



Time-stamped Resampling for Robust Evolutionary Portfolio Optimization

Sandra García^{a,**}, David Quintana^a, Inés M. Galván^{a,*}, Pedro Isasi^a

^aComputer Science Department, Carlos III University of Madrid
Avda. Universidad 30, 28911 Leganes, Spain
<http://www.evannai.inf.uc3m.es>

Abstract

Traditional mean-variance financial portfolio optimization is based on two sets of parameters, estimates for the asset returns and the variance-covariance matrix. The allocations resulting from both traditional methods and heuristics are very dependent on these values. Given the unreliability of these forecasts, the expected risk and return for the portfolios in the efficient frontier often differ from the expected ones. In this work we present a resampling method based on time-stamping to control the problem. The approach, which is compatible with different evolutionary multiobjective algorithms, is tested with four different alternatives. We also introduce new metrics to assess the reliability of forecast efficient frontiers.

Keywords: Financial Portfolio Optimization, Robust Portfolio, Multiobjective Evolutionary Algorithms

1. Introduction

Asset allocation is one of the core topics in financial management. The search for the optimal choice of financial assets to be included in a portfolio has been the subject of research for a long time and it is one of the most active lines in finance.

The academic literature on this subject is very large and mostly based on the work of Markowitz [12, 9]. Under this framework, the problem is introduced as a multiobjective optimization problem where the investor tries to figure out the weight that each of the investment alternatives should carry in the portfolio. The target of this investor would be both minimizing risk and maximizing return at the same time. The solution to the problem of optimizing for these two objectives in conflict defines a set of solutions called the efficient frontier. This Pareto front consists on portfolios that are neither better or worse than the rest. For each level of risk or return, there is no better alternative in terms of the other objective. This makes the election of one of them a choice to be made by investors according to the way they weight risk and rewards.

The basic version of the problem can be solved using Quadratic Programming (QP). Unfortunately, this approach is built on a set of assumptions that are unlikely to hold in the real world. The quest for alternatives has driven attention to metaheuristics that might not suffer this limitation. This is the reason why the framework of evolutionary computation is getting traction on this area.

*Corresponding author

**Principal corresponding author

Email addresses: sgrodrig@inf.uc3m.es (Sandra García), dquintan@inf.uc3m.es (David Quintana), igalvan@inf.uc3m.es (Inés M. Galván), isasi@ia.uc3m.es (Pedro Isasi)

The solutions based on metaheuristics mostly fall into one of these two categories: solutions that transform the multi-objective problem into a single-objective form, and those who deal with it using a multiobjective approach.

Among the first group, we would mention the work by Soleimani et al. [21]. These authors use a genetic algorithm and extend the mentioned classic mean-variance optimization model and consider constraints on transaction costs, round lots or cardinality. They tackle the multiobjective nature of the problem minimizing the risk objective while setting a range of minimum acceptable returns in the constraints. Chiranjeevi and Sastry [4] use another popular approach to transform the multiobjective problem into a version that can be handled by single-objective algorithms. Instead of keeping one of the objectives on the objective function and the other in the constraint, they consider the objectives in the fitness function weighting them. In this particular case, they manage five objectives that are obtained breaking down the basic two. Chang et al. [3] suggest a similar solution and optimize a function that weights risk and return using a risk aversion parameter. Zhu et al. [23] target a popular metric, the Sharpe Ratio, that combines both elements into a single expression.

The development of Multiobjective Evolutionary Algorithms (MOEAs) has resulted into a number of researchers exploring their performance in this area. Among these we could mention Skolpadungket et al. [20], who test a set of multi-objective algorithms (VEGA, SPEA2, NSGA-II...) on a constrained version of the two objective problem. More recently, Anagnostopoulos and Mamanis [1] compare the performance of different multiobjective algorithms, and Deb et al. [6] introduce a customized hybrid version of NSGA-II to tackle the problem. Finally, we will mention the work of Radziukyniene and Xilinskas [16], where authors compare FastPGA, MOCELL, AbYSS, and NSGA-II on both of the basic problems, and an extended version that considers the dividend yield as a third objective.

Despite of the amount of research on portfolio optimization, there are still open issues. A key one is the robustness of results provided by algorithms. Among the most important factors that asset managers have to consider when evaluating the results provided by any of the above-mentioned methods, there is reliability. Very often, the expected efficient frontier lies far from the real one. This is due to the fact that the estimates for the expected risk and returns for the assets in the solution, and the portfolios derived from them, are very inaccurate. Understandably, this results on mistrust by some practitioners. The search for solutions for this problem has cleared the way for the field of robust portfolio optimization. It is in this area where we focus our contribution. We introduce a time-stamping method to control the population that enhances the reliability of the solutions provided by MOEAs.

The process of optimizing the risk and return of a portfolio relies on two parameters: the estimates for the expected asset returns and the variance-covariance matrix. The values of these parameters are usually based on past data and they might be inaccurate due to, for instance, the presence of outliers. In this context, there are several potential ways to tackle the problem. The two most prevalent alternatives found in the literature rely on either putting an emphasis on having robust estimates for the parameters, or managing the optimization process itself. The first one usually tries to filter the estimates to control, for instance, the influence of extreme past events on their computation [14]. The authors focusing on the second alternative design approaches handle uncertainty in the parameters during the optimization process [15, 22]. The alternative suggested in this paper falls in the latter category. We will enhance the solutions of MOEAs testing the population for different values for the parameters, and selecting the portfolios that consistently offer a good performance.

Optimizing for a single scenario, a set of expected asset returns and the use of a single variance-covariance matrix, bears the risk of getting solutions that might be extremely sensitive to deviations. This could potentially be a problem as it is almost certain that the estimates will not be accurate. We have to bear in mind that having perfect estimates for the expected returns for instance, implies that we can make perfect predictions for future prices, which is highly unlikely. For this reason we consider that assessing the candidate solutions in different likely scenarios and favoring those that consistently offer a good performance, might be a promising approach. The requirement of consistency is key. In order to achieve it, we introduce a time-stamping mechanism that, together with performance in terms of risk and return, will drive the evolution process to find robust and stable solutions.

The approach introduced in this paper is related to alternatives based on resampling [19, 18]. The most comparable among the traditional approaches is the one described by Idzorek [10]. This author suggests using combining traditional QP with Monte Carlo simulation to derive a set of fronts that are merged into a single solution at a later stage. This idea is very interesting but, unfortunately, the approach suffers the shortcomings of QP, namely, the ability to deal with real-world constraints. This is the reason why we feel that adapting the idea to the framework of MOEAs is very promising, as they do not suffer from this limitation. There is a previous effort based on evolutionary algorithms along the lines of this work, but is based on a very simple resampling approach that optimizes for a different scenario

in each generation[8]. The time-stamping mechanism that we introduce in this work is based on the previous one extending the problem with a third implicit objective that favors solutions that are consistently reliable. As we will see in the experimental section, this results on significantly higher robustness.

The proposed approach is compatible with a wide array of MOEAs. Given their different nature and behavior, we will test the approach on a set of popular algorithms. Specifically, the experimental section will consider NSGA-II, SPEA2, SMPSO and GDE3. NSGA-II [5] is one the most referenced algorithms in the field of multiobjective optimization. This one, together with SPEA2 [24] have been confirmed [20] to offer good performance in portfolio optimization. Apart from the mentioned two, we consider GDE3 [11], a differential evolution strategy, and SMPSO, [13] a multiobjective algorithm based on particle swarm optimization.

In the context of multiobjective problems, an important issue is the metric used to evaluate the solutions. It is generally admitted that there is no ideal single metric that should be used to evaluate different objectives simultaneously in every circumstance. The metrics that are most commonly used in this field, such as Hypervolume, Spread or SetCoverage are not appropriate indicators of stability in this context. For this reason, we define a set of metrics that capture different aspects of robustness in efficient frontiers. These, Estimation Error, Stability, Extreme Risk and Unrealized Returns draw on the basic principle that the expected risk and returns for the portfolios in the solution should be close to the observed ones ex-post.

The rest of the paper is organized as follows. First, we make a formal introduction to the financial portfolio optimization problem. After, we describe in detail the evolutionary approach proposed in this work. This section includes a brief description of the MOEAs, the solution encoding and the fitness mechanism in order to find robust and stable portfolios. Next, the different metrics used in this work to evaluate robustness of solutions are described. That will be followed by the experimental results and, finally, there will be a section devoted to summary and conclusions.

2. Financial Portfolio Optimization Problem

Financial portfolios can be defined as a collection of investments or assets held by an institution or a private individual. The Modern Portfolio Theory was originated in the article published by Harry M. Markowitz, in 1952 [12]. It explains how to use the diversification to optimize the Portfolio. In general, the portfolio optimization problem is the choice of an optimum set of assets to include in the portfolio and the distribution of investor's wealth among them. Markowitz [9] assumed that solving the problem requires the simultaneous satisfaction of maximizing the expected portfolio return $E(R_p)$ and minimizing the portfolio risk (variance) σ_p^2 , that is, solving a multiobjective optimization problem with two output objective functions [21, 4, 20, 16]. The portfolio optimization problem can be formally defined as:

- Minimize the **risk** (variance) of the portfolio:

$$\sigma_p^2 = \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} \quad (1)$$

- Maximize the **return** of the portfolio:

$$E(R_p) = \sum_{i=1}^n w_i \mu_i \quad (2)$$

- Subject to:

$$\sum_{i=1}^n w_i = 1 \quad (3)$$

$$0 \leq w_i \leq 1; i = 1 \dots n \quad (4)$$

where n is the number of available assets, μ_i the expected return of the asset i , σ_{ij} the covariance between asset i and j , and w_i are the decision variables giving the composition of the portfolio. The constrains referenced in equations 3 and 4 require the full investment of funds and prevent the investor from shorting any asset, respectively. In a quantitative way, the risk is represented with the standard deviation σ_p .

The solution to the problem should also consider some real world constraints [2] such as:

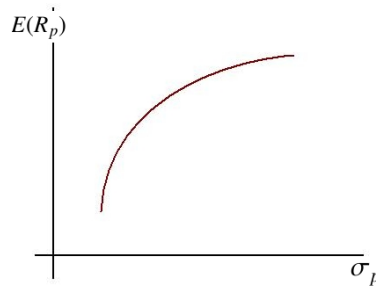


Figure 1. Efficient frontier

- Cardinality constraint: it is possible to define the maximum C_{max} and minimum C_{min} number of assets in which it is possible to invest ($w_i \neq 0$):

$$C_{min} \leq \sum(w_i \neq 0) \leq C_{max} \quad (5)$$

- Values limit constraint: each weight w_i must have a value in the interval $[lim_{inf}, lim_{sup}]$, where:

$$0.0 \leq lim_{inf} \leq w_i \leq lim_{sup} \leq 1.0 \quad (6)$$

All of these equations are solved by a set of points that constitute the efficient frontier of the problem. The points will define a curve similar to fig.1, plotted in the *risk-return* space of all possible portfolios. The points of this curve represent portfolios which have the minimum amount of risk given a certain expected return (and viceversa).

3. Evolutionary Approach for Robust Portfolio Optimization

As it was mentioned in the introduction, in this paper we tackle the problem of achieving robust or stable portfolios using MOEAs. In order to do that, we suggest replacing the traditional fitness function with a new one that extends it with a resampling mechanism and an implicit third objective to control the robustness of the solutions in the front.

We will test the effectiveness of the approach on different MOEAs using the same chromosome structure and fitness evaluation procedure. In this section, we introduce briefly the MOEAs tested and provide details regarding the chromosome encoding and fitness function.

3.1. Tested Evolutionary Multi-objective Algorithms

The following MOEAs will be tested: NSGA-II, GDE3, SMPSO, SPEA-II. All the algorithms are implemented in jMetal [7], a Java framework aimed at multiobjective optimization with metaheuristics. By reusing the base classes of jMetal, all the techniques share the same basic core components (solution encodings, operators, etc.). This ensures a fair comparison of the considered algorithms.

NSGA-II, proposed by Deb et al. [5], is one of the most widely used multiobjective metaheuristics. It represents the new version of the NSGA algorithm developed by the same author. It is a generational genetic algorithm based on an auxiliary population derived from the original one applying the usual genetic operators (selection, crossover and mutation). Then, the two populations are merged and the individuals are sorted according to their rank. Inside each of these ranks, the crowding distance is used to sort the individuals from less to more crowded. A solution with a smaller value of this distance measure is, in some sense, more crowded by other solutions. Finally, the best solutions are selected to compose the new population that will be used to create a new population.

GDE3 [11] is the third version of the Generalized Differential Evolution algorithm (GDE), an extension of Differential Evolution (DE) for global optimization with an arbitrary number of objectives and constraints. GDE3 starts with a population of random solutions. At each generation, an offspring population is created using the differential evolution operators; then, the current population for the next generation is updated using the solutions of both, the

offspring and the current population. Before creating the next generation, the size of the population is reduced using non-dominated sorting and a pruning technique aimed at diversity preservation, in a similar way as NSGA-II. However, GDE3 modifies the crowding distance of NSGA-II in order to solve some of its drawbacks when dealing with problems having more than two objectives.

SMPSO, introduced by Nebro et al. [13], is a multiobjective particle swarm optimization algorithm (PSO) characterized by the use of a strategy to limit the velocity of the particles. It is based in OMOPSO [17] but including the velocity constriction procedure. This mechanism is useful when the velocity becomes too large because it can produce new effective particles positions. SMPSO also relies on an external archive that stores the non-dominated solutions found during the search process. Polynomial mutation is used in the algorithm as a factor of turbulence.

SPEA-II algorithm [24], developed by Zitzler et al., solves some weaknesses of a previous version by the same authors called SPEA. Among the improvements, we could mention a fitness functions that takes into account, for each individual, the number of individuals dominated by this one and the number of individuals which dominate it. This version also adds a density estimation for the population. This algorithm uses a core population and an archive. It assigns to each individual a fitness value that is the sum of its strength raw fitness plus a density estimation. In each generation, the non-dominated individuals of both the original population and the archive are used to update the archive; if the number of non-dominated individuals is greater than the population size, a truncation operator based on calculating the distances to the k -th nearest neighbor is used. All this procedure is known as Environmental Selection. Then, the algorithm applies the selection, crossover, and mutation operators to members of the archive in order to create a new population of offsprings which becomes the population for the next generation.

3.2. Solution encoding

The encoding chosen will represent each portfolio as a vector of real numbers. Each of these numbers represents the percentage of investment per asset (also called weight: w_i where i ranges from $1 \dots n$, and n is the number of investable assets). Here, each portfolio will be represented by a single element of the population.

Every individual must meet the constraints specified by eqs. 3, 4 explained before. The sum of weights per individual must be 1, that is full investment is required. Also, the individuals must satisfy additional real-world constraints showed in eqs. 5 and 6. The satisfaction of these constraints is guaranteed by repairing algorithms. The individuals are repaired both after initializing the population (see alg. 1), and applying the genetic operators (see alg. 2). The repairing algorithms transform non-complying chromosomes into portfolio satisfying constraints. Whenever the number of invested assets do not belong to the interval $[C_{min}, C_{max}]$, its number is adjusted to ensure compliance with the cardinality constraint. This is done adding or dropping assets until the requirement is met. In case the sum of weights per individual is not 1.0, the algorithm fine tunes the holdings adding or subtracting random amounts up to the required adjustment. These changes are forced meet the investment limits $[lim_{inf}, lim_{sup}]$. The details of this process are described in algorithms 1 and 2.

Algorithm 1 Reparation after initialization

```

Initialize population  $P$  as a set of vectors with real numbers  $\bar{x}_i = (x_{i1}, \dots, x_{in})$ ;  $x_{ij} \in [lim_{inf}, lim_{sup}]$ 
for each individual  $\bar{x}_i$  of  $P$  do
  A random number  $\in [C_{min}, C_{max}]$  of values 0 is assigned to coordinates of vector  $\bar{x}_i$ 
  while  $\sum_{j=1}^n x_{ij} \neq 1$  do
    if sum of the  $lim_{inf}$  of  $x_{ij} \neq 0$  is  $> 1$  then
      select randomly one coordinate  $j$  of vector  $\bar{x}_i$  such that  $x_{ij} \neq 0$  and assign  $x_{ij} = 0$ 
    end if
    if sum of the  $lim_{sup}$  of  $x_{ij} \neq 0$  is  $< 1$  then
      select randomly one coordinate  $j$  of vector  $\bar{x}_i$  such that  $x_{ij} = 0$  and assign  $x_{ij} = lim_{sup}$ 
    end if
    if  $\sum_{j=1}^n x_{ij} \neq 1$  then
      select randomly one coordinate  $j$  of vector  $\bar{x}_i$  such that  $x_{ij} \neq 0$ 
      add/subtract the quantity left to make the  $\sum_{j=1}^n x_{ij} = 1$  respecting the limits  $[lim_{inf}, lim_{sup}]$ 
    end if
  end while
end for
Return( $P$ )

```

Algorithm 2 Reparation after genetic operators

```

for each individual  $\bar{x}_i$  of  $P$  do
  if there are less  $x_{ij} = 0$  than  $C_{min}$  then
    set one random  $x_{ij} = 0$  to  $x_{ij} = lim_{inf}$ 
  end if
  if there are more  $x_{ij} = 0$  than  $C_{max}$  then
    set the  $x_{ij} \neq 0$  with less value to  $x_{ij} = 0$ 
  end if
  while  $\sum_{j=1}^n x_{ij} \neq 1$  do
    if sum of the  $lim_{inf}$  of  $x_{ij} \neq 0$  is  $> 1$  then
      select randomly one coordinate  $j$  of vector  $\bar{x}_i$  such that  $x_{ij} \neq 0$  and assign  $x_{ij} = 0$ 
    end if
    if sum of the  $lim_{sup}$  of  $x_{ij} \neq 0$  is  $< 1$  then
      select randomly one coordinate  $j$  of vector  $\bar{x}_i$  such that  $x_{ij} = 0$  and assign  $x_{ij} = lim_{sup}$ 
    end if
    if  $\sum_{j=1}^n x_{ij} \neq 1$  then
      select one random  $x_{ij} \neq 0 \in [lim_{inf}, lim_{sup}]$ 
      add/subtract one random quantity left to make the  $\sum_{j=1}^n x_{ij} = 1$  respecting the limits  $[lim_{inf}, lim_{sup}]$ 
    end if
  end while
end for
Return( $P$ )

```

3.3. Fitness function. Time-stamped resampling

In this paper we introduce a resampling strategy combined with an implicit age-based third objective to identify robust portfolios. In this subsection, the procedure to evaluate the fitness function is described in detail.

The starting point for the evaluation of portfolios is the framework introduced in section 2, where part of the fitness of each individual is determined by evaluating two objective functions: the return $E(R_p)$ (eq. 2) to be maximized and the risk σ_p (eq. 1) to be minimized. As it is apparent, $E(R_p)$ and σ_p are very tightly related to the values of the expected asset returns and the variance-covariance matrix. One of the most important challenges that portfolio managers face when they operate within Markowitz's framework is the dependence of the solutions on the estimates for these parameters. Given the challenges inherent to financial forecasting, it is normal that the mentioned parameters are not accurate.

The mentioned difficulty is likely to result in a set of portfolios that could end up behaving in an unexpected way. Fig.2 shows a real example of one Pareto front where the solutions are evaluated using the forecast parameters (in red) vs. the real parameters (in green). The set of portfolios that were optimized for the traditionally considered most likely scenario (mean return for each asset over a period of time, and variance-covariance matrix computed using the same data) define the upper efficient frontier. However, once we calculate the risk and return for the same portfolios using the real observed parameters (actual assets returns instead of the mentioned averages), we realize that their profiles can be very different. As can see, the difference between the two fronts, that represent the same solution, might be potentially quite dramatic.

A first step towards robust solutions would be resampling during the evolution process. This approach would require replacing the mentioned parameters in the fitness functions at every generation. Doing this, the evolution process would favor the individuals that show good performance in terms of risk and return over different scenarios, and would discard those overspecialized on specific values for the parameters, including the predicted ones. A key element of this process is the creation of these scenarios. We would need a set of solutions able to handle uncertainty, but there is no need for absolute generalization. The scenarios should be considered according to their likelihood.

The approach that we use to generate likely scenarios is based on nonparametric bootstrap. As we mentioned before, the usual way to forecast the value of parameters for the model is averaging the returns and computing the associated variance-covariance matrix over a number of periods. Our algorithm has the same starting point, the definition of a time window. However, instead of using all the data to derive a single estimate for the parameters, data are resampled. The process selects a random set of time periods that has the same size as the original window (each period might be selected more than once). Then, we average the returns for those time periods and compute the variance-covariance matrix. Every time we do this, we generate new estimates for the parameters that are based on

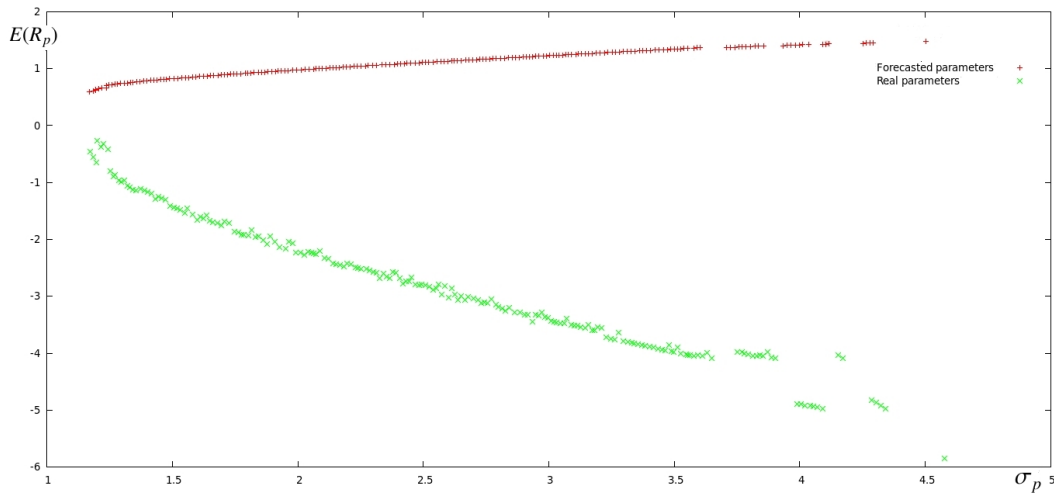


Figure 2. Solution evaluated with forecast and real parameters

past data. These estimates can subsequently be used to calculate the risk and returns of the portfolios. This process is described in 3.

Algorithm 3 Resampling method

S is the original sample set with a size N_s .
 S' is the new sample set with a size N'_s . At the beginning, $S' = \emptyset$ and $N'_s = 0$.
while $N'_s \neq N_s$ **do**
 It selects one instance X_i , at random, from S .
 This instance is added to the new set. $S' = S' + X_i$.
end while
Return(S')

Every generation, the algorithm evaluates all the individuals under a new resampled scenario. This is relevant because using a shared set of parameters makes the comparison for dominance purposes meaningful.

This process alone would be enough to weed out overspecialized solutions over time. However, as it was described, it has a relevant flaw. The results would be very dependant on the last iterations of the evolution process. Every generation, we replace some solutions from the previous population with new ones and, chances the lasts scenarios generated might be very extreme. If that were the case, a percentage of the best solutions that might have fared well over time, might offer a poor performance. This could result on new solutions specialized on the last scenario replacing the old ones. If this happened during the first few generations, the specialized solutions would be replaced over time. However, in case we faced this in the final stages, we might end up obtaining poor results. We will prevent this adding to the described resampling an implicit third objective based on a time-stamping mechanism.

This feature extends the resampled mean-variance model with a third objective, the age of the solution, to be maximized. Each individual will have a counter that will keep track of the number of generations it has remained in the population. The explicit maximization of this variable rewards good performance in previous generations and mitigates the risk described before. Once the evolution process is over, the population is evaluated once more using the forecast values for the parameters for the period. The final set of portfolios, provided as solutions, consists of the elements of the population that are not dominated in terms of risk and return. At this stage, all the information concerning the third objective is disregarded. Algorithm 4 shows the modifications carried out in the basic MOEAs to implement the fitness function described in this section.

We expect that the sum of these changes in the traditional fitness function would discard those portfolios that, under normal circumstances, could potentially result in particularly bad performance, and prioritize those that offer consistently good solutions under likely scenarios.

Algorithm 4 Modifications on the basic MOEAs

```

(Specific steps of MOEA)
Reparation after initialization (alg. 1)
for each generation do
  A new sample set ( $S'$ ) is generated by alg. 3
  (Specific steps of MOEA)
  Reparation after genetic operators (alg. 2)
  for every existing individual  $p$  do
    Calculate  $E(R_p)$  on  $S'$ 
    Calculate  $\sigma_p$  on  $S'$ 
     $Tp = Tp + 1$ 
  end for
  (Specific steps of MOEA)
end for
(Specific steps of MOEA)
MOEA returns a non-dominated solution's set  $P$ 
Compute forecast parameters ( $S$ : original sample set)
for each individual  $p$  in  $P$  do
  Discard time-stamping objective  $Tp$ 
  Calculate  $E(R_p)$  on  $S$ 
  Calculate  $\sigma_p$  on  $S$ 
end for
Get non-dominated individuals ( $P_f$ ) from  $P$ 
Return ( $P_f$ )

```

4. Evaluation Metrics

Solutions resulting from different runs of multiobjective algorithms must be compared using quantitative metrics. There is wide range of alternatives, some of them dependent on knowing the Pareto-optimal front. It is generally admitted that there is no single perfect metric that can be used to compare solutions in every circumstance. Actually, the choice of the right metric among the wide range available is itself, a multiobjective problem.

The metrics most widely used in multiobjective context are Hypervolume and Spread [25]. These metrics could be useful when we look for good distributed and strong dominant fronts. However, they are not appropriate to capture the effect showed in fig. 2. In order to measure the robustness of the solutions, we introduce four metrics that evaluate different aspects related portfolio robustness. They are named: Estimation Error, Stability, Extreme Risk and Unrealized Returns. Next, we describe them in detail.

- **Estimation Error:** It evaluates the average difference between the expected risk and return for every portfolio in the efficient frontier and the actual risk and return a posteriori once the real values of the parameters are observed. That is, the mean distance between the estimates for t_n based on data from t_0 to t_{n-1} , and the actual values at t_n . For this purposes, we consider the Mahalanobis distance, $d_M(\bar{x}, \bar{y})$, which is defined as,

$$d_M(\bar{x}, \bar{y}) = \sqrt{(\bar{x} - \bar{y})^T \Sigma^{-1} (\bar{x} - \bar{y})} \quad (7)$$

where \bar{x} and \bar{y} are the patterns to be compared, and Σ is the variance-covariance matrix.

This metric is calculated measuring the average distance (d_M) between \bar{x}_p and \bar{x}_p' for all the portfolios in the solution. Here, \bar{x}_p represents, on one hand, the pair $(E(R_{np}), \sigma_{np}^2)$ for portfolio p and period t_n calculated using forecast parameters. On the other hand \bar{x}_p' is defined by the pair $(E(R_{np})', \sigma_{np}^2')$ where both return and risk for the same portfolio and moment in time are computed using the the real parameters (the ones observed for t_n). Formally, it can be expressed as follows:

$$EE = \frac{\sum_{p=1}^N [d_M(\bar{x}_p, \bar{x}_p')]^2}{N} \quad (8)$$

where N is the number of portfolios in the Pareto front. The smaller is the difference between the forecast and reality, the lower is the value of this metric and, therefore, the higher is the reliability of the original front.

- **Stability:** The Stability of a portfolio is measured averaging the Mahalanobis distances among the expected pair of risk/return in the efficient frontier and the expected risk/return in S different scenarios. Unlike the Estimation Error, which considers the difference between the expected scenario and the real scenario, this metric measures the average difference between the expected scenario and a wide range of feasible alternatives. The concern is the aggregate sensitivity to the distribution of potential scenarios, not to the one that happened to materialize. The metric is given by the equation:

$$ST = \frac{\sum_{i=1}^S \frac{\sum_{p=1}^N [d_M(\bar{x}_p, \bar{x}_{pi})]^2}{N}}{S} \quad (9)$$

where $d_M(\bar{x}_p, \bar{x}_{pi})$ is the Mahalanobis distance between $(E(R_{np}), \sigma_{np}^2)$ and $(E(R_{npi}), \sigma_{npi}^2)$, the return and risk for portfolio p and period t_n calculated using parameters for the scenario i .

The set of scenarios is generated using nonparametric bootstrap technique described in Alg. 3. The relevance of this metric would subject to the size of the data set used to resample and the value of S . A larger set of scenarios is likely to result on a more accurate approximation to the potential distribution of parameters. As it was mentioned before the value of the metric is obtained averaging all the average distances. Therefore, high values of this metric would represent higher sensitivity to likely scenarios and lower reliability.

- **Extreme Risk:** This metric evaluates the sensitivity of the solutions to worst-case scenarios. It is closely related to Stability, and it matches the same basic definition. The difference is that, instead of considering the average for all S scenarios, we will only take into account a small subset. Specifically, the computation would include only the w worst-case ones. For this purposes we define worse-case scenarios as the parameter combinations that result in the highest average Mahalanobis distance between the expected risk and returns, and the risk and returns for the same portfolios using the resampled parameters. The rationale for this indicator is providing an estimate for the expected average negative outcome of the realization of the worst scenario with probability w/S . The higher the metric, the higher the risk.
- **Unrealized Returns:** This indicator provides information on the average potential return left on the table. Namely, it measures the average squared difference between the realized return and the maximum potential return for that risk level for all the portfolios in the solution. This is defined as:

$$UR = \frac{\sum_{p=1}^N [R_{npe} - R_{np}]^2}{N} \quad (10)$$

where N is the number of portfolios in the solution, R_{np} is the return of portfolio p computed using the ex-post parameters for time t_n and R_{npe} is the return of the portfolio, pe , with the most similar risk level in the efficient frontier derived using the ex-post parameters. High values on this metric would indicate large unrealized potential returns.

5. Experimental Validation

In this section we test the aforementioned approach on a specific asset allocation problem. We try to identify the mixes of broad investment categories that provide the optimal balance between expected risk and return, while increasing robustness. As it was mentioned in the introduction, four different multiobjective meta-heuristics are considered on a constrained version of the problem. For each of them, NSGA-II, SPEAII, SMPSO and GDE3, we compare the performance of the standard version versus the robust equivalent.

For experimental purposes, the cardinality constraints $[C_{min}, C_{max}]$ and limits to the minimum and maximum weight that each asset can carry in the portfolio, $[lim_{inf}, lim_{sup}]$, are set to $[2,6]$ and $[0.1,0.8]$ respectively.

Subsection 5.1 describes the data sets that we used in this work. Subsection 5.2 gives details about the parameter settings for the experiments, multiobjective algorithms, and metrics. Finally, subsection 5.3 presents and analyzes the experimental results.

5.1. Data Sets

For experimental analysis, we use a sample that consists of 240 monthly returns for eight broad financial indexes representing that many asset classes. The series of monthly returns cover the time period from January, 1990 to December, 2009 and the source for the data is *Datastream*. The list of indexes is provided in Table 1.

Name	Code
Frank Russell 2000 Value	FRUS2VA
Frank Russell 2000 Growth	FRUS2GR
Frank Russell 1000 Value	FRUS1VA
Frank Russell 1000 Growth	FRUS1GR
S&P GSCI Commodity Total Return	GSCITOT
MSCI EAFE	MSEAFEL
BOFA ML CORP MSTR (\$)	MLCORPM
BOFA ML US TRSY /AGCY MSTRAAA(\$)	MLUSALM

5.2. Experimental Parameters

The experimentation is based on a sliding window approach. Each window has a size n of 120 sets of returns for the eight indexes, that is 10-year worth of data. This means that the algorithm will rely on data from t_0 to t_{n-1} to identify the best possible allocations for the period t_n . The 10-year window will move one month at a time, 120 times. This will cover the date interval from 31/01/1990 to 31/12/2009.

The algorithms will be run 20 times per window using the parameters described in table 2. This means that, for each window, we will obtain 20 solutions sets per algorithm.

Table 2. Parameters. $L = 8$ (individual length). The termination condition is to compute 300 iterations.

SPEA2	
Population size	200 individuals
Archive size	200 individuals
Crossover	SBX, $p_c = 0.9$
Mutation	Polynomial, $p_m = 1/L$
Selection of Parents	Binary tournament
NSGA-II	
Population size	200 individuals
Crossover	SBX, $p_c = 0.9$
Mutation	Polynomial, $p_m = 1/L$
Selection of Parents	Binary tournament
SMPSO	
Archive size	200 particles
Swarm size	200 particles
Mutation	Polynomial, $p_m = 1/L$
GDE3	
Population size	200 individuals
Crossover	DE crossover, $CR = 0.9$
Mutation	DE mutation, $F = 0.5$
Selection of Parents	DE selection

As for the parameters required by the metrics, the number of scenarios S necessary to compute Stability and Extreme Risk, will be set to 500. The number of worst-case scenarios, w , considered for the second, will be 5. That means that comparison will be based on the 1% worst expected average outcomes.

5.3. Experimental Results

This section shows the results from the experimental process described before. For every multiobjective algorithm tested, we compare the performance of the standard and the robust version of the algorithms over 20 runs using the metrics introduced in 4. In the tables, the first one is labeled as "name-of-algorithm" and the second one as "name-of-algorithm R+T" (R+T means resampling and third objective). Results are provided in terms of descriptive statistics

(average, median and variance) for metrics described in section 4. The percentage of improvement of the robust version over the basic one, labeled as % Av., is also reported.

Table 3 shows the set of results for Estimation Error metric (EE). The advantage of using the robust approach over the standard one ranges from 27.46% to 54.93%. The lowest prediction errors are achieved using the robust version SMPSO, however the highest improvement takes place for SPEA2 which is the algorithm with highest initial EE value. We also note that the worse EE in the basic MOEA, the more it is enhanced by the robust approach.

The results for the Stability metric are reported in table 4. The table shows how the proposed approach (R+T) improves significantly the stability of portfolios achieved by the standard MOEAs. Once again, the improvements are specially remarkable for the algorithms that, in their standard formulation, lead to less stable fronts. We observe that SPEA2 is the algorithm that provided the highest values for the metric in the basic version and the lowest values in its robust alternative. This makes the difference between using R+T mechanism and the standard algorithm specially large (67.53%). Conversely, GDE3 R+T shows the lowest improvement %Av (35.61%) and the worst final value for the metric, once we use the robust approach.

Table 3. "Estimation Error"

	Average	Median	Variance	% Av.
SPEA2	2,5198	1,7945	5,1421	
SPEA2 R+T	1,1356	0,5538	2,4147	-0,5493
NSGAI	2,2823	1,7996	3,9817	
NSGAI R+T	1,2401	0,6609	2,3748	-0,4566
SMPSO	1,5204	1,2878	1,7354	
SMPSO R+T	1,0144	0,7403	0,9302	-0,3328
GDE3	1,4820	1,2580	1,5347	
GDE3 R+T	1,0750	0,6721	1,2762	-0,2746

Table 4. "Stability"

	Average	Median	Variance	% Av.
SPEA2	7,4057	7,0743	20,7329	
SPEA2 R+T	2,4044	2,1024	2,2526	-0,6753
NSGAI	6,5508	6,1640	12,9661	
NSGAI R+T	2,7433	2,5085	2,1520	-0,5812
SMPSO	5,2373	4,8981	7,2544	
SMPSO R+T	3,2510	2,8582	3,6025	-0,3793
GDE3	5,1666	4,8507	6,8238	
GDE3 R+T	3,3270	2,9755	3,8045	-0,3561

The results for the Unrealized Returns metric are reported in table 5. We observe that the solutions obtained by R+T using observed parameters are closer to the optimal efficient frontier than the solutions derived from the standard algorithms. The best results, once again, were obtained by SMPSO R+T. Results show improvements that range from 19.56% to 32.38%. GDE3 is the algorithm with the lowest % Av. despite not being the algorithm with the lowest value for the metric in the standard version.

Finally, solutions evaluated under the less predictable parameters or scenarios (see table 6) show the same pattern of behavior already observed for EE and Stability metrics. It is remarkable that solutions obtained by R+T approach and evaluated under these worst scenarios are significantly closer to the observed parameters than the solutions of standard MOEAs evaluated under the forecast parameters. In this case, even though SPEA2 R+T offers the best median result, on average, it is beaten by the R+T versions of GDE3 and SMPSO, which provides both the smallest average value and the lowest variance for the metric.

Table 5. "Unrealized Returns"

	Average	Median	Variance	% Av.
SPEA2	3,5848	2,7989	8,5604	
SPEA2 R+T	2,4240	1,8792	3,9853	-0,3238
NSGAI	3,4871	2,7326	7,8823	
NSGAI R+T	2,5670	1,9975	4,1648	-0,2638
SMPSO	2,9805	2,4745	4,9961	
SMPSO R+T	2,2514	1,7717	2,9235	-0,2446
GDE3	3,3518	2,9221	5,7049	
GDE3 R+T	2,6962	2,1723	4,5393	-0,1956

Table 6. "Extreme Risk"

	Average	Median	Variance	% Av.
SPEA2	3,5308	3,0614	4,9670	
SPEA2 R+T	1,6622	1,0862	2,5638	-0,5292
NSGAI	3,2407	2,8451	3,9687	
NSGAI R+T	1,8172	1,2318	2,5525	-0,4393
SMPSO	2,2485	2,0297	1,8884	
SMPSO R+T	1,4671	1,2126	1,0257	-0,3475
GDE3	2,2091	1,9701	1,7397	
GDE3 R+T	1,6425	1,2577	1,5775	-0,2565

The differences between the average values for the metrics for the basic and the suggested R+T versions of the algorithm, % Av., were tested for statistical significance. We used the protocol described in algorithm 5 and all the differences were significant at 1%.

The best performing algorithm, in terms of robustness, seems to be SMPSO R+T. It offers the lowest values for three out of the four metrics considered (Estimation Error, Unrealized Returns and Extreme Risk). The best performing one in terms of Stability is SPEA2 R+T. Among the standard MOEAS, GDE3 provides the highest robustness

Algorithm 5 Statical testing protocol

```

if the values follow a Normal distribution (applying the Kolmogorov Smirnov test) then
  if the variances are homogeneity (the Levene test is used to check it) then
    A t-test is performed.
  else
    A Welch test is executed.
  end if
else
  A Wilcoxon test is applied to compare the medians of the solutions
end if

```

followed by SMPSO.

As we have have seen, the proposed R+T mechanism resulted in a systematic and significant increase in the reliability of solutions across the algorithms. The introduced R+T mechanism generates more robust solutions than any of the basic approaches of the MOEAs. The minimum decrease in average metric value was 19.56% evaluating Unrealized Returns for GDE3. The largest improvement, 67.53%, was achieved SPEA2 for Stability. Furthermore, we also observed that our approach helped to improve those algorithms which produced more unstable solutions.

6. Conclusions

Portfolio optimization represents one of the main topics in financial research. The basic framework targets the combination of financial assets to be included in a portfolio in order to optimize the balance of risk and return. This process is usually based on two parameters: the estimates for the expected asset returns, and the variance-covariance matrix. Quadratic Programming can be used to solve the basic version of this problem. However, in the real world there are some constraints that cannot be easily tackled by traditional techniques. This is the reason why evolutionary multiobjective algorithms (MOEAs), that do not present these kind of limitations, are getting traction in the domain.

One of the main problems that portfolio managers face is the uncertainty regarding the expected frontier derived from their forecasts for future returns. Very often, it lies far from the actual one, resulting in inaccurate forecast risk/return profiles for the portfolios. Our work is focused on this point, adding robustness to the results by preventing optimizing for a single expected scenario that may produce solutions that are hyper-specialized and might be extremely sensitive to likely deviations. We handle the uncertainty in these parameters during the optimization process testing the population for different values of parameters and selecting the portfolios that consistently offer a good performance. We also use a time-stamping mechanism that, in collaboration with performance in terms of risk and return, drives the evolution process to find robust and stable solutions.

The assessment of the robustness of the efficient frontiers resulting from the process requires metrics that differ from the most traditional ones. For this reason, we introduce a set of four metrics "Estimation Error", "Stability", "Extreme Risk" and "Unrealized Returns".

The approach was tested through experimentation using four popular MOEAs (NSGA-II, SPEA2, SMPSO and GDE3). The results were compared in both the standard and the time-stamped resampled versions of the algorithms. They were tested on a sample of monthly returns for eight indexes representing different broad investment categories including stock, bonds, etc. The results show that the suggested approach enhances significantly the reliability of solutions for all algorithms. The time-stamped resampled versions improved the results across all the considered MOEAs. The improvements got up to 67.53% in "Stability", 54.93% in "Estimation Error", 52.92% in "Extreme Risk" and 33.32% in "Unrealized Returns". These improvements were higher when standard MOEA provided worse results in terms of the metrics. We also observed that the R+T version of SMPSO tended to outperform the alternatives.

Even though these results are good and promising in terms of robustness, there are several issues left open that could lead to future extensions of this work. Among them, the analysis of the performance of this robust approach on other MOEAs or the scalability of the results with different number of investment alternatives.

7. Acknowledgements

The authors acknowledge financial support granted by the Spanish Ministry of Science under contract TIN2008-06491-C04-03 (MSTAR), TIN2011-28336(MOVES) and Comunidad de Madrid (CCG10-UC3M/TIC-5029).

References

- [1] Anagnostopoulos, K., Mamanis, G., 2011. The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications* 38 (11), 14208–14217.
- [2] Barbosa, H. J., Lemonge, A. C., 2005. A genetic algorithm encoding for a class of cardinality constraints. In: *Proceedings of the 2005 conference on Genetic and evolutionary computation. GECCO '05*. ACM, New York, NY, USA, pp. 1193–1200.
- [3] Chang, T.-J., Yang, S.-C., Chang, K.-J., 2009. Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications* 36 (7), 10529 – 10537.
- [4] Chiranjeevi, C., Sastry, V. N., 2007. Multi objective portfolio optimization models and its solution using genetic algorithms. *Computational Intelligence and Multimedia Applications, International Conference on* 1, 453–457.
- [5] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2), 182–197.
- [6] Deb, K., Steuer, R. E., Tewari, R., Tewari, R., 2011. Bi-objective portfolio optimization using a customized hybrid NSGA-II procedure. In: *EMO*. pp. 358–373.
- [7] Durillo, J., Nebro, A., Alba, E., July 2010. The jmetal framework for multi-objective optimization: Design and architecture. In: *CEC 2010*. Vol. 5467 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Barcelona, Spain, pp. 4138–4325.
- [8] Garcia, S., Quintana, D., Galvan, I. M., Isasi, P., 2011. Portfolio optimization using SPEA2 with resampling. In: *Proceedings of the 12th international conference on Intelligent data engineering and automated learning. IDEAL11*. Springer-Verlag, Berlin, Heidelberg, pp. 127–134.
- [9] H.M., M., 1959. *Portfolio Selection: efficient diversification of investments*. John Wiley & Son.
- [10] Idzorek, T. M., 2006. *Developing robust asset allocations*. Tech. rep.
- [11] Kukkonen, S., Lampinen, J., 2005. GDE3: The third evolution step of generalized differential evolution. In: *IEEE Congress on Evolutionary Computation (CEC'2005)*. pp. 443 – 450.
- [12] Markowitz, H., 1952. Portfolio selection. *The Journal of Finance* 7 (1), 77–91.
- [13] Nebro, A., Durillo, J., García-Nieto, J., Coello Coello, C., Luna, F., Alba, E., 2009. SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In: *2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009)*. IEEE Press, pp. 66–73.
- [14] Perret-Gentil, C., Victoria-Feser, M.-P., Apr. 2005. Robust mean-variance portfolio selection. *Fame research paper series, International Center for Financial Asset Management and Engineering*.
- [15] Pflug, G., Wozabal, D., 2007. Ambiguity in portfolio selection. *Quantitative Finance* 7 (4), 435–442.
- [16] Radziukyniene, I., Xilinskas, A., 2008. Evolutionary methods for multi-objective portfolio optimization. In: *Ao, S. I., Gelman, L., Hukins, D. W., Hunter, A., Korsunsky, A. M. (Eds.), Proceedings of the World Congress on Engineering 2008 Vol II, WCE '08, July 2 - 4, 2008, London, U.K. Lecture Notes in Engineering and Computer Science. International Association of Engineers, Newswood Limited*, pp. 1155–1159.
- [17] Reyes, M., Coello, C. C., 2005. Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance. In: *Coello, C., Hernández, A., Zitzler, E. (Eds.), Third International Conference on Evolutionary MultiCriterion Optimization, EMO 2005. Vol. 3410 of LNCS. Springer*, pp. 509–519.
- [18] Ruppert, D., June 2006. Statistics and finance: An introduction. *Journal of the American Statistical Association* 101, 849–850.
- [19] Shiraishi, H., 2008. Resampling procedure to construct value at risk efficient portfolios for stationary returns of assets. Tech. rep.
- [20] Skolpadungket, P., Dahal, K., Harnpornchai, N., 2007. Portfolio optimization using multi-objective genetic algorithms. In: *Proceeding of 2007 IEEE Congress on Evolutionary Computation*. pp. 516–523.
- [21] Soleimani, H., Golmakani, H. R., Salimi, M. H., April 2009. Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Syst. Appl.* 36, 5058–5063.
- [22] Tütüncü, R. H., Koenig, M., 2004. Robust asset allocation. *Annals OR* 132 (1-4), 157–187.
- [23] Zhu, H., Wang, Y., Wang, K., Chen, Y., 2011. Particle swarm optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications* 38 (8), 10161 – 10169.
- [24] Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the strength pareto evolutionary algorithm. Tech. Rep. 103, *Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland*.
- [25] Zitzler, E., Thiele, L., 1998. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Tech. Rep. 43, *ss, Gloriastrasse 35, CH-8092 Zurich, Switzerland*.
URL citeseer.ist.psu.edu/article/zitzler99evolutionary.html