Implementation of Novel Sector Weight and Google Trends Data Objectives using MOEA/D Curtails Systematic Risk for Quintessential Investors

Abishek Ravindran

Category: Behavioral and Social Sciences

A. Rationale

I. Introduction

Portfolio optimization is a dilemma that has daunted researchers and investors alike for decades for its dynamic and complex nature [16]. After the development of Modern Portfolio Theory, alternatively called the Markowitz Mean-Variance Model, a variety of models have been designed, including Tobin's two-fund theorem and the Black-Scholes-Merton model for procuring options prices [4, 15, 20]. While such advancements have been made, the Markowitz model endures as a hallmark of portfolio optimization [16]. Though the Markowitz Mean-Variance Model provides a paramount foundation for subsequent inquiry, it fails to consistently yield feasible solutions [21]. Major developments have been made to rectify this problem through the implementation of metaheuristics, constraints, and additional objectives [16].

With the inclusion of constraints, including the budget, asset weight, and class constraints, the problem becomes non-deterministic polynomial-time hard (NP-hard), rendering an exact solution to be nearly unattainable [17]. Metaheuristics pose great efficacy in such scenarios by providing a framework to formulate sufficient solutions when information may be incomplete or imperfect [22]. One such metaheuristic is the evolutionary algorithm that seeks to imitate the process of reproduction in life under the premise that "more fit" members will be more inclined to survive and proliferate while "unfit" members will be gradually removed from the population [11]. The employment of evolutionary algorithms for multi-objective problems provides a collection of benefits: the population-based nature of the algorithm allows for a set of solutions to be developed in a single run and for the solutions to have enhanced diversity by exploring a wider section of the efficient set of solutions, termed the Pareto Front. Multi-objective evolutionary algorithm based on decomposition (MOEA/D), is an evolutionary algorithm that utilizes a scalarization function to compartmentalize a multi-objective problem into multiple single-objective subproblems [18].

Recent developments in multi-objective evolutionary algorithms have allowed for the exploration of hyper-dimensional search spaces [18]. Rather than producing solutions representing the traditional two-dimensional Pareto Front for risk and return, these models strive to create efficient surfaces that portray the trade-off between three or more factors. Additional objectives including expected shortfall and annual dividends have been implemented to provide portfolios that meet the preferences of investors [1, 16].

A major flaw that has been noted in the Markowitz Model is a failure to produce efficacious portfolios under unstable market conditions. In the event of a market failure (e.g. Tech Bubble of 2001, Financial Crisis of 2008), average investors must account for trends in human behavior and create portfolios that are well diversified among classes [13]. As investors allocate capital towards goals such as education and retirement, diversified portfolios that minimize both risks seen in measures such as behavioral data and over-allocation to sectors are as imperative as standard risk measures including variance at risk and covariance.

II. Markowitz Model

Prior to the development of Modern Portfolio Theory, portfolio selection was primarily based on the dividend discount model, which was criticized for its fundamental assumption of income solely in the form of dividends and what quantified a “discount” being ambiguous [6]. Modern Portfolio Theory, pioneered by Harry Markowitz in 1952, created an empirical framework to analyze the efficacy of portfolios [15]. The model is seen below:

$$min\ \rho(x) = \sum_{i=1}^{n} w_i w_j \sigma_{ij} \qquad (1)$$

$$max\ \mu(x) = \sum_{i=1}^{n} w_i \mu_i \qquad (2)$$

where w_i, σ_{ij}, μ_i represents the weight of asset i, covariance of assets i and j, and return of asset i, respectively.

The model relies on the assumption that all investors seek to maximize return while minimizing risk, with the two being conflicting in nature. While each individual possesses a different risk preference, the model relies on an efficient set of portfolios each with a different risk and return. These portfolios are denoted as components of the Pareto efficient set, in which no potential change in asset allocation can warrant a further optimized portfolio.

III. Evolutionary Algorithms

Holland’s work is noteworthy for bolstering the recognition of evolutionary algorithms as a metaheuristic, or search algorithm that can distinguish sufficient solutions in problems with imperfect information [11]. A schematic for evolutionary algorithms can be seen in Figure 2. After constructing a set of solutions that constitute the population, customarily pseudo-randomly, the EA develops offspring through recombination and mutation, representative of propagation in nature [17]. As the population procreates over a multitude of generations, the solutions approach the Pareto efficient frontier.

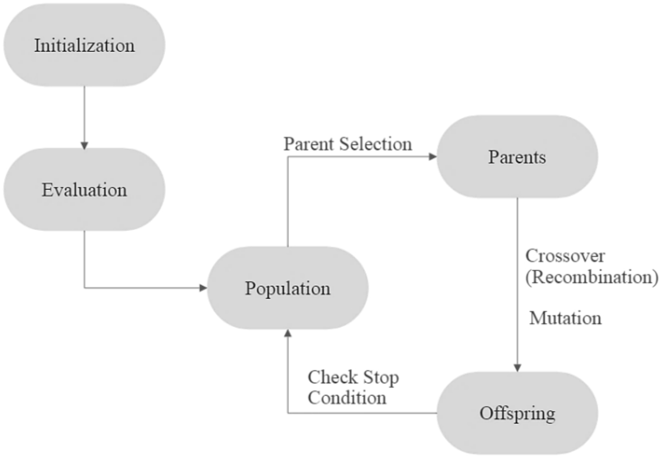


Figure 1: Coding schematic for evolutionary algorithms. Portrays population-based nature through which enhanced offspring are created and solutions are optimized (Adapted from Holland, J.H [11])

The efficacy of evolutionary algorithms is by virtue of their population-based nature. The metaheuristic is frequently recognized for its capacity for self-optimization, as advantageous traits are enhanced in the solution set as the algorithm progresses [18]. Secondly, the population-based nature of evolutionary algorithms aids in solving problems with multiple conflicting objectives by creating a solution set that is Pareto efficient [16]. The most prevalent algorithms currently include Strength Pareto Evolutionary Algorithm, Pareto-envelope based selection algorithm (PESA-II), and Non-Dominated Sorting Genetic Algorithm [7, 9, 25]. However, there are several notable flaws with these frameworks including dimensionality, making it difficult to form a population of diverse solutions, and saturability, as the solution set becomes populated with suboptimal members to curtail selection pressure [18]. A Multi-objective evolutionary algorithm based on decomposition (MOEA-D) resolves the aforementioned dilemmas by engaging scalarization and independent subproblems to handle disparate values and saturation, respectively [23].

B. Research Question and Hypothesis

This study proposes the inclusion of sector weight and Google Trends data objectives, to reduce systematic risk within the model by accounting for factors not seen in traditional empirical data. The model can be seen in equations 3 and 4, respectively

$$min\ \tau(x) = \sum_{c=1}^{m} (U_c - \tau_c)\ if\ \tau_c > U_c \quad (3)$$

$$min\ \beta(x)\ = \sum_{i=1}^{n} w_i \beta_i \quad (4)$$

where $U, \tau,$ and β represent the class upper bound, class weight, and variance of Google Trends data, respectively.

.

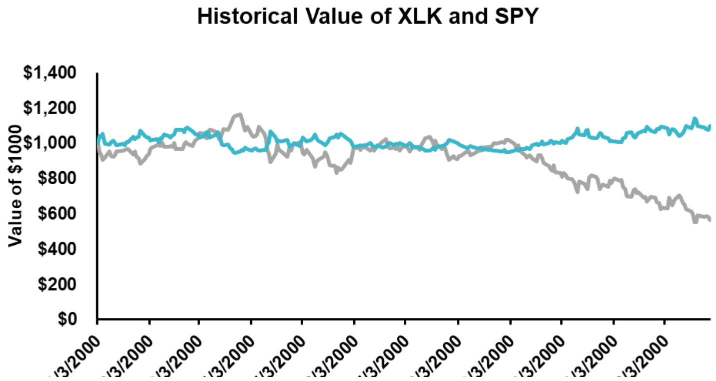


Figure 2: Historical value of 1000 Dollars invested into XLK and SPY during 2000. XLK represents the technology sector that faced dramatic losses while SPY represents an overall market-weighted index with consistent returns

The former objective was implemented to reduce the risk associated with overweighting a sector. This is highlighted in Figure 1; while the S&P 500 fund, SPY, that encompasses assets from diverse classes gained 9.7 percent in value from January 2000 to December 2002, the technology sector fund lost 43.5 percent in value over this time period. Events such as the “Tech Bubble” tend to have a significant impact on a specific sector. While class constraints have been implemented in previous studies, the class-weight objective utilizing a summation of overweighting for classes is a novel metric introduced in this paper.

The latter objective utilizes the capacity of Google Trends to function as a leading indicator to predict potential risk that may arise. The concept of Google Trends data originated in the work of Kristoufek, in which a portfolio normalized based on an inverse function between asset weights and search queries was found to outperform a uniform-weighted DOW index [12]. However, studies later criticized Google Trends as having no more efficacy than historical returns, imploring for further research to find a more effective approach [5]. It is crucial to note that stocks primarily move due to investor perception of events rather than the events themselves; Google Trends offers utility in anticipating such variation in attitude. In the literature, the risk based on Google Trends metric relied on the mean number of search queries, offering that a larger number of queries correlates with greater risk. However, while this may have worked for brief time spans and for an index with large-capitalization constituents, this fails to cohere to the Markowitz model. Therefore, this study proposes the use of variance over a longer time span of Google Trends data. This is the first study to implement Google Trends as a supplementary metric for the Markowitz model, in comparison to studies that have normalized asset weights exclusively based on Trends data.

In order to broadening the pertinence of the proposed model, several additional constraints were covered in the model. The constraints can be seen below.

$$\sum_{i=1}^{n} w_i \ = \ 1 \quad (5)$$

$$l_i \leq w_i \leq u_i \ (6)$$

$$i \ \in \{1, \ldots, n\} : \ w_i \geq 0 \ | \ \leq \ N \ (7)$$

where l_i and u_i represent the lower-bound and upper-bound of asset weights, respectively (Adapted from Crama & Schyns [8]). Equation 6 conveys the budget constraint, an intuitive constraint that ensures that the asset weights of the portfolio consistently aggregate to 1. The asset floor and ceiling constraints, seen in Equation 6, confine the weight of assets to prevent short selling, or negative weights, and overweighting of assets, respectively. Finally, the cardinality constraint in equation 7 prevents portfolios comprised of an inordinate number of assets. While turnover and trading constraints have been proposed in the literature, these constraints are only imperative for models that are intended for the short term [8]. The proposed model addresses long-term investments and therefore does not require such constraints.

The purpose of this study has two facets. In order to tailor financial models towards the quintessential investor, the gap between computational and behavioral models must be bridged. Towards this aim, the primary goal of this study is to effectively implement and optimize MOEA-D towards the specificities of the problem to develop an algorithmic framework for future inquiry. The secondary goal of this study is to determine the efficacy of the supplementary objectives (Google Trends and sector-weight) towards developing risk minimizing portfolios.

The hypothesis of this study was that the integration of the supplementary objectives will aid in minimizing losses in the out-of-sample data set.

C. Methodology

I. Introduction

Experimentation will be divided into two components. The first component will consist of parameter fine tuning, to ensure that parameters including crossover rate, mutation rate, and population size are optimized within the algorithm. Statistical computation will then be completed to determine the optimal combination of parameters. After this is completed, tests will be done on both the in-sample and out-of-sample data sets for a total of 7 variables over 5000 experiments to determine the relative efficacy of each objective on two separate data sets. A control will consist of the bi-objective model without any additional objectives.

II. Data Set Collection

Data will be compiled for the S&P 100 during the time period 2003-2009. The data will then be partitioned between an in-sample and out-of-sample time period to allow for tests of the efficacy of the augmented model rather than the algorithm. The in-sample time period will extend from January 1, 2003 to December 31, 2007, while the out-of-sample time period will encompass data from January 1, 2008 to December 31, 2009. To clarify, only the in-sample data will be used for parameter fine tuning. In the second portion of experimentation, after a portfolio is developed based on the in-sample data set, tests of its projected return and variance were completed on the out-of-sample data set. All historical returns will be collected via *Yahoo! Finance* and calculations including covariance were completed using *R* for statistical computing.

The constituents chosen will be a part of the S&P 100 at the center of the time period on June 30, 2006. These assets were strategically chosen to maximize the time that chosen assets would have been a part of the index. While indices such as the S&P are dynamic, changing assets within the data set for the evolutionary algorithm will elicit computational errors. Out of a total of 100 constituents, data for a total of 88 constituents was compiled. Data for the following assets, denoted by ticker, were not included due to a lack of public availability: AMAT, BLS, BUD, CAH, EMC, FDC, GOOGL, VIAB, WB, and WMIH. Data for companies becomes unavailable due to restructuring or bankruptcy, which may skew positive results. However, similar limitations have been found in older data sets, and uniformity within the data set would allow for conclusions of the relative efficacy of different parameters and objectives.

In addition, the search volume data for the 88 constituents utilized will be collected via *Google Trends*. The technique applied for selecting search queries will be adopted from the literature, as a ticker strategy was found to be more effective than a ticker "stock" strategy. This was elucidated as investors need not specify that they are searching for the stock when it is intrinsic in the symbol [12]. However, for this study several tickers were found to be ambiguous and easily mistaken for another phrase. These tickers were AA, CAT, DD, EBAY, LOW, MET, S, T UPS, and USB. For these tickers, a ticker stock strategy will be implemented (i.e., AA stock).

III. MOEA-D and Parameter Fine-Tuning

Though MOEA-D is to be implemented, the details of implementation are rather ambiguous at this time. However, it can be valuable to garner a foundational understanding of the algorithmic framework itself. The initial objective function will be decomposed into *N* scalarized subproblems, correlating with the population size *N* [22]. Each subproblem will be updated through recombination at each iteration of the algorithm. Novel solutions will be compared with neighboring subproblems, as a solution may be preferred for a select set of weights. By limiting the neighborhood of solutions, saturation becomes improbable [23].

In optimizing an evolutionary algorithm, there are assorted factors that have an extensive impact on its utility. The exact values that will be used have not been extrapolated at this time. However, a short description of what entails an optimal value for each can be found below:

- Number of iterations: algorithm should be run to ensure convergence at optimal solutions, while concurrently minimizing computational time
- Crossover Rate: an optimized crossover rate should prevent solutions from losing critical elements of parent solutions while allowing for variation from the offspring
- Population Size: an appropriate size should ensure that solutions comprehensively represent the efficient surface while avoiding negligibly different solutions
- Mutation Rate: the mutation rate should allow for exploration of unexamined portions of the solution space while preserving unique elements of solutions

IV. Objective Analysis

After the algorithm is optimized, tests will be conducted on the efficacy of the two objectives proposed. The control group will apply a bi-objective model that does not consider any supplementary objectives. All testing will be completed for 5000 iterations, due to the volatile nature of metaheuristics and the considerable differences that arise with a novel objective being included within a model. It is worth noting that while fitness values will be tested, they posed little efficacy at this point in testing; the nature of the experiments raised the significance of the future, or projected, returns and variance. While the two supplementary objectives are meant to be used concurrently, they must be tested discretely to ascertain their independent value.

The model's ability to produce portfolios with greater return or lower risk on historical data is inconsequential, but the future results that will be conveyed through the out-of-sample data sets includes valuable information.

D. Bibliography

[1] Anagnostopoulos, K., & Mamanis, G. (2010). A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research*, *37*(7), 1285–1297. doi: 10.1016/j.cor.2009.09.009

[2] Anagnostopoulos, K., & Mamanis, G. (2011). Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science*, *8*(3), 259–279.

[3] Bäck Thomas. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. New York: Oxford Univ. Press.

[4] Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy, 81*(3), 637–654.

[5] Challet, D., & Ayed, A. B. H. (2014). Do Google Trend Data Contain More Predictability than Price Returns? *SSRN*.

[6] Coello, C. C. (2006). Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, *1*(1), 28–36. doi: 10.1109/mci.2006.1597059

[7] Corne, D. W., Jerram, N. R., Knowles, J. D., & Oates, M. J. (2001). PESA-II: region-based selection in evolutionary multiobjective optimizatio. *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 283–290.

[8] Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, *150*(3), 546–571. doi: 10.1016/s0377-2217(02)00784-1

[9] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computing*, 182–197. doi: 10.1109/4235.996017

[10] Ehrgott, M., Klamroth, K., & Schwehm, C. (2004). An MCDM approach to portfolio optimization. *European Journal of Operational Research*, *155*(3), 752–770. doi: 10.1016/s0377-2217(02)00881-0

[11] Holland, J. H. (1984). Genetic Algorithms and Adaptation. *Adaptive Control of Ill-Defined Systems*, 317–333.

[12] Kristoufek, L. (2013). Can Google Trends search queries contribute to risk diversification? *Scientific Reports*, *3*(1). doi: 10.1038/srep02713

[13] Lo, A. W. (2005). Reconciling Efficient Markets with Behavioral Finance: The Adaptive Markets Hypothesis. *Journal of Investment Consulting*, *7*(2), 21–44.

[14] Lwin, K., Gu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 757–772.

[15] Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, *7*(1), 77–91.

[16] Metaxiotis, K., & Liagkouras, K. (2012). Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literature review. *Expert Systems with Applications*, *39*, 11685–11698.

[17] Moffaert, K. V., Drugan, M. M., & Nowe, A. (2013). Scalarized multi-objective reinforcement learning: Novel design techniques. *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. doi: 10.1109/adprl.2013.6615007

[18] Purshouse, R. C., & Fleming, P. J. (n.d.). On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, *11*(6).

[19] Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *The Journal of Finance*, *19*(3).

[20] Sharpe, W. F. (1994). The Sharpe Ratio. The Journal of Portfolio Management, 21(1), 49–58. doi: 10.3905/jpm.1994.409501

[21] Steinbach, M. S. (n.d.). Markowitz Revisited: Mean-Variance Models in Financial Portfolio Analysis. *SIAM Review*, *43*(1), 33–85.

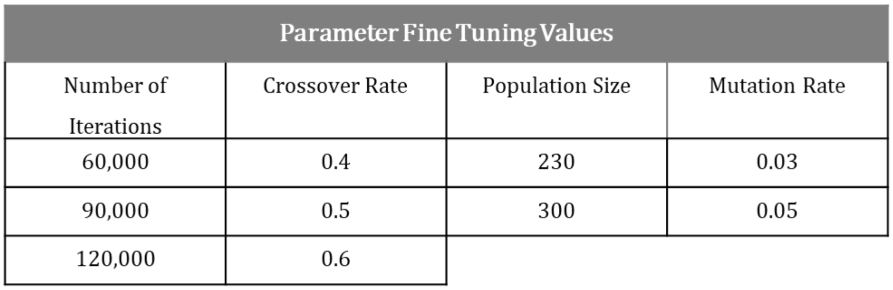
[22] Zhang, Q., & Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731. doi: 10.1109/tevc.2007.892759

[23] Zhang, Q., Li, H., Maringer, D., & Tsang, E. (2010). MOEA/D with NBI-style Tchebycheff approach for portfolio management. IEEE Congress on Evolutionary Computation. doi: 10.1109/cec.2010.5586185

[24] Zhang, Z., Xu, Y., Yang, J., & Li, X. (2015). A Survey of Sparse Representation: Algorithms and Applications. *IEEE Access*. Retrieved from https://hal.archives-ouvertes.fr/hal-01311245/document

[25] Zitzler, E., Laumanns, M., & Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods Design*, 95–100.

Project Summary: Revisions to the Research Plan



Parameter Fine Tuning Values			
Number of Iterations	Crossover Rate	Population Size	Mutation Rate
60,000	0.4	230	0.03
90,000	0.5	300	0.05
120,000	0.6		

Figure 3: Parameter values used for algorithm optimization. All combinations of the 10 values seen above were employed for a total of 36 sets of parameters

The structure for parameter fine tuning was not developed until the implementation of MOEA-D was completed, as many different factors had to be considered. The parameter fine tuning values were strategically chosen based on previous experiments to ensure that the algorithm converged effectively. It must be stressed that parameters that were found to be optimal for a specific set of objectives and constraints may not remain effective for a different model; by testing a set of 36 different combinations of parameters, the performance of the algorithm could be refined. The parameter values tested are found in Figure 3.

The number of iterations and crossover rate have the most considerable impact on results and consequently were tested for three different values. The three values tested for number of iterations were elucidated based on literature review; it is common to complete iterations in generations, or as multiples of the population size. The number of iterations were selected in accordance with a population size of 300 at generations of 200, 300, and 400. The crossover ratio is intuitive, as a uniform crossover stipulates a crossover rate of 0.5. However, values of .4 and .6 were tested to determine if creating more concentrated or dispersed portfolios, respectively, would yield improved outcomes. A requisite for population size is that it equaled the summation of a sequence of integers starting with zero, considerably restricting potential values. 300 was viewed as an effective value for the optimization of the algorithm, but 230 was also tested in the scenario that 300 would entail a solution set with trivial differences among solutions. Finally, mutation rate generally has an arbitrary impact on results, but was tested with two values to ensure that the problem specificities did not require any specific value for convergence. While 1000 iterations may appear to be an excessive number of experiments, search algorithms can have drastically varied results due to the complexity of a large solution space; 1000 experiments will ensure that the efficacy of specific parameter values becomes evident. Parameters will only be tested on the first data set; future returns, variance and Sharpe ratio will solely be tested in the second portion to determine the relative capacity of objectives.

Though the implementation of MOEA-D was rather ambiguous at the beginning and was simply adopted from the literature, a more refined schematic was developed throughout the process of developing code. In lieu of binary values for assets, with 0 denoting that an asset is absent from the portfolio and 1 denoting that an asset is present, a real-value structure was applied to abridge the computational time of the algorithm. Assets and sectors were ordered by market capitalization and alphabetically, respectively. Assets and sectors were respectively characterized by a number ranging from 0 to 87 and 0 to 9. Real values for asset and sectors were utilized to allow the algorithm to progress through vectors without wasting efforts on trivial calculations. However, this structure did elevate the computational complexity of the algorithm by requiring nested loops. The asset and sector weights were also indicated by real values. All population members were encapsulated in subproblems that contained the current solution, weight vectors, and the indices of the neighbors [22]. The weight vectors designated the relative importance of the conflicting objectives and ensure that the algorithm produces an accurate portrayal of the Pareto efficient surface by converging at diverse solutions. These weight vectors were also utilized in Chebyshev scalarization, where the solution is compared to an ideal solution, z, to manage variables of different magnitudes. The neighborhood structure bolstered the diversity of the population by constraining recombination within the solution set to ensure that recombination does not create analogous solutions at different points within the population [22]. A uniform crossover, where each asset is chosen from either parent with identical probability was employed. A point-based crossover (eg. single-point crossover) is viewed as a poor choice, as assets with similar market capitalizations would become clustered within offspring.

One of the most critical obstacles in the implementation of metaheuristics for portfolio optimizations is the formulation of an appropriate repair heuristic that preserves the quality of a solution while ensuring adherence to constraints. Constrained multiobjective optimization required noteworthy computational intricacy to limit a hyperdimensional search space. The repair heuristic ultimately used was adapted from the work of Angosptolopolous and Mammanis [1]. First, assets were classified into discrete vectors corresponding to each sector. After sector weights were normalized as real class proportions (RCP), the proportion associated with each sector is shared among the corresponding assets as indicated in equation 8.

$$rcp(m) = \frac{C_m}{\sum_{j=1}^{m} c_j} \qquad m = 1 \ldots . M \qquad (8)$$

After asset weights were normalized, the repair heuristic sought to reduce weights of any assets that failed to meet the ceiling constraint or randomly removes an asset from the portfolio if the cardinality constraint was not satisfied. Asset weights were then redistributed in accordance with the aforementioned technique.

After the maximum number of iterations is reached, an archival method was engaged to identify solutions that exist on the Pareto Front. After an initial solution is positioned in the archive, solutions that are more fit, for one or two objectives, denoted as non-dominating solutions, were introduced as members of the archive. If a solution is further optimized for all three objectives, the solution was designated as dominating the current members. The new solution was included as a member of the archive while the dominated member, or members, is removed.