Florentina-Mihaela APIPIE, PhD Student E-mail: mihaela.apipie@gmail.com Department of Statistics and Informatics University of Craiova Professor Vasile GEORGESCU, PhD E-mail: v\_geo@yahoo.com Department of Statistics and Informatics University of Craiova

# ASSESSING AND COMPARING BY SPECIFIC METRICS THE PERFORMANCE OF 15 MULTIOBJECTIVE OPTIMIZATION METAHEURISTICS WHEN SOLVING THE PORTFOLIO OPTIMIZATION PROBLEM

Abstract. The financial market has undergone a major revolution in recent decades with the advance and spread of multiobjective optimization metaheuristics, which can be more successfully applied to various aspects of financial decisions. Portfolio optimization is one of them. The purpose of this paper is to adapt, implement in Matlab, assess and compare the performance of 15 metaheuristics belonging to four different classes (NSGA, MOPSO, MOEA/D and SPEA classes) when applying to the Markowitz's Portfolio Optimization Problem. For comparing the performance of these algorithms, we use several specific metrics quantifying the convergence and/or the diversity of the approximate Pareto front against the true Pareto front. The optimal portfolios in the sense of Pareto are selected from a universe of 20 assets listed on the Bucharest Stock Exchange.

*Keywords*: Multiobjective optimization metaheuristics, portfolio optimization, performance metrics, convergence, diversity.

## JEL Classification:C58, C61, C63, G11, G17

## 1. Introduction

The Modern Portfolio Theory is a subject that has generated a lot of research in the last sixty years, in particular since Henry Markowitz (1952) revolutionized the financial landscape with his Mean-Variance model. It consists of defining an efficient portfolio, that is, one that has a minimum level of risk for a given expected return or, equivalently, a maximum expected return for a given level of risk.

Many of the optimization problems that arise in the real world present multiple conflicting objectives. The portfolio selection problem is a typical example of a multiobjective optimization problem. It can now benefit from the spectacular advances in the field of Evolutionary Computing, where an impressive number of metaheuristics have been specifically designed to handle multiobjective optimization problems.

A multiobjective optimization problem (MOP) can be defined as:

$$MOP = \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{subject to } x \in S \end{cases}$$
(1)

where  $n \ (n \ge 2)$  is the number of objectives and  $x = (x_1, x_2, ..., x_k)$  is the vector that represents the decision variables. The search space S represents the decision space of the MOP, or the set of feasible solutions. The space to which the objective vector belongs is called the objective space. The set Y = F(S) represents the feasible points in the objective space.  $F(x) = (f_1(x), f_2(x), ..., f_n(x))$  is the vector of objectives to be optimized and  $y = F(x) = (y_1, y_2, ..., y_n)$  is a point in the objective space, with  $y_i = f_i(x)$ .



Figure 1. Correspondence between the search space and the objective space

Ideally, one would like to obtain a solution minimizing all the objectives. Suppose that the optimum for each objective is known to be optimized separately.

A point  $y^* = (y_1^*, y_2^*, ..., y_n^*)$  is an *ideal vector* if it minimizes each objective function  $f_i$  in F(x), that is:

$$y_i^* = \min(f_i(x)), \quad x \in S, \ i \in [1, n]$$
 (2)

The ideal vector is generally a utopic solution in the sense that it is not usually a feasible solution in the decision space. Most often, in MOPs with contradictory objectives there is no single solution that can be considered the best, but a set of alternatives that represent the best compromises among all the objectives, in the sense that each solution is better than the others in some objective, but none is better than another in all objectives simultaneously. This fact implies that it is not possible to reduce an objective without worsening, at least, one of the others.

Multiobjective optimization methods are based on important concepts such as dominance, Pareto optimality, Pareto optimal set or Pareto fronts. The concept of dominance is the basis of many algorithms to solve multi-objective problems 40

and is essential to recognize the set of optimal solutions. It is applied when comparing two solutions and the corresponding associated objective values. We say that a solution  $x_1$  dominates another solution  $x_2$ , if both conditions 1 and 2 are true:

1.Solution  $x_1$  is not worse than  $x_2$  in all objectives,  $f_j(x_1) \le f_j(x_2)$  for all values j = 1, 2, ..., m (minimum problem).

2.Solution  $x_1$  is strictly better than  $x_2$  for at least one objective,  $f_j(x_1) < f_j(x_2)$  for at least one value j = 1, 2, ..., m (minimum problem).

A solution  $x^* \in S$  is Pareto optimal if for all  $x \in S$ , F(x) does not dominate  $F(x^*)$ , i.e.:

$$F(x) \neg \prec F(x^*) \tag{3}$$

In MOPs we can have a set of solutions known as the Pareto optimal set. For a given MOP(F, S), the Pareto optimal set is defined as follows:

$$P^* = \{ x \in S \mid \nexists x' \in S, \ F(x') < F(x) \}$$
(4)

The image of this set in the objective space is known as the Pareto front. For a given MOP(F, S) and its corresponding Pareto optimal set  $P^*$ , the Pareto front is defined as:

$$PF^* = \{F(x), \ x \in P^*\}$$
(5)

Obtaining the Pareto front of an MOP is the main objective of the multiobjective optimization. However, since a Pareto front normally contain an infinite number of points, a good approximation to this front that contains a limited number of Pareto solutions may also be useful. These solutions should be as close as possible to the exact or true Pareto front and, in addition, should present a uniform distribution. If either of these two properties is not met, the approximation obtained may not be useful to the decision maker who must have complete information about the Pareto front.

Over the years many techniques for solving multi-objective optimization problems have been developed. The use of Evolutionary Algorithms (EAs) for the treatment of MOPs proved to be of a particular interest, both theoretical and practical. It has led to the so-called Multiobjective Evolutionary Algorithms (MOEAs). MOEAs belong to the category of population-based metaheuristics and can be seen as an iterative improvement applied to a population of solutions.

Multiobjective Evolutionary Algorithms (MOEAs) represent promising candidates to obtain a set of efficient portfolios, associated with various commitments between the performance and risk objectives of the selected portfolio.

The Markowitz model has been originally proposed as a single-objective optimization problem, in which either the expected return is maximized, subject to certain established risk, or equivalently, the risk is minimized for an established return.

The emergence of multiobjective optimization techniques enabled to consider the expected return and the risk simultaneously as two distinct goals and thus to solve the portfolio optimization problem as a Multi Objective Problem (MOP).

The set of specific investments, called portfolio, are denoted by  $w = (w_l, w_2, ..., w_N)$ , where  $w_i$  is the proportion (weight) of the capital that is invested in asset *i*. The two objectives will be denoted by  $f_1(w)$  and  $f_2(w)$ , which denote the risk of the portfolio quantified by its variance,  $\sigma_p^2 = w' \Sigma w$ , and the expected return,  $\mu_p = w' \mu$ , respectively. The corresponding model is formulated as follows:

$$\begin{cases} max & f_1(w) = w'\mu\\ min & f_2(w) = w'\Sigma w \end{cases}$$
(6)

subject to:

$$w'1 = 1 \tag{7}$$
$$0 \le w_i \le 1$$

where  $\Sigma = (\sigma_{ij})$  is the covariance matrix, with  $\sigma_{ij}$  the covariance between the assets *i* and *j*,  $\mu = (\mu_1, ..., \mu_N)'$  is the vector of expected returns of assets, 1 = (1, ..., 1)' is an N-vector of ones and N is the number of assets available in the market.

The equality constraint (called the budget constraint) indicates that the total available capital must be invested in the portfolio.

# 2. Multiobjective optimization metaheuristics included in our experiment

The Multiobjective optimization metaheuristics evolved over the years. The first generation of algorithms was characterized by the use of sharing, niching and a fitness function, combined with a Pareto ranking. The second generation of this type of tools introduced new algorithms, in addition to improving other existing ones. The concept of elitism was incorporated, which refers to the use of an external population to retain non-dominated individuals. The third generation extends from addressing multi-objective problems (typically two and three objectives) to many-objective problems that involve four or more objectives and are also capable of handling more complex kinds of constraints.

For the purposes of our experiment we have selected 15 multiobjective optimization metaheuristics belonging to four different classes (NSGA, MOPSO, MOEA/D and SPEA classes):

## A. Algorithms in NSGA class

1.NSGAII - Nondominated Sorting Genetic Algorithm II

2.gNSGAII- g-dominance based Nondominated Sorting Genetic Algorithm II

3.ANSGAIII - Adaptive Nondominated Sorting Genetic Algorithm III

4.NSGAIII – Nondominated Sorting Genetic Algorithm III

NSGA was one of the first Evolutionary Algorithms used to find multiple Pareto-optimal solutions in one single simulation run, but the main criticisms of the 42

NSGA approach was the high computational complexity of nondominated sorting specially for large population size, nonelitism approach and the need for specifying a sharing parameter.

Kalyanmoy Deb et al. (2002) proposed an algorithm called NSGA-II, which try to reduce the above three drawbacks. In the new algorithm they replaced the sharing function with a crowding distance approach. With this improvement was observed a better computational complexity and there is no need for any user to define parameter to maintain diversity.

Also, to improve the NSGA-II algorithm, Julián Molinaet al. propose a concept called g-dominance. This concept was designed to be used with any multiobjective metaheuristic (without modifying the main architecture of the main method) and it allows us approximate the efficient set around the area of the most preferred point without using any scalarizing function.

Kalyanmoy Deb et al.(2013) developed the evolutionary multiobjective optimization algorithms to solve many-objective optimization problems (four or more objectives). Their algorithm was called NSGA-III and was tested on test problemswith2 to 15 objectives. NSGA-III uses the basic framework of NSGA-II, but unlike NSGA-II, the maintenance of diversity among population members in NSGA-III is ensured by supplying and adaptively updating a number of wellspread reference points.

At the same time, Himanshu Jain et al. proposed an extension of NSGA-III to solve generic constrained many-objective optimization problems. During the study they observed that not all specified reference points will correspond to a Pareto-optimal solution and to solve this aspect they suggested an adaptive NSGA-III algorithm (ANSGA-III) that identifies non-useful reference points and adaptively deletes them and includes new reference points in addition to the supplied reference points.

# B. Algorithms in MOPSO class

5.MOPSO – Multi-Objective Particle Swarm Optimizer

6.CMOPSO- Competitive Mechanism based Multi-Objective Particle Swarm Optimizer

7.MMOPSO – Multiple Search Strategies based Multi-Objective Particle Swarm Optimizer

8.MOPSOCD – Multi-Objective Particle Swarm Optimizer using Crowding Distance

Carlos A. CoelloCoello et al. (2002) introduced the so called MOPSO, which allows particle swarm optimization (PSO) algorithm to be able to deal with multiobjective optimization problems. The approach is population-based and uses an external memory (called "repository") and a geographically-based approach to maintain diversity.

Another attempt to make the PSO algorithm to handle multiobjective optimization problems was made by Carlo R. Raquel et al. (2005) and called MOPSO-CD. MOPSO-CD incorporate the mechanism of crowding distance into

the algorithm of PSO specifically for global best selection and the deletion method of an external archive of nondominated solutions. MOPSO-CD also has a constraint handling mechanism for solving constrained optimization problems.

Qiuzhen Lin et al. (2015) proposed a new MOPSO algorithm called MMOPSO. The MMOPSO algorithm uses multiple search strategies to update the velocity of each particle, which is beneficial for the acceleration of convergence speed and for keeping of population diversity. Decomposition approach is exploited for transforming multiobjective optimization problems into a set of aggregation problems and then each particle is assigned accordingly to optimize each aggregation problem.

Xingyi Zhang et al. (2017) proposed a competitive mechanism based multi-objective PSO, termed CMOPSO. With this competitive mechanism-based learning strategy, the algorithm is able to achieve a better balance between convergence and diversity than original PSO. The main difference lies in the fact that the search process is guided by the competitors in the current swarm instead of the historical positions. Also, it does not need external archive.

C. Algorithms in MOEA/D class

9.DMOEA $\epsilon$ C – Decomposition-based Multiobjective Evolutionary Algorithm with the  $\epsilon$ -Constraint framework

10.EAG-MOEA/D – External Archive Guided Multiobjective Evolutionary Algorithm based on Decomposition

11.MOEA/D-PaS – Multiobjective Evolutionary Algorithm based on Decomposition using Pareto Adaptive Scalarizing strategy

12.MOEA/D-DRA – Multiobjective Evolutionary Algorithm based on Decomposition with Dynamical Resource Allocation

The Multiobjective evolutionary algorithm based on decomposition (MOEA/D) decomposes a multiobjective optimization problem into a number of scalar optimizations subproblems and each subproblem is optimized by using information only from its several neighbouring subproblems.

Qingfu Zhang et al.(2009) proposed MOEA/D-DRA using Tchebycheff approach as decomposition technic. They define a utility  $\pi_i$  for each subproblem *i*. So, computational efforts are distributed to subproblems based on their utilities, in comparison with MOEA/D where the subproblems receive about the same amount of computational effort.

Another improvement of MOEA/D was made by XinyeCai et al.(2015) and called EAG-MOEA/D. EAG-MOEA/Dworks with an internal (working) population and an external archive. Also, uses a decomposition-based strategy for evolving its working population and a domination-based sorting for maintaining the external archive. So, the domination-based sorting and the decomposition strategy can complement each other by using information extracted from the external archive to decide which search regions should be searched at each generation.

Further, Rui Wang et al. (2016) proposed a method called Pareto adaptive scalarizing (PaS) to improve the MOEA/D algorithm. The PaS avoids an estimation of the Pareto front geometrical shape and is computationally efficient.

# D. Algorithms in SPEA class

13.SPEA2 – Strength Pareto Evolutionary Algorithm 2

14.SPEA2+SDE – Strength Pareto Evolutionary Algorithm 2 with Shift-Based Density Estimation

 $15.\ensuremath{\text{SPEA/R}}\xspace - \ensuremath{\text{Strength}}\xspace \ensuremath{\text{Pareto}}\xspace$  Algorithm based on Reference direction

The SPEA algorithm was proposed by Zitzleret al.(1999) for finding the Pareto-optimal set for multiobjective optimization problems.

An improved version of SPEA, namely SPEA2, was proposed by Eckart Zitzler et al. (2001). In contrast with its predecessor, this version incorporates a fine-grained fitness assignment strategy which takes into account, for each individual, how many individuals it dominates and it is dominated by, a density estimation technique which allows a more precise guidance of the search process and an enhanced archive truncation method which guarantees the preservation of boundary solutions.

After SPEA2, Miqing Li et al. (2014) proposed a shift-based density estimation (SDE) strategy for modifying the diversity maintenance mechanism in the algorithm SPEA2. The new algorithm was called SPEA2+SDE. Their objective was to develop a general modification of density estimation in order to make Pareto-based algorithms suitable for many-objective optimization. SDE covers both the distribution and convergence information about individuals.

Shouyong Jiang et al. (2017), by introducing an efficient reference direction-based density estimator, a new fitness assignment scheme, and a new environmental selection strategy, for handling both multiobjective and many-objective problems, proposed a new improvement to SPEA algorithm, called SPEA/R. SPEA/R inherits the advantage of fitness assignment of SPEA2 in quantifying solutions' diversity and convergence in a compact form, but replaces the time-consuming density estimator by a reference direction-based one, and also takes into account both local and global convergence.

The aforementioned algorithms have been adapted and implemented in Matlab to solve the Markowitz portfolio selection problem as a bi-objective optimization problem with the default constraints defined in section 1.

#### **3. Performance Metrics**

In the context of multiobjective optimization it is not possible to find or enumerate all elements of the Pareto front. Hence to solve a multiobjective problem, one must look for the best discrete representation of the Pareto front. Evaluating the quality of a Pareto front approximation is not trivial. To compare

multiobjective optimization algorithms, the choice of good performance metrics is crucial.

Various performance metrics for measuring the quality of Pareto-optimal sets have been reported for MOEAs in the literature. Considering an approximated front A and a reference front R in the objective space, the performance metrics can be grouped as:

• *Cardinality metrics:* the cardinality of A refers to the number of nondominated points that exists in A. Intuitively, a larger number of points is preferred.

• *Convergence (accuracy) metrics:* measure how close the approximated front A of non-dominated points is from the true Pareto front ( $PF_{true}$ ) in the objective space. If  $PF_{true}$  is unknown, a reference front *R* is considered instead.

• Diversity metrics: check whether the points in the obtained nondominated front are as diverse as possible; both the distribution and spread of the Pareto front approximation are to be evaluated. The distribution refers to the relative distance among points in A, while the spread refers to the range of values covered by the points in A (the extent of the approximated front should be maximized, i.e., A should contain the extreme points of the true Pareto front).

The table below presents some of the most used performance metric:

Metric	Туре	Arity
Hypervolume(HV)	Accuracy and Diversity	Unary
Generational distance(GD)	Accuracy	Unary
Inverted generational distance(IGD)	Accuracy and Diversity	Unary
Averaged Hausdorff distance $(\Delta_p)$	Accuracy and Diversity	Unary
Spread:Delta indicator( $\Delta$ )	Diversity	Unary
Two set coverage(C)	Accuracy and Diversity	Binary
Spacing(S)	Diversity	Unary

**Table 1. The performance metrics** 

**Hypervolume (HV)** is an indicator of both the convergence and diversity of an approximation front. Given a set X of solutions and their image A in the objective space, the hypervolume of A is the volume generated by the relation of the points of the Pareto front obtained with a given reference point, called the nadir point. The latter is usually chosen to be the worst values reached for each objective of the problem, thus guaranteeing that all the solutions of the obtained front will not be dominated by that corresponding to the nadir point. The HV of X is the sum of the volumes ( $V_x$ ) formed between each point in A and the reference point:

$$HV = \bigcup_{x_i \in X} V_x \tag{8}$$

Since it uses the nadir point as a reference, the HV calculation does not depend on the availability of an optimal Pareto front. One of the main advantages of hypervolume is that it is able to capture in a single number both the closeness of the solutions to the optimal set and, to some extent, the spread of the solutions across objective space.



Figure 2. Hypervolume calculation for a bi-objective minimization problem

**Generational distance (GD)** is used to measure the proximity of the approximate front A found by the algorithm from a reference front R, which is either the true Pareto front or a very good approximation to it. The distances between each objective vector a in A and the closest objective vector r in R are averaged over the size of A. Formally,

$$GD_p(A,R) = \frac{1}{|A|} \left( \sum_{a \in A} \min_{r \in R} d(r,a)^p \right)^{\frac{1}{p}}, \quad d(a,r) \sqrt{\sum_{m=1}^M (a_m - r_m)^2}$$
(9)

where M is the number of objectives. When p=2, GD2(A,R) is the average Euclidean distance.

The GD metric is fast to compute and correlates with convergence to the reference front, but is very sensitive to the number of points found by a given algorithm. Thus, large approximation fronts of poor quality may be ranked highly by GD.

The **inverted generational distance** (**IGD**) was proposed as an improvement over the GD based on the very simple idea of reversing the order of the fronts considered as input by the GD, i.e., IGD(A, R) = GD(R, A)

$$IGD_p(A,R) = \frac{1}{|R|} \left( \sum_{r \in R} \min_{a \in A} d(r,a)^p \right)^{1/p}$$
(10)

In other words, the IGD equals the GD metric when computing the distance between each objective vector in the reference front and its closest

DOI: 10.24818/18423264/53.3.19.03

47

objective vector in the approximation front, averaged over the size of the reference front.

The main advantages of the IGD measure are twofold. One is its computational efficiency, the other is its capability to show the overall quality of an obtained approximation front A (i.e., its convergence to the Pareto front and its diversity over the Pareto front).

The **averaged Hausdorff distance** ( $\Delta p$ ) was proposed as an attempt to address potential drawbacks of the IGD. It is defined as an averaged Hausdorff distance metric, controlled by the parameter *p*. In particular, larger values of *p* mean stronger penalties for outliers. The formal definition of  $\Delta p$  is given as follows:

$$\Delta_{p}(A,R) = max \left( IGD_{p}(A,R), IGD_{p}(R,A) \right)$$
(11)

**Spread metric** ( $\Delta$ ) examines how evenly the solutions are distributed among the approximation fronts in objective space. First, it calculates the Euclidean distance between the consecutive solutions in the obtained nondominated set of solutions. Then it calculates the average of these distances. After that, from the obtained set of non-dominated solutions the extreme solutions are calculated. Finally, using the following metric it calculates the nonuniformity in the distribution:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (n-1)\bar{d}}$$
(12)

where  $d_i$  is the Euclidean distance between consecutive solutions  $d_f$  and  $d_i$  are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained nondominated set. The parameter  $\bar{d}$  is the average of all distances  $d_i$ , i = 1, 2, ..., (n-1), where *n* is the number of solutions on the best nondominated front. A low value of  $\Delta$  metric indicates wide and uniform spread out of the solutions across the Pareto front. Thus,  $\Delta = 0$  indicates that the approximation front is as uniformly distributed as possible.



DOI: 10.24818/18423264/53.3.19.03

48

The **spacing metric** (S) was proposed by Schott (1995) and measures the dispersion of the obtained approximate front in comparison with the optimal Pareto front. It is given by the following relationship:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (d_i - \bar{d})^2}$$
(13)

where  $d_i$  is the Euclidean distance in objective space between point *i* of the obtained approximate front and point *j* most closely belonging to the Pareto optimal front;  $\overline{d}$  is the mean of all  $d_i$  and *n* is the number of solutions present in the obtained approximate front. The spacing must be as small as possible for the solution set to be of superior quality. A value of zero for this metric indicates that the solutions in the approximate front are equidistantly spaced.

**Coverage of two sets (C)** metric was proposed by Zitzler and Thiele (1999) and compares the quality of two non-dominated sets. Let A and B be two different sets of non-dominated solutions, then the C metric maps the ordered pair (A, B) into the interval [0, 1]:

$$C(A,B) = \frac{|\{b \in B | \exists a \in A : a \succ b \text{ or } a = b\}|}{|B|}$$
(14)

where *a* and *b* are candidate solutions of sets *A* and *B* respectively and a > b means that *a* dominates *b*. If C(A, B) = 1, all the candidate solutions in *B* are dominated by at least one solution in *A*. If C(A, B) = 0, no candidate solutions in *B* is dominated by any solution in *A*.

#### 4. Experimental results

In the experimental work addressed in this paper, the optimal portfolios in the sense of Pareto are selected from a universe of 20 assets listed on the Bucharest Stock Exchange. The data used is the quoted shares listed from January 1, 2012 to December 14, 2018. The results of applying the 15 metaheuristics to solve the Markowitz's Portfolio Optimization Problem are summarized by means of a series of graphics and tables. Apart from specific settings, all 15 metaheuristics have in common the following settings: Population size = 100 and Maximum number of function evaluations = 50000.

We choose to present in more detail the results for two metaheuristics out of 15: CMOPSO that performs better with respect to most of the performance metrics for our datasets and the problem at hand, and MOPSO that appears to be the worst performer in the same context. Finally, we will summarize the results for all 15 algorithms and present an overall rank from multiple ranked lists.

# 4.1.CMOPSO (Competitive Mechanism based Multi-Objective Particle Swarm) – the best performer

As can be seen in Fig. 4, CMOPSO performs very well in terms of both convergence (the computed Pareto front is very close to the true Pareto front) and diversity (the points of the computed Pareto front are uniformly distributed along the true Pareto front).



Figure 4.Comparing True Pareto Front with Pareto Front obtained using CMOPSO

The Performance metrics are shown	n in	Figure 5	5
-----------------------------------	------	----------	---

Pareto Front Compare PF	Metrics	Metrics Plot	Efficientfronti	er PF Summary	Risks&Retu	rns Weights
Metric		Iter_10000	Iter_20000	Iter_30000 li	ter_40000	Iter_50000
Coverage (min)		1	1	0.9900	0.9900	0.9900
Diversity Metric (max)		0.8453	0.8535	0.8262	0.8658	0.8435
Averaged Hausdorff distance	0.0019	0.0016	0.0017	0.0016	0.0016	
Generational distance (min)	9.2138e-05	7.9064e-05	8.7181e-05	7.7804e-05	7.4715e-05	
Hypervolume (max)		0.5558	0.5754	0.5818	0.5857	0.5857
Inverted generational distance	(min)	0.0019	0.0016	0.0017	0.0016	0.0016
Pure diversity (max)	434.0796	542.5991	550.8940	416.7562	460.0524	
Spacing (min)	0.0012	0.0012	0.0013	0.0012	0.0011	
Spread (min)		0.1471	0.1506	0.1651	0.1489	0.1351

Figure 5. Performance metrics for CMOPSO

Apart from the Efficient frontier computed using CMOPSO, Fig. 6. Shows the 20 assets constituting the universe from which the optimal portfolios are formed. The riskiest portfolio is that one containing only one asset (TBM).

Figs. 7, 8 and 9 display various numerical results: a Pareto front summary, the daily and annualized risks and returns and the portfolio weights computed using CMOPSO.



Figure 6. Efficient frontier using CMOPSO and the 20 assets constituting the universe

Pareto Front	Compare PF	Metrics	Metrics Plot	Effici	entfrontier	PF Summary	Risks&Returns	Weights
Measure	Risk	Retum	Annualized	Risk	Annualized	d_Retum		
Min	0.0086	3.8368e-04	C	.1357		0.0959		
Max	0.0344	0.0014	C	.5441		0.3594		
Range	0.0258	0.0011	C	.4085		0.2635		
Std	0.0083	2.9443e-04	C	.1310		0.0736		
Mean	0.0188	0.0010	C	.2966		0.2550		

Figure 7. Pareto front summary for CMOPSO

Pareto Front C	ompare PF	Metrics	Metrics	Plot E	Efficient fro	ntier	PF Sumr	mary Risks	&Returns	Weights
Variable	Port1	Por	t2	Port3	Port4		Port5	Port6	Port7	Port
Risk	0.00	0 86	.0086	0.008	87 0.00	88	0.0091	0.0094	0.0098	3 O.
Return	3.8368e	-04 4.487	7e-04 5	.0220e-0	04 5.5074e	-04 6	.0990e-04	6.6674e-04	7.1524e-04	4 7.6610
Annualized_Risk	0.13	357 0	.1366	0.137	76 0.13	95	0.1432	0.1483	0.1543	3 <b>0</b> .
Annualized_Retu	rn 0.09	959 0	.1122	0.125	55 0.13	377	0.1525	0.1667	0.1788	<b>3</b> O.
<										>

Figure 8. Daily and annualized risks and returns for CMOPSO

Pareto Front	Compare	PF Metrics	Metrics Pl	ot Efficien	tfrontier	PF Summary	Risksℜ	turns Weig	jhts
Ticker	Port1	Port2	Port3	Port4	Port5	Port6	Port7	Port8	
ATB	0.2375	0.2122	0.2110	0.1957	0.1757	0.1261	0.0760	0	1
BIO	0.1438	0.1346	0.1154	0.1108	0.1344	0.1270	0.1106	0.1201	
BRD	0.0586	0.0567	0.0034	0.0016	(	) 0	1.4810e-04	0.0290	8
BRK	0.0067	6.5782e-04	2.4867e-04	0	3.8306e-06	6 0	2.8320e-06	0	
CMP	0.0603	0.0685	0.0883	0.1013	0.0901	0.1272	0.1356	0.1529	
EBS	0.0052	0.0078	0.0096	0.0090	0.0115	0.0093	0.0085	0.0063	
ELMA	0.0039	0.0034	7.5660e-04	2.2430e-04	5.1048e-05	9.9047e-05	5.4451e-05	5.3841e-05	8
FP	0.1769	0.2310	0.2476	0.2419	0.3029	0.3340	0.3024	0.2960	
OIL	0.0450	0.0411	0.0461	0.0166	(	) 0	0	0	
RRC	0.0292	0.0224	0.0355	0.0207	0.0127	0.0073	0.0036	0.0045	
SIF1	0	0.0276	0.0340	0.0363	0.0652	0.0943	0.1314	0.1565	
SIF2	4.8583e-04	1.1317e-04	0	7.8826e-04	3.0916e-04	3.1656e-04	2.5238e-04	8.8368e-05	6
SIF3	0	0	0	0	(	) 0	0	0	
SIF4	0.0397	0.0087	0	0.0016	(	) 0	0	0	
SIF5	0.0347	0.0357	0.0702	0.0566	0.0614	0.0253	0.0203	0.0287	6
SNP	0.0462	0.0393	0.0153	0.0116	0.0053	0.0018	4.0422e-04	0.0013	
TBM	0.0077	0.0122	0.0258	0.0330	0.0516	0.0667	0.0848	0.1144	
TEL	0.0187	0.0146	0.0025	0.0194	0.0089	0.0028	0.0070	0.0031	
TGN	0.0586	0.0485	0.0541	0.0574	0.0167	0.0078	0.0194	0.0033	
TLV	0.0268	0.0350	0.0401	0.0855	0.0632	0.0699	0.0994	0.0836	
<									>

Figure 9. Portofolio weights for CMOPSO

# 4.2.MOPSO (Multi-Objective Particle Swarm Optimizer) - the worst performer

Fig. 10 shows why MOPSO is the worst performer when using the same running settings for all metaheuristics (in our case, Population size = 15 and Maximum number of function evaluations = 50000). Indeed, we observe that, MOPSO cannot approximate well the efficient frontier in term of both convergence (the computed Pareto front departs significantly from the true Pareto front) and diversity (the points of the computed Pareto frontier does not distribute uniformly along the true Pareto frontier, but concentrate only in a part of the two-dimensional objective space).

However, it is worth to mention that increasing default values for Population size and Maximum number of function evaluations, MOPSO becomes capable to approximate well the true Pareto front. So MOPSO converges much slowly, compared to CMOPSO.



#### Figure 10. Comparing True Pareto Front with Pareto Front obtained using MOPSO The Defense metrics are shown in Figure 11

1

Pareto Front Compare PF Metrics	Metrics Plot	Efficientfrontie	r PF Summary	Risks&Ret	urns Weights
Metric	Iter_10000	Iter_20000	Iter_30000 It	ter_40000	Iter_50000
Coverage (min)	1	1	1	1	1
Diversity Metric (max)	0.2009	0.2627	0.3318	0.3459	0.3236
Averaged Hausdorff distance (min)	0.0976	0.0809	0.0714	0.0681	0.0674
Generational distance (min)	0.0025	0.0029	0.0030	0.0028	0.0028
Hypervolume (max)	0.5215	0.5223	0.5140	0.5240	0.5177
Inverted generational distance (min)	0.0976	0.0809	0.0714	0.0681	0.0674
Pure diversity (max)	95.2746	166.1965	137.7785	157.3645	175.6856
Spacing (min)	9.5761e-04	0.0011	0.0014	0.0014	0.0013
Spread (min)	0.9750	0.9604	0.9612	0.9566	0.9856

Figure 11. Performance metrics for MOPSO

Fig. 12 shows the inaccurate representation of Efficient frontier using MOPSO and the 20 assets constituting the universe. Figures 13, 14 and 15 display various numerical results.



Figure 12. Inaccurate representation of Efficient frontier using MOPSO and the 20 assets constituting the universe

Pareto Front	Compare PF	Metrics	Metrics Plot Effici	ent frontier PF	Summary	Risks&Returns	Weights
Measure	Risk	Retum	Annualized_Risk	Annualized_Re	etum		
Min	0.0089	5.0959e-04	0.1413	0	).1274		
Max	0.0176	9.2809e-04	0.2776	0	0.2320		
Range	0.0086	4.1850e-04	0.1363	0	0.1046		
Std	0.0025	1.1270e-04	0.0400	0	0.0282		
Mean	0.0127	7.3919e-04	0.2005	0	0.1848		

Figure 13. Pareto front summary for MOPSO

Pareto Front Comp	are PF Met	rics Metric	s Plot Effic	ient frontier	PF Summa	ary Risks&	Returns V	/eights
Variable	Port1	Port2	Port3	Port4	Port5	Port6	Port7	Ро
Risk	0.0089	0.0090	0.0091	0.0093	0.0096	0.0099	0.0102	2
Return	5.0959e-04	5.3193e-04	5.5317e-04	5.7266e-04	5.9640e-04	6.2041e-04	6.3956e-04	6.61
Annualized_Risk	0.1413	0.1424	0.1445	0.1471	0.1514	0.1567	0.1616	;
Annualized_Return	0.1274	0.1330	0.1383	0.1432	0.1491	0.1551	0.1599	
<								>

Figure 14. Daily and annualized risks and returns for MOPSO

Pareto Front	Compare	PF Metrics	Metrics Pl	ot Efficien	tfrontier	PF Summary	Risks&Ret	turns Weigl	hts
Ticker	Port1	Port2	Port3	Port4	Port5	Port6	Port7	Port8	
ATB	0.1740	0.1643	0.1562	0.1483	0.1377	0.1277	0.1199	0.1110	
BIO	0.0865	0.0817	0.0777	0.0737	0.0685	0.0635	0.0596	0.0552	
BRD	0.0133	0.0242	0.0343	0.0436	0.0550	0.0665	0.0756	0.0858	
BRK	0.0019	0.0018	0.0017	0.0016	0.0015	0.0014	0.0013	0.0012	
CMP	0.1095	0.1034	0.0983	0.0933	0.0867	0.0803	0.0754	0.0698	
EBS	0.0084	0.0153	0.0219	0.0278	0.0351	0.0425	0.0484	0.0549	
ELMA	4.5715e-04	4.3164e-04	4.1039e-04	3.8957e-04	3.6185e-04	3.3543e-04	3.1499e-04	2.9167e-04	2
FP	0.1709	0.1614	0.1534	0.1457	0.1353	0.1254	0.1178	0.1091	
OIL	8.4541e-04	7.9821e-04	7.5893e-04	7.2042e-04	6.6916e-04	6.2029e-04	5.8250e-04	5.3938e-04	4
RRC	0.0606	0.0572	0.0544	0.0516	0.0480	0.0444	0.0417	0.0386	
SIF1	0.0268	0.0346	0.0396	0.0455	0.0545	0.0619	0.0677	0.0744	
SIF2	0	0	0	0	0	0	0	0	
SIF3	0	0	0	0	0	0	0	0	
SIF4	0.0428	0.0409	0.0381	0.0359	0.0340	0.0317	0.0295	0.0273	
SIF5	0.0328	0.0358	0.0361	0.0377	0.0415	0.0443	0.0458	0.0479	
SNP	8.6668e-06	8.1829e-06	7.7802e-06	7.3854e-06	6.8599e-06	6.3590e-06	5.9715e-06	5.5295e-06	4
TBM	0.0236	0.0445	0.0649	0.0832	0.1052	0.1278	0.1458	0.1660	
TEL	0.0254	0.0240	0.0228	0.0216	0.0201	0.0186	0.0175	0.0162	
TGN	0.1214	0.1146	0.1089	0.1034	0.0961	0.0890	0.0836	0.0774	
TLV	0.1009	0.0953	0.0906	0.0860	0.0799	0.0740	0.0695	0.0644	
<									>

Figure 15. Portfolio weights for MOPSO

A summary of the most significant performance metrics is given in the table 2. We marked in bold the best value obtained for each metric.

In table 3 we present a score for the results obtained by our 15 algorithms tested. The best position was rated with 1 (positions was numbered from 1 to 15) for each metric. The final score was obtained by summarizing the position obtained by each algorithm for all those 6 metrics presented in the Table 2. The best performance was obtained by CMOPSO and the worst performer was MOPSO.

Metrics	Averaged	Generational	Hypervolume	Inverted	Spacing	Spread
	Hausdorff	distance	(max)	generational	(min)	(min)
	distance	(min)		distance		
Algorithm	(min)			(min)		
Name						
DMOEAeC	0.0026	2.96E-05	0.5387	0.0026	0.0029	0.3561
EAGMOEAD	0.0021	8.46E-05	0.579	0.0021	0.0018	0.2686
MOEADDRA	0.0091	1.90E-05	0.5817	0.0091	0.0208	1.2295
MOEADPaS	0.0030	4.87E-05	0.5895	0.0030	0.0034	0.4625
CMOPSO	0.0016	7.47E-05	0.5857	0.0016	0.0011	0.1351
MMOPSO	0.0019	9.13E-05	0.5723	0.0019	0.0024	0.3541
MOPSO	0.0674	0.0028	0.5177	0.0674	0.0013	0.9856
MOPSOCD	0.0021	1.40E-04	0.5829	0.0021	0.0021	0.3092
ANSGAIII	0.0025	6.19E-05	0.5508	0.0025	0.0029	0.3108
gNSGAII	0.0022	1.17E-04	0.561	0.0022	0.0019	0.3246
NSGAII	0.0024	1.16E-04	0.5448	0.0024	0.0023	0.3495
NSGAIII	0.0025	5.35E-05	0.5527	0.0025	0.0021	0.2939
SPEA2	0.0032	1.26E-04	0.5851	0.0032	0.0036	0.3126
SPEA2SDE	0.0027	6.25E-05	0.5382	0.0027	0.0023	0.4844
SPEAR	0.0024	2.60E-05	0.5501	0.0024	0.0034	0.3536

Table 2. The performance metrics

# Table 3. Scores obtained for each algorithm

Algorithm name	Score	Algorithm name	Score
CMOPSO	14	MOEADPaS	44
EAGMOEAD	26	NSGAII	47
MOPSOCD	33	DMOEAeC	49
NSGAIII	34	SPEA2	51
MMOPSO	38	MOEADDRA	54
gNSGAII	39	SPEA2SDE	56
ANSGAIII	41	MOPSO	70
SPEAR	41		

# 5. Conclusions and further work

In this paper we adapted, implemented in Matlab, assessed and compared the performance of 15 metaheuristics belonging to four different classes (NSGA, MOPSO, MOEA/D and SPEA classes) when applying to the Markowitz's Portfolio Optimization Problem.

The fact there is no universally efficient algorithm is consensually accepted in the realm of optimization algorithms in general, and multiobjective optimization metaheuristics in particular. Several no Free Lunch theorems have been proposed under specific assumptions (e.g., (Schumacher et al., 2001) proved that no optimization technique has performance superior to any other over any set of functions closed under permutation). However, in practice some algorithms perform better than others for a specific problem, motivating the effort of finding the right algorithms for the right type of problem. A substantial improvement has been also obtained in the design principles of metheuristics: thus, introducing an external archive of nondominated solutions can lead to multiobjective optimizers that are better than others.

In general, our experimental work shows than some representatives of the new generations of multiobjective optimization metaheuristics perform better than the older and selecting them can be beneficial in solving problems such as Portfolio Optimization. On the other hand, there is no a commonly accepted framework for performance comparison of optimization algorithms. In section 3 we overviewed a set of performance metrics to measure the convergence (accuracy) and/or the diversity of the Pareto front approximation when compared with the true Pareto front. However, expecting all these metrics to produce similar rankings is unlikely. The overall performance of the selected metaheuristics for the problem at hand can only be assessed by the aggregation of single-criterion rankings. However, the aggregation method should be carefully chosen to avoid the Condorcet Paradox that may occur in such cases.

Portfolio Optimization problems with more complex constraints are also intended to be addressed using Penalty Function and Repair strategies as well as metaheuristics incorporating a constraint handling mechanism (e.g., ANSGA-III and MOPSO-CD).

# REFERENCES

 [1] Cai, X., Li, Y., Fan, Z. and Zhang, Q.(2015), An External Archive Guided Multiobjective Evolutionary Algorithm Based on Decomposition for Combinatorial Optimization; IEEE Transactions on Evolutionary Computation, 19(4): 508-523;

[2] Chen, J., Li, J. and Xin, B. (2017),*DMOEA-εC: Decomposition-based Multiobjective Evolutionary Algorithm with the ε-constraint Framework*; IEEE Transactions on Evolutionary Computation, 21(5), 714-730;

[3] CoelloCoello, C. A. and Lechuga, M. S. (2002), *MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization*; Proceedings of the IEEE Congress on Evolutionary Computation, 1051-1056;

[4] CoelloCoello, C. A. and Lechuga, M. S. (2002), *MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization*; Proceedings of the IEEE Congress on Evolutionary Computation, 1051-1056;

[5] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.(2002), *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*; IEEE Transactions on Evolutionary Computation, 6(2), 182-197;

[6] Deb, K. and Jain, H. (2014), An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Non-Dominated Sorting Approach, Part I: Solving Problems with Box Constraints; IEEE Transactions on Evolutionary Computation, 18(4), 577-601;

[7] Jain, H. and Deb, K. (2014), *An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Non-Dominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach*; IEEE Transactions on Evolutionary Computation, 18(4), 602-622;

[8] Jiang, S. and Yang, S. (2017), A Strength Pareto Evolutionary Algorithm Based on Reference Direction for Multiobjective and Many-Objective Optimization; IEEE Transactions on Evolutionary Computation, 21(3), 329-346;
[9] Li, M., Yang, S. and Liu, X. (2014), Shift-based Density Estimation for Pareto-based Algorithms in Many-objective Optimization; IEEE Transactions on Evolutionary Computation, 18(3), 348-365;

[10] Lin, Q., Li, J., Du, Z., Chen, J. and Ming, Z. (2015), *A Novel Multi-Objective Particle Swarm Optimization with Multiple Search Strategies*; European Journal of Operational Research, 247(3), 732-744;

[11] Markowitz, H. (1952), *Portfolio Selection*; The journal of finance, 7(1), 77-91;

[12] Molina, J., Santana, L. V., Hernandez-Diaz, A.G., CoelloCoello, C. A. and Caballero, R. (2009),*g-dominance: Reference Point Based Dominance for Multiobjective Metaheuristics*; European Journal of Operational Research, 197(2), 685-692;

[13] Raquel, C. R. and Naval Jr, P. C. (2005), *An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization*; Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, 257-264;

[14] Riquelme, N., Von Lucken, C., Baran, B. (2015), *Performance Metrics in Multi-Objective Optimization*; XLI Latin Computing Conference (CLEI);

[15] Schumacher, C., Vose, M. and Whitley D. (2001), *The No Free Lunch and Problem Description Length*, in: Genetic and Evolutionary Computation Conference, GECCO-2001, 565-570;

[16] Wang, R., Zhang, Q. and Zhang, T. (2016), *Decomposition-based Algorithms using Pareto Adaptive Scalarizing Methods*; IEEE Transactions on Evolutionary Computation, 20(6), 821-837;

[17] Zhang, Q., Liu, W. and Li, H. (2009),*The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances*; Proceedings of the IEEE Congress on Evolutionary Computation, 203-208;

[18] **Zitzler, E., Laumanns, M. and Thiele, L. (2001)**,*SPEA2: Improving the Strength Pareto Evolutionary Algorithm*; Proceedings of the Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, 95-100.