

# Multiobjective Robustness for Portfolio Optimization in Volatile Environments

Ghada Hassan  
Department of Computer Science  
University College London  
Gower Street, London WC1E 6BT  
g.hassan@cs.ucl.ac.uk

Christopher Clack  
Director of Financial Computing  
Department of Computer Science  
University College London  
Gower Street, London WC1E 6BT  
c.clack@cs.ucl.ac.uk

## ABSTRACT

Multiobjective methods are ideal for evolving a set of portfolio optimisation solutions that span a range from high-return/high-risk to low-return/low-risk, and an investor can choose her preferred point on the risk-return frontier. However, there are no guarantees that a low-risk solution will remain low-risk — if the environment changes, the relative positions of previously identified solutions may alter. A low-risk solution may become high-risk and vice versa.

The robustness of a Multiobjective Genetic Programming (MOGP) algorithm such as SPEA2 is vitally important in the context of the real-world problem of portfolio optimisation. We explore robustness in this context, providing new definitions and a statistical measure to quantify the robustness of solutions.

A new robustness measure is incorporated into a MOGP fitness function to bias evolution towards more robust solutions. This new system (“R-SPEA2”) is compared against the original SPEA2 and we present our results.

## Categories and Subject Descriptors

I.2.M [Artificial Intelligence]: Miscellaneous

## General Terms

Algorithms, Experimentation

## Keywords

GP, Multiobjective Optimization, Robustness, Portfolio Optimization, Finance, Dynamic Environment

## 1. INTRODUCTION

At the start of the Summer of 2007, the most attractive area of the financial markets was structured credit products — which coupled the advantage of high returns with a controllable level of risk. Credit risk could be divided among

different tranches and risk exposure would be controlled by choosing the appropriate tranche. These structured products therefore occupied a range of attractive positions on the efficient frontier [20] of risk versus reward, and leading investment banks allocated significant funds to them. All was well until the markets changed, US homeowners defaulted on their mortgages, and mortgage-backed structured products (in particular) failed in a spectacular fashion. Merrill Lynch, for example, lost over \$23,000,000,000 as well as their Head of FICC and their CEO. This was unfortunate.

A key problem with these structured products is that they are not sufficiently robust to changes in the market. Their risk-reward profiles change significantly when the market changes (e.g. when short-term interest rates increase).

Multiobjective (MO) optimization algorithms [3, 5], attempt to solve problems that have multiple, possibly conflicting, objectives. In such problems, a single solution that can optimize all the objectives simultaneously often does not exist. Rather, the search produces a set of “trade-offs” between different objectives, known as the Pareto front. Almost all algorithms developed to evolve the Pareto front share the following main ideas:

- A fitness assignment that uses the *Pareto Dominance* concept to measure the quality of the solution;
- An external population, where an archiving technique is used to maintain the solutions found on the front of each generation;
- A density estimation that measures proximity of solutions, and helps the evolution of a well distributed and well spread front by biasing against crowding of solutions.

Research in MO algorithms has been active since the 1980s. They have been applied in a variety of mathematical and real life problems. Examples of applying MO Evolutionary algorithms in financial applications include: risk management [24], bank loan management [22], and management of financial portfolios [1, 16, 17, 25]. For a review of applications of MO Evolutionary algorithms in finance and economics see [27, 4].

Despite the broad range of research in MO algorithms, most of this work has focused on generating solutions on the Pareto front that are diverse and well distributed. Little attention has been paid to the *robustness* of solutions evolved, and they are rarely validated in unseen environments. In volatile environments such as the financial markets, robustness is of major importance. If robustness is not achieved,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '08, July 12–16, 2008, Atlanta, Georgia, USA.  
Copyright 2008 ACM 978-1-60558-130-9/08/07...\$5.00.

the solutions will exhibit unstable performance that may render them unfit in subsequent environments, and the algorithm is only useful as an analysis tool of historical data. Investors ideally prefer a solution whose specific risk and return never changes. Given that this is unlikely ever to be achieved in a volatile market, the next best solution is one that sustains the characteristics of its objectives. For example, given a solution with the lowest risk relative to the available alternative solutions, it should continue to give the lowest relative risk as the environment changes (even though the precise objective value of risk may change).

In this work, we look at the robustness problem through consideration of the performance of an evolved Pareto front in a new environment. We analyse what constitutes a robust solution in dynamic environments, and define a new metric for measuring robustness. The SPEA2 algorithm is altered to include the robustness measure in its fitness calculations. The results of the new algorithm (“R-SPEA2”) show improved robustness in comparison with the original.

The paper is organized as follows. Section two presents related approaches to robustness in a multi-objective context. Section three describes the portfolio optimization problem, and explains the system architecture used for experiments. Section four shows the results of applying SPEA2 to the portfolio optimization problem on out-of-sample data. In section five, we introduce and discuss the new robustness metrics. Experiments and results that compare the new algorithm to the standard algorithm are presented in Section six. Finally, Section seven draws conclusions.

## 2. RELATED WORK

Different definitions for robustness in the literature exist. The majority of research in this area [2, 6, 7, 12, 13] defines robustness of solutions as insensitivity to small perturbations in the decision variables, and adds a robustness measure to the fitness assessment. Other definitions of robustness include: consistency of results between different runs [26], and reliability of results in uncertain environments, where the fitness function optimum is time varying [8]. Jin and Mian [14, 18] improve the robustness of a single objective optimization problem by treating it as a multiobjective problem where robustness is an extra objective.

In [2], the concept of *degree of robustness* was introduced to measure the robustness of solutions against small variations in decision variables. The aim of the study was to determine the effect of the value of a threshold  $p$  in the determination of the Pareto front of robust solutions. The degree of robustness measure was embedded in the fitness assessment of solution to bias the search towards robust solutions. The new concept was tested on two mathematical functions used in [6], with the Pareto front plotted such that the solutions are distinguished by their corresponding degree of robustness with various  $p$  values. The new concept was considered an extra tool to aid the decision maker in her choice of a solution from the front. However, the robustness of the identified solutions was not actually tested in a new environment. Furthermore, the test problems were only 2 dimensional; since the perturbations in the decision variables can occur along any dimension, when the number of dimensions increases, computation of all possible combinations of perturbations in the hyper-cube that is the neighbourhood of a solution becomes very expensive.

The authors of [6, 7] came across the same problem when

they extended a definition of robust solutions used in single objective optimization to be suitable for multi objective optimization. The definition of the robust solution was one such that it is the global minimum of the *mean effective functions*, defined with respect to a predefined neighbourhood of size  $\delta$ . They generated 50 or 100 solutions in the neighbourhood, which effectively makes the method 50 or 100 times slower. Another result common with the previous research was always discovering that some areas of the Pareto front seem to exhibit concentration of robust solutions, and some areas have only a sparse number of robust solutions or none. Another recent work by Gasper and Covas [12] used a combination of two types of robustness measures; expectation and variance of fitness of a particular solution  $x$ . Expectation of the fitness is calculated as the weighted average of several points in the solution neighbourhood, and the variance assesses the deviation from the original fitness in the neighbourhood considered.

Robustness in the work of [26] was used to describe the standard deviation of a multiobjective algorithm between different runs each using a different random seed. Using this definition, algorithms based on the Pareto dominance relationship such as the NSGAI are quite robust. Other multi-objective algorithms based on the *Preferred* [10] relationship instead of dominance lead to poorly robust algorithms.

Our scope of work is on problems where, with the change of the environment, both the decision variables change, as well as the objectives’ ranges and the optimal solution. In such case, the whole fitness hyper-surface is shifted in the objective space, changing the range of objectives and possibly its shape, in response to the new environment. For a solution to be judged as robust, it should retain its optimality on the new fitness hyper surface. In single objective problems, measuring the robustness is dependent on how close the solution is to the real optimal value. In the case of multiple objectives, more needs to be achieved. First, it is necessary to examine a *set* of solutions; they are still required to be as close as possible to the optimal trade-off surface, but also they should maintain their objectives’ profile ranks as much as possible, so that a solution that was achieving high values on some objectives and low on some other, will keep the same high-low profile of objectives in the new environment. Second, we will also be interested in a front which retains its diversity and uniform distribution, so that all regions of the trade-off hyper-surface are well represented.

## 3. THE MULTI-OBJECTIVE PORTFOLIO OPTIMIZATION SYSTEM

In this section the portfolio optimization problem is described, followed by details of the system used in the experiments; including simulation of an investment fund with real life constraints and parameters, and the GP system used to evolve equations that determine the attractiveness of stocks, and hence their probability to be included in the portfolio.

### 3.1 Portfolio Optimization Problem

A portfolio is a collection of investments or assets held by an institution or a private individual. In this research, all the assets are assumed to be stocks. A portfolio provides diversification and therefore a degree of protection against the price volatility of underlying individual stocks. In our experiments, we simulate a sector-neutral portfolio with bal-

anced investment across several industries to guard against the price shocks of an individual sector. However, there is a limit to the protection that diversification can provide and we do not expect any investment model to be robust against extreme shocks to the entire market.

The general 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, such that the objectives sought by holding the portfolio are maximized. Markowitz [20] assumed that the objectives of the investor are maximizing the return on investment and minimizing the associated risk. Hence, solving the problem requires the simultaneous satisfaction of maximizing the expected portfolio return  $E$  and minimizing the portfolio variance  $V$  (the average squared deviation of the return from its expected mean value):

$$E = \sum_{i=1}^n x_i \mu_i \quad (1)$$

$$V = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{ij} \quad (2)$$

where  $n$  is the number of securities in portfolio,  $x_i$  is the relative amount invested in security  $i$ ,  $\sum_{i=1}^n x_i = 1$ ,  $\mu_i$  is the mean expected return of security  $i$ , and  $\sigma_{ij}$  is the covariance between assets  $i$  and  $j$ .

These equations are solved by a set of points that constitute the efficient frontier of the problem. The points define a curve similar to that of Figure 1 plotted in the risk return space of all possible portfolios. The points that constitute the curve represent portfolios for which there is the highest expected return given a certain amount of risk, or the minimum amount of risk given a certain expected return [20].

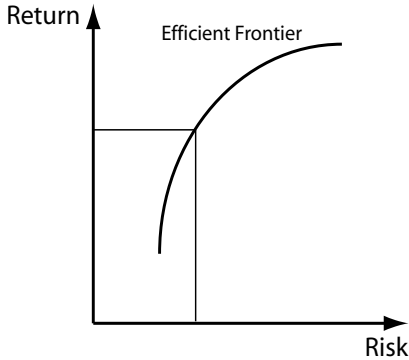


Figure 1: Efficient Frontier

### 3.2 System Architecture

Our system consists of a multiobjective GP, as well as an embedded simulation of an investment strategy, which is used for fitness assesment of solutions. A system schematic is given in Figure 2. The MOGP fitness function passes an individual (an equation representing a factor model) to the simulator; the equation is used to rank stocks on a monthly basis during simulation. The rankings drive buy/sell decisions and at the end of simulation various metrics (e.g. return, risk) are returned to the fitness function.

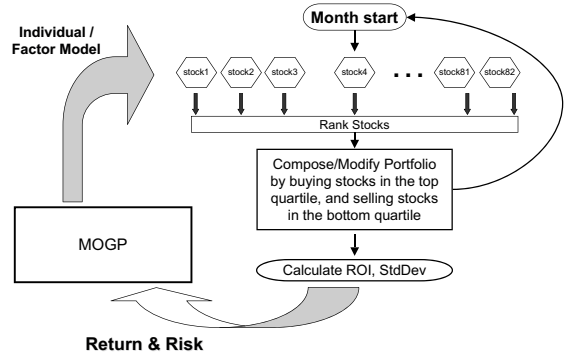


Figure 2: System Architecture

### 3.3 Investment Simulator

#### 3.3.1 Investment Strategy

The investment strategy employed is inspired by real world fund management practices. The portfolio held consists of one cash line and has a fixed cardinality of  $n = 25$  stocks. The initial portfolio value is  $C_o = \text{£}1,000,000$  in cash with no stock holdings. After that, the portfolio will constitute of  $n$  securities, and the current cash holding will be denoted by  $C$ , where we try to keep  $C$  less than or equal a maximum bound  $C_{max} = 3\%$  of the total fund value.  $S$  is the universe of equities,  $S_n$  is the set of securities held in the portfolio. For all buying and selling decisions in any day, it is assumed that we can trade at the opening price of that day. During the holding period, interest received on cash holdings is ignored.

For the duration of holding period, we do the following. At the start of each month, we calculate attractiveness of each stock in  $S$  according to the nonlinear factor model examined, and sort them accordingly. If any of the stocks we currently hold falls in the bottom quartile of the rank, it is sold. If the number of stocks currently in the portfolio is less than  $n$  or  $C > C_{max}$ , then we need to buy stocks from the top quartile, starting with the most attractive. The proportion to be invested in each stock is  $C_i$ , and is decided by:

$$C_i = \min\left(\frac{C}{n - |S_n|}, 4\% \text{ of total fund value}\right) \quad (3)$$

If we still have cash more than  $C_{max}$ , and there are some stock holdings with less than 4% of the total fund value, then we use all remaining cash to bring each of these stock holdings up to 4% or at least up to the maximum that the extra cash allows us to.

#### 3.3.2 Constraints and Additional Costs

In our problem formulation, we have included several realistic constraints and additional costs. The constraints are: portfolio cardinality, lower and upper bounds on investment per stock, maximum cash holding, and 2% transaction costs. With the addition of constraints, no analytic method exists for solving it (otherwise it can be solved in an exact manner by quadratic programming). Also, formulating the problem

with constraints may lead to a Pareto front that is non-convex or discontinuous making the problem harder to solve by a single objective evolutionary algorithm with a fitness function defined as the linear aggregation of objectives [27].

### 3.4 Multiobjective GP

The multiobjective algorithm used is SPEA2 [28]. The implementation (in Java) is based on the ECJ package [19]. Experiments have a population size of 1000, archive size 400, and run for 40 generations. The method of tree generation is ramped half and half [15]. The terminal set for the tree consists of technical and fundamental financial factors describing a company’s performance, plus constants. The function set includes addition, subtraction, multiplication, division, power 2, and power 3. The MOGP has two conflicting objectives to satisfy; return maximization and risk minimization. Return is defined as the annualized average return, and risk is the standard deviation of the annualized average return. The MOGP solutions are trees, each of which represents a non-linear model of financial factors.

### 3.5 Historical Data

Our research uses historical data from the London Stock Exchange market, the FTSE100 for 80 months, from May 1999 to December 2005. The stock universe consists of 82 stocks that were part of the FTSE100 constituent for the duration of the time period investigated; i.e. companies that merged split or fell out of the FTSE100 are excluded. For these 82 stocks, the monthly values of 22 factors describing the represented companies’ data were downloaded from Reuters. A small sample of a company’s data is shown in Table 1. We normalize all the factor values before using them within the investment simulator and the GP, in order to minimize the effect of a number of parameters with high ranges dominating the model. Also, normalization of the parameters should have a positive effect on robustness, because all perturbations in parameters in different environments are put into similar perspective with changes in other parameters and with changes from the values dealt with in the training environment.

The total period is divided into training and validation. For training (in-sample), 48 months from May 1999 to April 2003 are used. For validation (out-of-sample), the data is that of the last 20 months from May 2004 to December 2005. The return on investment (ROI) of an “Index Fund” portfolio that invests one million pounds, with equal proportions in the 82 stocks of the universe, over the two time periods selected for training and validation, is depicted respectively in Figures 3 and 4.

Table 1: A sample of Company Data (BT)

Date	Close	Momen- tum	Volume	Price toCash	Price toBook
03-05	205.5	-0.0143	0.9517	-2.6E-12	0.4849
04-05	199.75	-0.0278	0.7323	-5.1E-12	0.4989
05-05	213.25	0.0675	0.8774	1.2E-11	0.4673
06-05	230	0.0785	1.2227	1.4E-11	0.43328
07-05	227.5	-0.0108	0.9232	-2.0E-12	0.4380
08-05	215.5	-0.0527	1.0339	-9.7E-12	0.4624
09-05	222.25	0.0313	0.8678	5.8E-12	0.4483
10-05	213	-0.0416	0.9671	-7.7E-12	0.4678
11-05	213.5	0.0023	1.1891	4.3E-13	0.4667
12-05	222.75	0.0433	0.7577	8.0E-1 2	0.4473

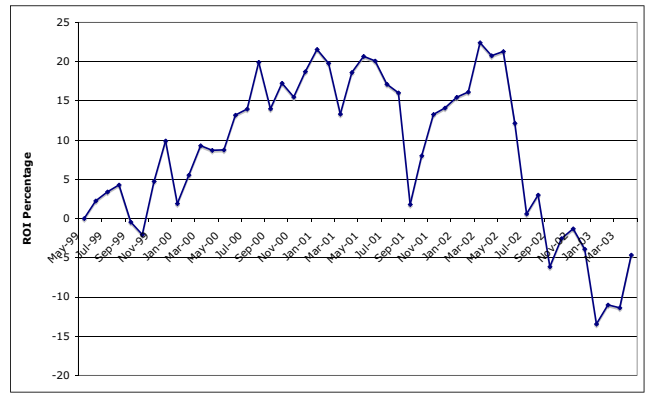


Figure 3: Performance of Index Fund During Training Period

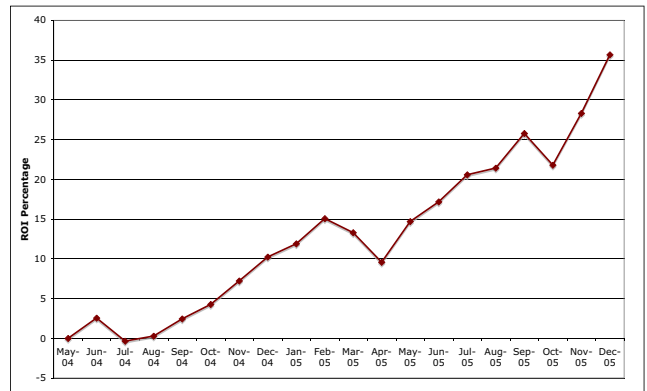


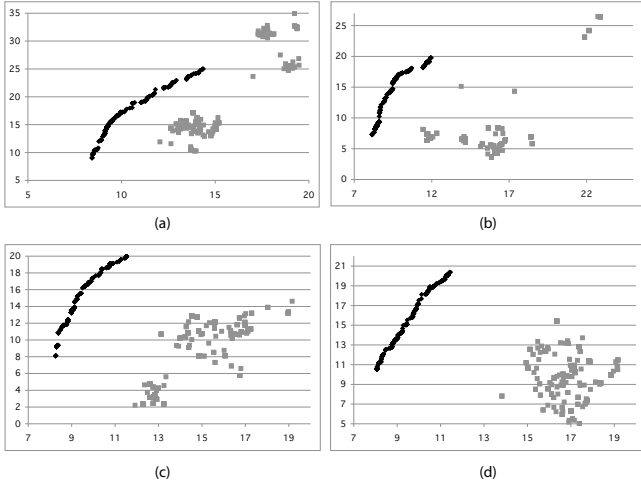
Figure 4: Performance of Index Fund During Validation Period

## 4. SPEA2 PERFORMANCE

The SPEA2 algorithm [28] is currently one of the best general-purpose MO algorithms. Furthermore, in a recent paper, [25] demonstrated that SPEA2 had the best performance on a portfolio optimization problem with real life constraints. They compared the SPEA2 performance with that of NSGAI [9], MOGA [11], and VEGA [23], and found that the SPEA2 had the best performance in terms of quality of solutions and their distribution even with a small number of generations.

To test the SPEA2 performance on out-of-sample data, results of 15 runs of the SPEA2 algorithms are reported in this section. In each run, the solutions on the Pareto front were tested on the validation data of 20 months, equivalent to using an investment strategy represented by the solution tree to manage a new financial portfolio. The performance of the algorithm on the validation data varies between the runs. Figure 5 presents four runs with the Pareto front in training and in validation. It is noticed in these graphs not only that the performance is worse than in training, but also that the Pareto front as a whole loses its distribution characteristics. Another more serious problem is illustrated in Figure 6. The figure shows a solution P1 that in training displayed relatively high returns at relatively high risk — but in validation it had relatively the worst return with low-to-medium relative risk. Another solution P2 that was

relatively medium-return/medium-risk in training became relatively low-return with relatively medium-to-high-risk in validation, and also became dominated by other solutions. The solution P3 changed from relatively medium-return in training to relatively low-return in validation and clearly became dominated by several other solutions that achieved the same risk with higher return.



**Figure 5: SPEA2 Pareto Fronts in Training (black) and Validation (grey) over four runs**

This is of particular importance in our application. A fund manager employing an investment strategy clearly would require (at least) that the strategy maintains its objectives' characteristics relative to the other available strategies. Although the new set of solutions may still be on the efficient frontier, from the point of view of the fund manager they would still be wrong for her purpose.

## 5. NEW METRICS FOR MO ROBUSTNESS IN VOLATILE ENVIRONMENTS

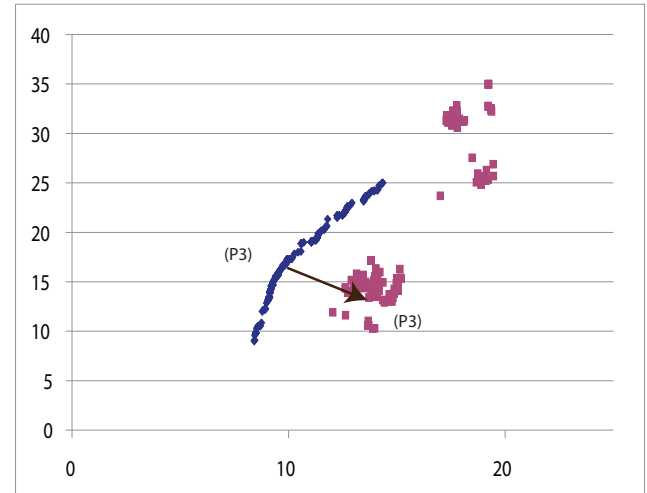
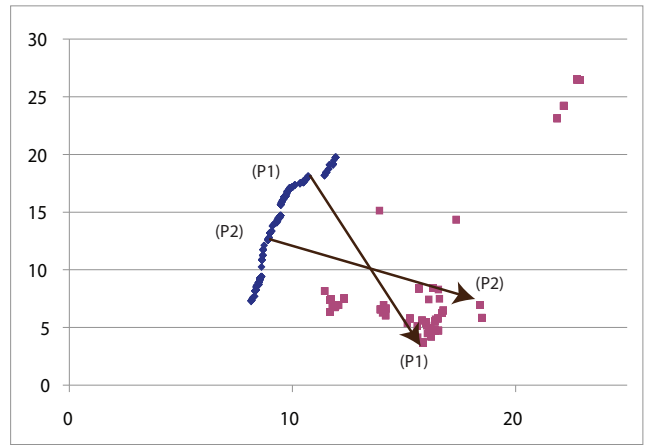
We propose that in order to quantify robustness we need to assess several aspects. First, the quality of the solutions in the new environment. Second, how much they have changed their objectives cluster and rank, amongst other solutions on the Pareto front. Third, these solutions were non-dominated in training; did they continue to be non-dominated in validation? Finally, how good is the spacing, spread and distribution of solutions on the new front formed in validation?

In this work, we are interested in the analysis of the performance on the new environment in terms of how much the solutions have changed their objectives profile and rank. In the following, we will start by defining what is meant by a cluster of objective values, and rank of a solution. Then, we will introduce new definitions for robustness. Followed by, the suggested change in the algorithm to provide bias towards solutions with robust behaviour across different environments.

### 5.1 Definitions

#### Definition 1: Clusters of Objective Values:

Consider a problem with  $m$  objectives to be minimized, and assume that the value of each objective can be classified as one of *high*, *medium*, *low*. Solutions on the Pareto front



**Figure 6: Example of Solutions Changing their Objectives Profile(Cluster). The vertical axis is percentage return on investment, and the horizontal axis measures risk (standard deviation of monthly returns).**

are then classified into sets (“clusters”) such that members of a cluster have similar classifications for each of their objectives. We say that each cluster has a certain *profile* of objective values. Thus, there is a maximum of  $3^m$ , and a minimum of 3, clusters (the minimum 3 would be one on each extreme, and one with medium values for all objectives). Each cluster  $C_i$  is a set containing  $N$  solutions together with a centroid  $c$ , which is a vector of  $m$  real values. The profile of a cluster is a vector of the  $m$  classification values of the centroid, relative to the other centroids.

Hence, we have:

$$Profile(C_i) = \langle Profile(c_1), Profile(c_2), \dots, Profile(c_m) \rangle \quad (4)$$

Where  $Profile(c_i) \in \{L, M, H\}$

**Definition 2: Rank of a Solution:**

Each solution on the front will have a rank (see Section 5.2) relative to the other solutions on the front, and based on its objectives' values. The objectives' ranking order of solutions is not defined by the absolute values of the objectives, but rather by their relative value in comparison to the other solutions on the current front. Each solution on the front has a cluster membership, and a rank.

For any solution  $x_k$ , the following two functions are defined:

$$Cluster(x_k) = C_i \mid x_k \in C_i \quad (5)$$

$$Rank(x_k) = (r_{o1}, r_{o2}, \dots, r_{om}) \quad (6)$$

where  $r_{oi}$  is the rank of objective  $i$ . The algorithm used to calculate the rank value is explained in Section 5.2.

**Definition 3: Robustness of a Solution:**

Robustness of a solution to a multiobjective problem is defined qualitatively as the degree of its insensitivity to changes in the environment, and is measured quantitatively by how well it preserves its cluster and rank in a new environment. For example, in a two-objective problem where only three clusters exist, if a solution moves from a cluster  $\langle high, high \rangle$  to  $\langle medium, medium \rangle$  then we measure this as a move of length 2, whereas if it moves from  $\langle high, high \rangle$  to  $\langle low, low \rangle$  then this is given a measure of 4. Changes in rank can be measured in a straightforward manner, for example using element-wise differences across the rank vector and then adding the differences.

$$\delta(x_k) = \sum_{i=1}^m (r_{oi}^{env1} - r_{oi}^{env2}) \quad (7)$$

**Definition 4: Robustness of the Pareto Front:**

Robustness of the Pareto front between two environments is defined qualitatively by two measures:

1. How well its solutions maintain their cluster of objectives between the two environments.
2. How well its solutions' ranks have remained closely correlated between the two environments.

The former is measured quantitatively by calculating the mean difference in cluster movements across all  $n$  solutions in the front.

$$Mean\ cluster\ distance\ change = \sum_{k=1}^n (\delta(x_k)) \quad (8)$$

The latter could be measured quantitatively using a rank correlation test (e.g. Spearman Rank Correlation [21]), though the results should be interpreted with care since it presents a very high hurdle — typically a rank correlation test will only give high values where almost every individual on the front retains its rank.

**Definition 5: Robustness of a MO Algorithm:**

Robustness of a multiobjective algorithm is defined qualitatively as its Pareto solutions' insensitivity to changes in the environment such that they maintain their profile of objectives in a variety of environments, as well as the Pareto front maintaining its diversity and uniform distribution.

## 5.2 Clustering and Ranking Algorithms

In our experiments, clustering of solutions on the front is implemented using the k-means algorithm to group solutions into 3 clusters (high return/high risk), (medium return, medium risk), (low return, low risk). The clustering analysis is performed twice: at the end of the last generation of training, and after validation.

The ranking of solutions' objectives values is done independently for each objective. Each solution rank is a vector of  $n$  values, each corresponding to its rank along a specific dimension. It is implemented by sorting them in ascending order, with ties assigned the average rank of all values with the same value. In the case of SPEA2, ranking is performed in the last generation only for comparison purposes. In case of the modified algorithm (R-SPEA2 — see below), it was performed in each generation after the evaluation of the solutions.

## 5.3 Fitness Assignment for R-SPEA2

We implement a robust version of SPEA2 — R-SPEA2 — where solutions' robustness is measured during evolution and used to bias the fitness as follows:

- In each generation, and after the front has been identified:
  - Identify the ranks of all solutions;
  - Run every solution in a different environment (12 months from May 2003 - April 2004), and identify the rank of each solution in the new environment;
  - Assign a robustness value  $R$  to each solution based on how well the solution preserved its rank — the less the change, the better the robustness value.
  - The solution fitness value is incremented by the robustness value.
- Tournament selection is performed as usual. It will now prefer solutions which are: non-dominated, in less dense areas of the front, and which are more robust across the diverse training environments.

An interesting point here is how to combine the difference of ranks along all the dimensions. Since we have separate ranks for each objective dimension, we end up with  $n$  rank differences that need to be combined into one value. The rank differences have been normalized to a value in the range  $[0, 1]$ , with 0 being the best robustness, and 1 the worst. In situations where a single metric is required, we choose to multiply these rank differences together, since this slightly biases in favour of solutions where all rank differences are very low. Hence, the robustness measure is:

$$R = \prod_{i=1}^n r_{oi} \quad (9)$$

The total fitness of the solution is then incremented by the robustness value, and is used for mating selection.

## 6. RESULTS

In this section, we present the results of applying the robustness measures to SPEA2, and the Pareto front produced by four runs of the modified algorithm during validation.

In Figure 7, the Pareto front on validation shows slightly better spread characteristics than achieved in the original algorithm, with fewer solutions losing their non-dominance.

Table 2 shows the mean distance of cluster change between training and validation of all solutions on the front (and the percentage of solutions that maintained their cluster). R-SPEA2 has on average more than half of the solutions on the Pareto front keeping within the objectives' profile achieved in training. A non-parametric Ranked T-Test applied to the two distributions of means gives a p-value of  $1.2 \times 10^{-6}$ , indicating that this difference between SPEA2 and R-SPEA2 is statistically highly significant (the p-value for the percentages is 0.07364).

By contrast, Table 3 indicates the much tougher test of Spearman rank correlation of all individuals: results of 15 runs show that R-SPEA2 achieved an average improvement of only 10% of the coefficient value (significant only at the 15% level for Objective 2).

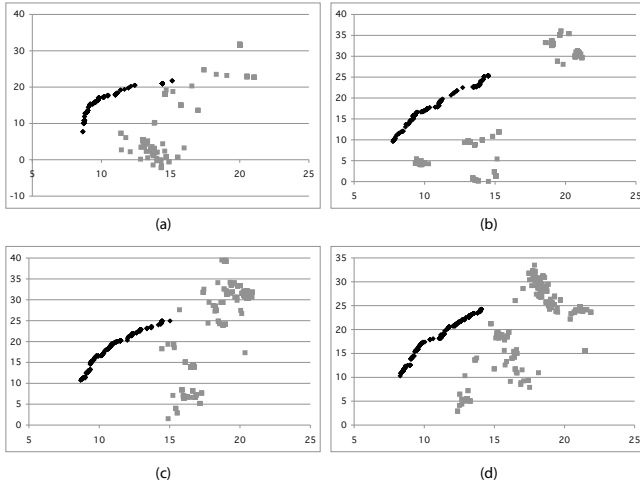


Figure 7: R-SPEA2 Performance in Training (black) and Validation (grey) over four runs

## 7. SUMMARY AND CONCLUSION

Multiobjective methods are ideal for evolving a set of portfolio optimisation solutions that span a range from high-return/high-risk to low-return/low-risk, and an investor can choose her preferred point on the risk-return frontier. However, there are no guarantees that a low-risk solution will remain low-risk.

The robustness of a Multiobjective Genetic Programming (MOGP) algorithm such as SPEA2 is vitally important in the context of the real-world problem of portfolio optimisation. We have demonstrated that SPEA2 (acknowledged to be one of the best general-purpose MOGP algorithms, and giving good results on portfolio optimisation problems) is, while being excellent in other respects, unable to guarantee robust solutions.

We have analysed the robustness of individual solutions and of a Pareto front in terms of insensitivity to changes in

Table 2: Mean distance of cluster change and percentage number of solutions changing cluster for SPEA2 and R-SPEA2

Run	SPEA2		R-SPEA2	
	Mean	Percentage	Mean	Percentage
1	0.656	57.75%	0.657	100.00%
2	1.615	63.00%	0.340	20.25%
3	1.100	52.00%	0.402	68.50%
4	1.365	62.25%	0.440	47.25%
5	1.120	89.00%	0.720	47.75%
6	1.742	100.00%	0.525	54.00%
7	1.420	61.25%	0.425	28.00%
8	0.965	42.25%	0.295	46.00%
9	0.747	73.00%	0.590	61.25%
10	0.890	44.50%	0.397	70.75%
11	0.365	18.25%	0.590	56.00%
12	1.460	69.00%	0.530	42.00%
13	0.675	33.75%	0.245	35.00%
14	1.575	78.75%	0.085	21.00%
15	1.707	100.00%	0.430	32.25%
<b>AVG</b>	1.160	62.98%	0.444	48.66%
<b>StdDev</b>	0.432	23.27	0.166	21.15

Table 3: Correlating between Training and Validation: Spearman Coefficients for Objectives 1 and 2

Run	SPEA2		R-SPEA2	
	Obj1	Obj2	Obj1	Obj2
1	0.6491	0.6763	0.7664	0.0922
2	0.0228	0.1440	0.9484	0.7047
3	0.4590	0.3856	0.7482	0.7594
4	0.6331	0.4315	0.6372	0.6868
5	-0.2085	0.7051	0.5859	0.5655
6	0.0361	0.2633	0.3686	0.6912
7	0.5495	0.5905	0.2326	0.3963
8	0.6903	0.3665	0.5740	0.8032
9	0.9494	0.3184	0.5533	0.2687
10	0.7765	0.7307	0.7255	0.4083
11	0.6379	0.8857	0.4605	0.5543
12	0.6119	0.7510	0.8242	0.8078
13	0.7345	0.6060	0.8546	0.8514
14	0.7930	0.0710	0.3325	0.6808
15	0.6957	-0.3042	0.5706	0.7235
<b>AVG</b>	0.5354	0.4414	0.6122	0.5996
<b>StdDev</b>	0.3268	0.3149	0.2043	0.2189

the environment, and have provided new definitions of robustness for both individual solutions and the whole front. We have demonstrated the problem by comparing a training environment with a very different validation environment, showing how SPEA2 solutions on the Pareto front can swap their relative positions in terms of their objectives cluster. We have explained how we obtain quantitative measures of robustness, and we have utilised one such measure to create ‘‘R-SPEA2’’, a robust variant of SPEA2. The results of early experiments show that R-SPEA2 offers a statistically *highly significant* improvement in the mean number of cluster changes experienced by individual solutions when moving from a training environment to a validation environment.

Future work includes investigating different methods for combining ranks, and whether to incorporate these ranks into fitness measures or use them to exclude solutions that are low on robustness. We are also developing our definitions of robustness to facilitate recognition and measurement of a wide range of robustness behaviour, especially with respect to the robustness of the Pareto front as a whole. Addi-

tional further work includes improving population diversity to improve overall MOGP performance, and the use of the new robustness metrics for comparison of the performance of various MOEA algorithms on out-of-sample data.

## 8. REFERENCES

- [1] R. Armananzas and J. Lozano. A multiobjective approach to the portfolio optimization problem. *IEEE Congress on Evolutionary Computation*, 2:1388–1395, 2005.
- [2] C. Barrico and C. H. Antunes. Robustness analysis in multi-objective optimization using a degree of robustness concept. *IEEE Congress on Evolutionary Computation*, 2006.
- [3] C. A. Coello. Recent trends in evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pages 7–32. Springer, 2005.
- [4] C. A. Coello. Evolutionary multi-objective optimization and its use in finance. In *Handbook of Research on Nature Inspired Computing for Economy and Management*. Idea Group Publishing, 2006.
- [5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, Chichester, UK, 2001.
- [6] K. Deb and H. Gupta. Searching for robust pareto optimal solutions in multi-objective optimization. *Evolutionary Multiobjective Optimization*, pages 150–164, 2005.
- [7] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14:463–494, 2006.
- [8] K. Deb, D. Padmanabhan, S. Gupta, and A. K. Mall. Reliability-based multi-objective optimization using evolutionary algorithm. *Evolutionary Multi-Criterion Optimization, Springer LNCS*, 4403:1611–3349, 2007.
- [9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: NSGA-II. *Proc. of the Parallel Problem Solving from Nature VI Conference*, 2000.
- [10] N. Drechsler, R. Drechsler, and B. Becker. Multiobjective optimization based on relationship favour. *International Conference on Evolutionary Multi Criterion Optimization*, pages 154–166, 2001.
- [11] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proc. of the Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993.
- [12] A. Gaspar-Cunha and J. Covas. Robustness in multi-objective optimization using evolutionary algorithms. *Computational Optimization and Applications*, 2007.
- [13] H. Gupta and K. Deb. Handling constraints in robust multi-objective optimization. *IEEE Congress on Evolutionary Computation*, 1:25–32, September 2005.
- [14] Y. Jin and B. Sendoff. Trade-off between performance and robustness: An evolutionary multi-objective approach. *Evolutionary Multiobjective Optimization, Springer. LNCS*, 2632:237–251, 2003.
- [15] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [16] D. Laura. A multi-objective evolutionary approach to the portfolio optimization problem. *International Conference on Computational Intelligence for Modelling, Control, and Automation*, 2005.
- [17] J. Li and S. Taiwo. Enhancing financial decision making using multi-objective financial genetic programming. *Proc. of IEEE Congress on Evolutionary Computation*, pages 7935–7942, 2006.
- [18] M. Li, S. Azarm, and V. Aute. A multi-objective genetic algorithm for robust design optimization. In *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, pages 771–778. ACM, 2005.
- [19] S. Luke et al. A java-based evolutionary computation research system, version 15. <http://www.cs.gmu.edu/~eclab/projects/ecj/>.
- [20] H. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.
- [21] W. Mendenhall, R. Beaver, and B. Beaver. *Introduction to Probability and Statistics*. Duxbury Press, 1999.
- [22] A. Mukerjee, R. Biswa, K. Deb, and A. P. Mathur. Multi-objective evolutionary algorithms for the risk-return trade-off in bank loan management. *International Transactions in Operations Research*, 9:583–597, 2002.
- [23] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. *Genetic Algorithms and their Applications: Proc. of the First International Conference on Genetic Algorithms*, pages 93–100, 1985.
- [24] F. Schlottmann, A. Mitschele, and D. Seese. A multiobjective model framework for the integrated management of financial risks. *Quantitative Methods in Finance Conference, Sidney, Australia*, 2005.
- [25] P. Skolpadungket, K. Dahal, and N. Harnpornchai. Portfolio optimization using multi-objective genetic algorithms. *IEEE Congress on Evolutionary Computation*, pages 516–523, 2007.
- [26] A. Sulfow, N. Drechsler, and R. Drechsler. Robust multi-objective optimization in high dimensional spaces. *Evolutionary Multiobjective Optimization (EMO)*, pages 715–726, 2007.
- [27] M. G. C. Tapia and C. A. C. Coello. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. *IEEE Congress on Evolutionary Computation, CEC 2007*, pages 532–539, 2007.
- [28] E. Zitzler, M. Laumanns, , and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100. International Center for Numerical Methods in Engineering, 2002.