Multi-objective particle swarm optimization approach to portfolio optimization

Sudhansu Kumar Mishra[1], Ganapati Panda[2], Sukadev Meher[3],
Department of Electronics & Communication, NIT Rourkela, India-769008,
sudhansu.nit@gmail.com. ganapati.panda@gmail.com,smeher@nitrkl.ac.in

Abstract*— The problem of portfolio optimization is a standard problem in financial world and has received a lot of attention. Selecting an optimal weighting of assets is a critical issue for which the decision maker takes several aspects into consideration. In this paper we consider a multi-objective problem in which the percentage of each available asset is selected such a way that the total profit of the portfolio is maximized while total risk to be minimized, simultaneously. Four well-known multi-objective evolutionary algorithms i.e. Parallel Single Front Genetic Algorithm (PSFGA), Strength Pareto Evolutionary Algorithm 2(SPEA2), Nondominated Sorting Genetic Algorithm II(NSGA II) and Multi Objective Particle Swarm Optimization (MOPSO) for solving the bi-objective portfolio optimization problem has been applied. Performance comparison carried out in this paper by performing different numerical experiments. These experiments are performed using real-world data. The results show that MOPSO outperforms other two for the considered test cases.

***Index Terms*—Multiobjective optimization, Pareto optimal solutions, global optimization, crowding distance, portfolio optimization**

I. INTRODUCTION

Selecting an optimal portfolio weighting of available assets is main aim of portfolio optimization problem. Portfolio optimization is very complicated as it depends on many factors such as assets interrelationships, preferences of the decision makers, resource allocation and several other factors. As a result, the decision maker has to take several issues into consideration. These issues are conflicting which makes the problem as a multi-objective one. In fact all practical optimization problems, especially economical design optimization problems have a multi-objective nature much more frequently than a single objective one. In this paper work we suggest the use of multi-objective optimization algorithms for optimal weighting of assets as a portfolio optimization problem. Here we use PSFGA, SPEA2 and NSGA II and MOPSO for modeling the Pareto front and for optimizing the portfolio performance. The results obtained with these four algorithms are finally compared by performing different numerical experiments.

II. STATEMENT OF THE PROBLEM

A portfolio p consists of N assets. Selection of optimal weighting of assets (with specific volumes for each asset given by weights (w_i) is to be found.

$$\rho_p{}^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \tag{1}$$

$$\alpha_p = \sum_{i}^{N} w_i \mu_i \tag{2}$$

$$\sum_{i}^{N} w_i = 1 \tag{3}$$

$$0 \le w_i \le 1; \text{ and } i = 1,2...,N \tag{4}$$

Where N is the number of assets available, μ_i the expected return of asset i, σ_{ij} the covariance between asset i and j, and finally w_i are the decision variables giving the composition of the portfolio. ρ_p be the standard deviation of portfolio and α_p be the expected return of portfolio. This is a multi-objective optimization problem with two competing objectives. First is to minimize the variance (risk) of the portfolio and at the same the return of the portfolio will be maximized. Equation 3 and 4 gives the constraints for this portfolio optimization problem.

III. EVOLUTIONARY MULTIOBJECTIVE ALGORITHMS

Multi-objective evolutionary algorithms are popular approaches in dealing with problems which consider several objectives to optimize. In this paper we compared the performance of four recently developed multi-objective evolutionary algorithms such as PSFGA ,SPEA2, NSGA2 and MOPSO for optimal weighting of assets in portfolio optimization problem.

978-1-4244-5612-3/09/$26.00 ©2009 IEEE

A. PSFGA algorithm

Parameters: N (population size) , T (maximum number of generations) , *Pm* (mutation rate), *Pc* (crossover probability).

Output: A (non-dominated set)

Step 1: Initialization: $P_t = P_0$ (initial population)

Step 2: Fitness assignment: Determine the objective vector $f(x)$ for each individual in P_t. Determine A ,(the non-dominated set in P_t).

Step 3: Selection: If $S(A) = N$ apply a filter function to A producing $A^{'}$ (a well diversified set of nondominated individuals in P_t). Se t $P^{'} = A^{'}$

Step 4: Recombination:Chose two individuals i, j in $P^{'}$ (with replacement). Recombine i and j with probability *Pc* (by using a one-point crossover funct -ion) producing individual k. Mutate k with mutat -ion rate *Pm* producing $k^{'}$. Set $P^{'} = P^{'} \cup \{k^{'}\}$.

Repeat Step 4 until size of $P^{'}$ is e qua l to N.

Step 5: Termination: Set $t = t + 1$ and $P_{t+1} = P^{'}$. If $t > T$ then A is the non-dominated set of P_{t+1} else go to Step 2.

B. The SPEA2 Algorithm

Input: N (population size), $\bar{N}$ (archive size)
T (maximum number of generations)
Output: A (nondominated set)

Step1:Initialization: Generate an initial population P_0 and create the empty archive(external set) $\bar{P}_0 = \phi$ and set $t = 0$.

Step2: Fitness assignment: Calculate fitness values of individuals in p_t and $\bar{p}_t$.

Step3:Environmental selection: Copy all nondominated individuals in P_t and $\bar{p}_t$ to $\bar{P}_{t+1}$. If size of $\bar{P}_{t+1}$ exceeds $\bar{N}$ then reduce $\bar{P}_{t+1}$ by means of the truncation operator, otherwise if size of $\bar{P}_{t+1}$ is less than N then fill $\bar{P}_{t+1}$ with dominated individuals in P_t and $\bar{p}_t$.

Step 4: Termination: If $t \geq T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the nondominated individuals in $\bar{P}_{t+1}$. Stop.

Step 5: Mating selection: Perform binary tournament selection with replacement on $\bar{P}_{t+1}$ in order to fill the mating pool.

C . NSGA II Algorithm

1. Initialize Population
2.Generate random Parent Population p_0 of size N
3. Evaluate Objective Values
4. Assign Fitness (or Rank) equal to its nondominated level
5. Generate Offspring Population Q_0 of size N with Binary Tournament Selection, Recombination and Mutation
6. For $t = 1$ to Number of Generations
 - Combine Parent and Offspring Populations
 - Assign Rank (level) based on Pareto Dominance.
 - Generate sets of non-dominated fronts
 - until the parent population is filled do
 -- determine Crowding distance between points on each front F_i
 -- include the ith nondominated front in the next parent population (P_{t+1})
 -- check the next front for inclusion
 - Sort the front in descending order using Crowded comparison operator
 - Choose the first N - card (P_{t+1}) elements from front and include them in the next parent population (P_{t+1})
 - Using Binary Tournament Selection, Recombination and Mutation Create next generation
7. Return to 6

V. MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

Observing bird flocks and fish schools, Kennedy and Eberhart realized that an optimization problem can be formulated by mimicking these social behavior. This observation and inspiration by the social behavior resulted the invention of a novel optimization technique called particle swarm optimization (PSO). Two different techniques have been employed in the literature for gbest selection: (1) roulette wheel selection regime (2) quantitative standards [4][5]. Hu and Eberhart proposed a local lbest and a single pbest for each swarm member[6]. Coello and Lechunga applied a grid based selection scheme[7]. Liao, Tseng, and Luarn proposed a discrete version of PSO for flow shop scheduling problems[8].Niu, Zhu, He, and Wu proposed a cooperative PSO where the population comprises of a master swarm and several slave swarms[9].

MOPSO algorithm

Step 1: P_1, A_1 = Initialization

Step 2: FOR t = 1 to N

A. $P_{t+1} = Generate(P_t, A_t)$

for j =1 TO POPULATIONSIZE

$g_{j,t}$ = findgbest $(A_t, P_{j,t})$

$P_{j,t+1}$ = Update Particle $(P_{j,t}, g_{j,t})$

Evaluate $(P_{j,t+1})$

$p_{j,t}$ = UpdatepBest $(P_{j,t+1})$

NEXT

B. A_{t+1} = UpdateArchive (P_{t+1}, A_t)

C. P_{t+1} = Mutation (P_{t+1})

NEXT

Step 3: OutputArchive(A_{t+1})

where, t denotes the generation index, P_t is the population, A_t is the Archive set at t-th generation, $g_{j,t}$ is the gBest of j-th particle, $p_{j,t}$ is the pBest of j-th particle, and $P_{j,t}$ is the j-th particle of P_t at t-th generation.

VI. COMPARISON OF RESULTS

For comparison of two non-dominated solution sets obtained by the proposed multiobjective algorithm for efficient weighting of available assets, the following measures are computed:

S, Δ and C metric:

If the S metric of a nondominated front f_1 is less than another front f_2 then f_1 better then f_2. It has been proposed by Zitzler [4]. The metric called as spacing metric (Δ) measures how evenly the points in the approximation set are distributed in the objective space.It has been proposed by K Deb [1]. The lowest values for Δ indicate a better diversity. Two sets of nondominated solutions can be compared Using C metric and it was introduced by Zitzler[4].

VII. EXPERIMENTAL RESULTS

We run experiments on data from OR library that maintained by Prof. Beasley as a public benchmark data set and is derived from Heng Seng data set with 31 assets. The data can be found at http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html. The PSFGA, SPEA2 and NSGA-II is employed using a population size of 100, number of generation 100, crossover rate is taken as 0.8 and it mutation rate 0.05. The MOPSO used a population of 100 particles and a repository size of 100 particles.

TABLE I THE S AND Δ METRICS

Algorithm	PSFGA	SPEA 2	NSGA II	MOPSO
Metric S	0.000423124	0.000304616	0.0000067874	0.0000005943
Metric Δ	0.8898852851	0.8337976192	0.5967844252	0.561273589

Table I shows the S metric and Δ metric obtained using all four algorithms. Table I Shows that the S and Δmetric value for MOPSO is less than other three algorithms and hence its performance is better among all.

TABLE II THE RESULTS OBTAINED FOR C METRIC

	PSFGA	SPEA2	NSGA II	MOPSO
PSFGA	—	0.0000	0.0000	0.0000
SPEA2	0.2354	—	0.08534	0.0467
NSGA II	0.9446	0.2566	—	0.0856
MOPSO	0.9643	0.2867	0.09973	—

The values 0.9643 on the fourth row, first column means almost all solutions from final populations obtained by MOPSO dominate the solutions obtained by PSFGA. The values 0 on first row means that no solution from the nondominated population obtained by SPEA 2, NSGA II and MOPSO is dominated by solutions from final populations obtained by PSFGA. From result it clear that the performance of MOPSO significantly outperforms the compared algorithms in the considered optimal weighting of assets as a portfolio design problem.

VIII. CONVERGENCE CHARACTERISTICS

The Pareto fronts (between risk and return) generated by these four algorithms are depicted as:

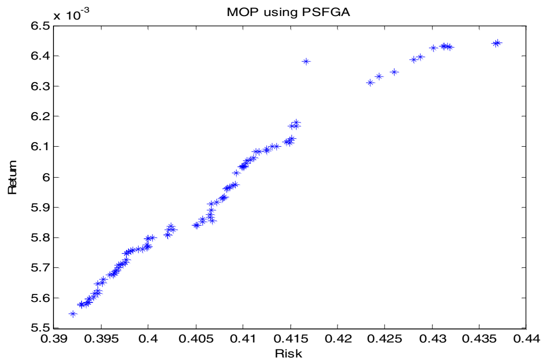

Figure 1. PSFGA

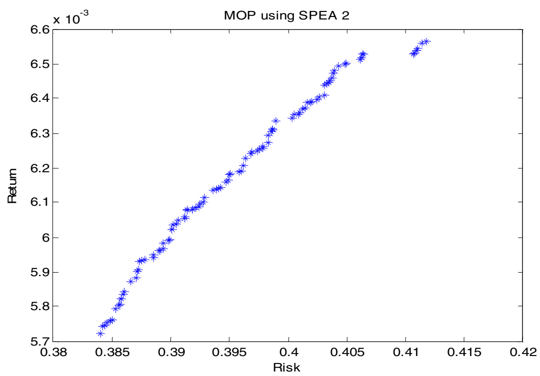

Figure 2. SPEA 2

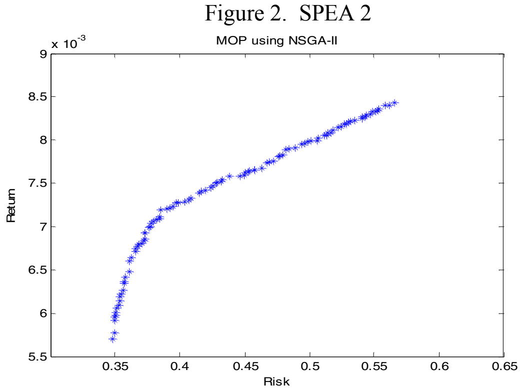

Figure 3. NSGA II

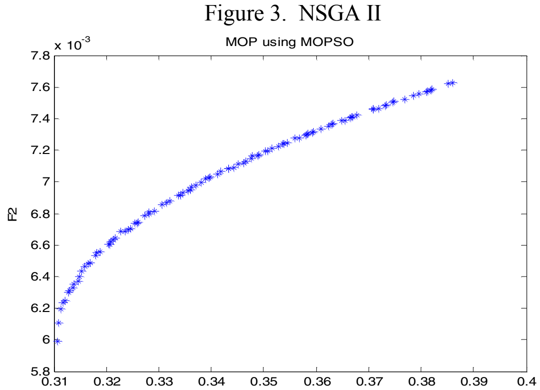

Figure 4. MOPSO

IX. CONCLUSIONS AND FURTHER WORK

In this paper a comparative study of four multi-objective evolutionary algorithms for solving efficient weighting of assets portfolio optimization problem is discussed. The compared algorithms are PSFGA, SPEA2, NSGA2 and MOPSO. An assets set has been considered for the numerical experiments. Results have shown that the MOPSO significantly outperforms its counterparts in all experiments.

REFERENCES

[1]K. Deb, A. Pratab, S. Agrawal, and T. Meyarivan. "A Fast Elitist Non-Dominated Sorting Genetic Algorithm : NSGA II". IEEE transactions on Evolutionary Computation, pp.182-197,2002.

[2] H. M. Markowitz Portfolio Selection, Journal of Finance, 1(7):77-91,1952

[3] H. M. Markowitz. Portfolio Selection: efficient diversification of investments. John Wiley & Sons, 1959.

[4]E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation, 3(4):257-271, 1999.

[5] Hu, X., Shi, Y., & Eberhart, R. C. "Solving constrained nonlinear optimization problems with particle swarm optimization". In Proceedings of the sixth world multiconference on systemics, cybernetics and informatics 2002, Orlando, USA.

[6]Hu,X.,&Eberhart,R.C.,"Multiobjective ptimization using dynamic neighborhood particle swarm optimization"..IEEEworld congress on computational intelligence, Hawaii, May.2002 12–17.

[7]Coello, C. A. C., & Lechunga, M. S. " MOPSO: A proposal for multiple objective particle swarm optimization".IEEE world congress on computational intelligence, Hawaii, 2002, pp. 1051–1056.

[8] Liao, C., Tseng, C., & Luarn, P. A discrete version of particle swarm optimization for flow shop scheduling problems. Computers and Operations Research, 2007, 3099–3111.

[9] Niu, B., Zhu, Y., He, X., & Wu, H. " MCPSO: A multi-swarm cooperative particle swarm optimizer. Applied Mathematics and Computation" 2007, 1050–1062.