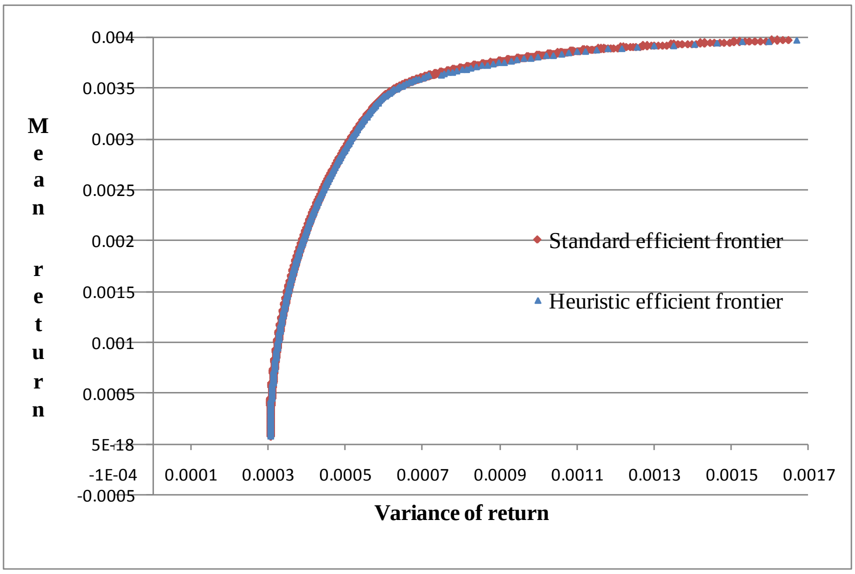

Fig 5: Standard and heuristic efficient frontiers for Nikkei 225 with 225 assets (P5)

The performance of the proposed approach is evaluated against another MOEA, namely Pareto Envelope-based Selection Algorithm - PESA [14].

Inspired by the analysis described in [15], 50 independent runs were performed for each of the two algorithms, corresponding to the five test problems. In order to ensure a fair comparison, the same random seed was assigned to each corresponding set of runs.

Table 1 presents the minimum, the maximum, the mean and the standard error of the two metrics: GD and S obtained for each algorithm.

Table 1. Performances of the proposed algorithm and the PESA algorithm

Metric / Algorithm	GD				S			
	Min	Max	Mean	Std.err	Min	Max	Mean	Std.err
PESA	0.96 E-2	2.71 E-2	1.82 E-2	0.49 E-2	8.02 E-3	9.46 E-3	8.64 E-3	2.71 E-3
Proposed approach	0.68 E-2	2.04 E-2	1.41 E-2	0.26 E-2	4.22 E-3	5.94 E-3	5.29 E-3	1.68 E-3

Table 2 presents the average of the run time results for each of the algorithms, corresponding to the five tests problems.

Table 2. The average of the run time results for each of the algorithms

Index	Assets	PESA	Proposed approach
Hang Seng	31	58 (s)	36 (s)
DAX 100	85	188 (s)	114 (s)
FTSE 100	89	218(s)	145 (s)
S&P 100	98	241 (s)	182 (s)
Nikkei 225	225	968 (s)	719 (s)

As presented in Table 2, the convergence speed of the proposed algorithm is better that of PESA algorithm.

6. CONCLUDING REMARKS

Experimental results show that our algorithm solves the problem efficiently (in terms of computation time and memory space), and produces a relatively well-distributed set of non-dominated solutions that approximate the Pareto-optimal front in a single simulation run.

The algorithm described can be additionally improved by altering genetic algorithm parameters selected for implementation. The population size, crossover rate, mutation rate, selection, crossover and mutation methods could be analyzed in order to find the best parameter design.

Due to the flexibility of the genetic algorithms, further complex constraints of practical interest can be easily integrated to enhance the realism of the model.

Further aspects will concern the integration of some other specific features of MOEAs in order to improve the robustness of the algorithm.

And because genetic algorithms are well appropriated for parallel implementations, one can achieve such implementations as well.

7. REFERENCES

[1] Busetti, F.R. 2000. Metaheuristic approaches to realistic portfolio optimisation. Master thesis. University of South Africa.

[2] Coello, C.A., Lamont, G.B. and Van Veldhuizen, D.A. 2007. Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York.

[3] Portfolio selection benchmark data set at http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files.

[4] Sefiane, S. and Benbouziane, M. 2012. Portfolio Selection Using Genetic Algorithm, Journal of Applied Finance & Banking, Vol.2(4), 143-154.

[5] Shoaf, J. and Foster, J.A. 1998. The efficient set GA for stock portfolios. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (CEC'98). Anchorage, Alaska: IEEE Press, 354–359.

[6] Vedarajan, G., Chan, L.C. and Goldberg, D.E. 1997. Investment Portfolio Optimization using Genetic Algorithms. In Proceedings of the Genetic Programming 1997 Conference, J. R. Koza, Ed. Stanford University, California: Stanford Bookstore, 255–263.

[7] Chiranjeevi, C. and Sastry, V.N. 2007. Multi-objective portfolio optimization models and its solution using genetic algorithms. In Proceedings of the Computational Intelligence and Multimedia Applications, 453–457.

[8] Branke J. and Deb, K. 2005. Integrating User Preferences into Evolutionary Multi-Objective Optimization. In Yaochu Jin (editor), Knowledge Incorporation in Evolutionary Computation, Springer, 461-477.

[9] Kukkonen, S. and Deb, K. 2007. An empirical scalability study of a non-dominated solutions pruning algorithm. In Late Breaking Papers Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization EMO 2007, Matsushima, Japan.

[10] Schaffer, J.D. 1985. Multiple objective optimization with vector evaluated genetic algorithms. In Proc. 1st ICGA, 1985, 93–100.

[11] Back, T. and Schutz, M. 1996. Intelligent mutation rate control in canonical genetic algorithms, Proc. of the Ninth International Symposium on Methodologies for Intelligent Systems, 158-167.

[12] Van Veldhuizen, D.A. and Lamont, G.B. 2000. On measuring multi-objective evolutionary algorithm performance. In Proceedings of the 2000 Congress on Evolutionary Computation, 204–211.

[13] Schott, J. R. 1995. Fault Tolerant Design Using Single and Multi-criteria Genetic Algorithm Optimization, M.S. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.

[14] Corne, D. W., Knowles, J. D. and Oates, M. J. 2000. The pareto envelope-based selection algorithm for multi-objective optimization. In M. S. et al. (Ed.), Parallel Problem Solving from Nature – PPSN VI, Berlin, Springer, 839–848.

[15] Mishra, S.K., Panda, G., Majhi, B. and Majhi, R. 2012. Improved Portfolio Optimization Combining Multi-objective Evolutionary Computing Algorithm and Prediction Strategy. Proceedings of the World Congress on Engineering, July 4 - 6, 2012, London, U.K., Vol. I, 470-474.