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

Abishek Ravindran

Behavioral and Social Sciences

Abstract

Decades of research have been conducted on economic models as a means of developing optimized portfolios [16]. After the creation of the Markowitz Mean-Variance model, a variety of factors have been integrated as constraints and additional objectives to bolster the applicability of the model [16]. However, the Markowitz model is adjusted to the preferences of institutional investors rather than retail investors. This paper proposes an extended version of the Markowitz Mean-Variance model that incorporates sector-weights and Google Trends data to reduce systematic risk for the quintessential investor. With the inclusion of such factors, the problem becomes non-deterministic polynomial-time hard (NP-Hard), requiring the employment of metaheuristics, or partial search algorithms, to identify adequate solutions [17]. A multi-objective evolutionary algorithm based on decomposition is utilized for its efficacy in creating a population of diverse solutions in a single run [23].

All experiments were conducted on data compiled for the S&P 100 over the period 2003-2009 to encompass a major market failure. The study was two-fold. First, parameter tuning with 36 combinations of parameters was completed over 1000 iterations with each combination to optimize the algorithm. Next, experimentation was conducted on tri-objective models, employing the two novel objectives discretely, in addition to the standard bi-objective model for 5000 iterations each. The following parameters were elucidated as effective for the specificities of the problem: population = 230, mutation rate = .03, crossover rate = .6, and number of iterations = 120,000. The behavioral objective showed promise in developing portfolios that minimized losses during the depth of the financial crisis with an out-of-sample loss of -0.00043 as opposed to the losses of -0.00051 for the bi-objective model with a p-value of .043 among the two datasets. While frequent rebalancing would be necessary to ensure favorable results due to greater variance over the out-of-sample data set, Google Trends data exhibits great promise as a supplementary objective for portfolio optimization, warranting further analysis.

1. 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.

The extended model introduced in this study considers both sector-weights as well as human behavior via Google Trends data to develop portfolios that reduce risk for investors. MOEA/D was implemented to test the efficacy of the extended computational model and was tested on Standard & Poor's 100 (S&P 100) and Dow Jones Industrial Average (DIJA) data over the time period 2003-2009 to incorporate a major market failure.

The paper continues as follows. Section 2 provides a review of the Markowitz Model and the extended model introduced in this paper. Section 3 provides a review of evolutionary algorithms and a clarification of the approach used in the implementation of MOEA/D. Section 4 provides the methodology used to refine the algorithm and an assessment of the utility of the new model. Section 5 provides the results of the experimentation and Section 6 provides concluding remarks.

2.1. Portfolio Optimization

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.

2.2 Extended Markowitz Model

This paper 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

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

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

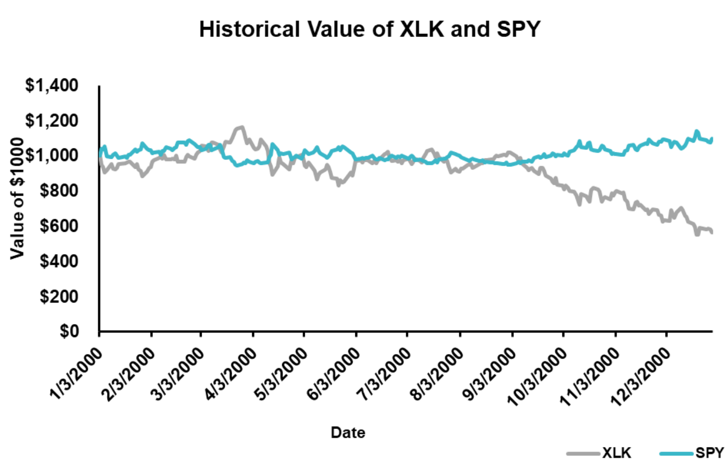

Figure 1: Historical value of 1000 Dollars invested into XLK and SPY during 2000. XLK represents the technology sector that faced dramatic losses while SPY represents an overall market-weighted index with consistent returns

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

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.

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

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

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

where l_i and u_i represent the lower-bound and upper-bound of asset weights, respectively (Adapted from Crama & Schyns [8]). Equation 3 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 4, confine the weight of assets to prevent short selling, or negative weights, and overweighting of assets, respectively. Finally, the cardinality constraint in equation 5 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.

3.1 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.

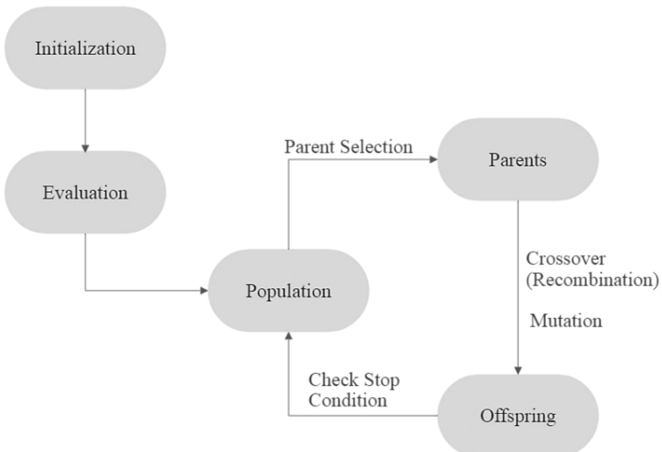

Figure 2: Coding schematic for evolutionary algorithms. Portrays population-based nature through which enhanced offspring are created and solutions are optimized (Adapted from Holland, J.H [11])

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.

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].

4 Experimentation Methodology

Experimentation was divided into two components. The first component consisted of parameter fine tuning, to ensure that parameters including crossover rate, mutation rate, and population size were optimized within the algorithm. A total of 36 different combinations of parameters were tested for 1000 iterations each. Statistical computation was then completed to determine the optimal combination of parameters. After this was completed, tests were 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 consisted of the bi-objective model without any additional objectives.

4.1 Data Set Collection

Current academic research continues to utilize historical data compiled at the end of the 20th century [8]. While this data provides an effective means of comparing different empirical models and algorithms, it fails to account for recent shifts in the global economic landscape. For this reason, data was compiled for the S&P 100 and DIJA during the time period 2003-2009. The data was 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 extended from January 1, 2003 to December 31, 2007, while the out-of-sample time period encompassed data from January 1, 2008 to December 31, 2009. To clarify, only the in-sample data was used for parameter fine tuning. In the second portion of experimentation, after a portfolio was 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 were collected via *Yahoo! Finance* and calculations including covariance were completed using *R* for statistical computing.

The constituents chosen were 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 would 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 were collected via *Google Trends*. The technique applied for selecting search queries was 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 was implemented (i.e., AA stock).

4.2 Encoding Structures

It is valuable to garner a foundational understanding of the encoding techniques employed within the algorithm to better understand the results garnered. The initial objective function is decomposed into *N* scalarized subproblems, correlating with the population size *N* [22]. Each subproblem is updated through recombination at each iteration of the algorithm. Novel solutions are 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 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 allowed the algorithm to progress through vectors without wasting efforts on trivial calculations, but this structure does 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 designate 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 utilized in Chebyshev scalarization, where the solution is compared to an ideal solution, z, to manage variables of different magnitudes. The neighborhood structure bolsters 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) was 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 requires noteworthy computational intricacy to limit a hyperdimensional search space. The repair heuristic employed here 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 are normalized, the repair heuristic seeks 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 was reached, an archival method was engaged to identify solutions that exist on the Pareto Front. After an initial solution was positioned in the archive, solutions that were more fit, for one or two objectives, denoted as non-dominating solutions, were introduced as members of the archive. If a solution was further optimized for all three objectives, the solution was designated as dominating the current members. The new solution is included as a member of the archive while the dominated member, or members, is removed.

4.3 Parameter Setting

In optimizing an evolutionary algorithm, there are assorted factors that have an extensive impact on its utility. A short description 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

The parameter fine tuning values were strategically chosen based on previous experiments to ensure that the algorithm converges 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.

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 here were employed for a total of 36 sets of parameters

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 was 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 was 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 ensure that the efficacy of specific parameter values becomes evident. Parameters were only tested on the first data set; future returns, variance were solely tested in the second portion to determine the relative capacity of objectives.

4.4 Objective Experimentation

After the algorithm was optimized, tests were conducted on the efficacy of the two objectives proposed. The control group applied a bi-objective model that did not consider any supplementary objectives. All testing was completed for 5000 iterations, due to the volatile nature of metaheuristics and the considerable differences that arise with a novel or different objective being included within a model. It is worth noting that while fitness values were 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 conveyed through the out-of-sample data sets includes valuable information.

5.1 Parameter Setting Results

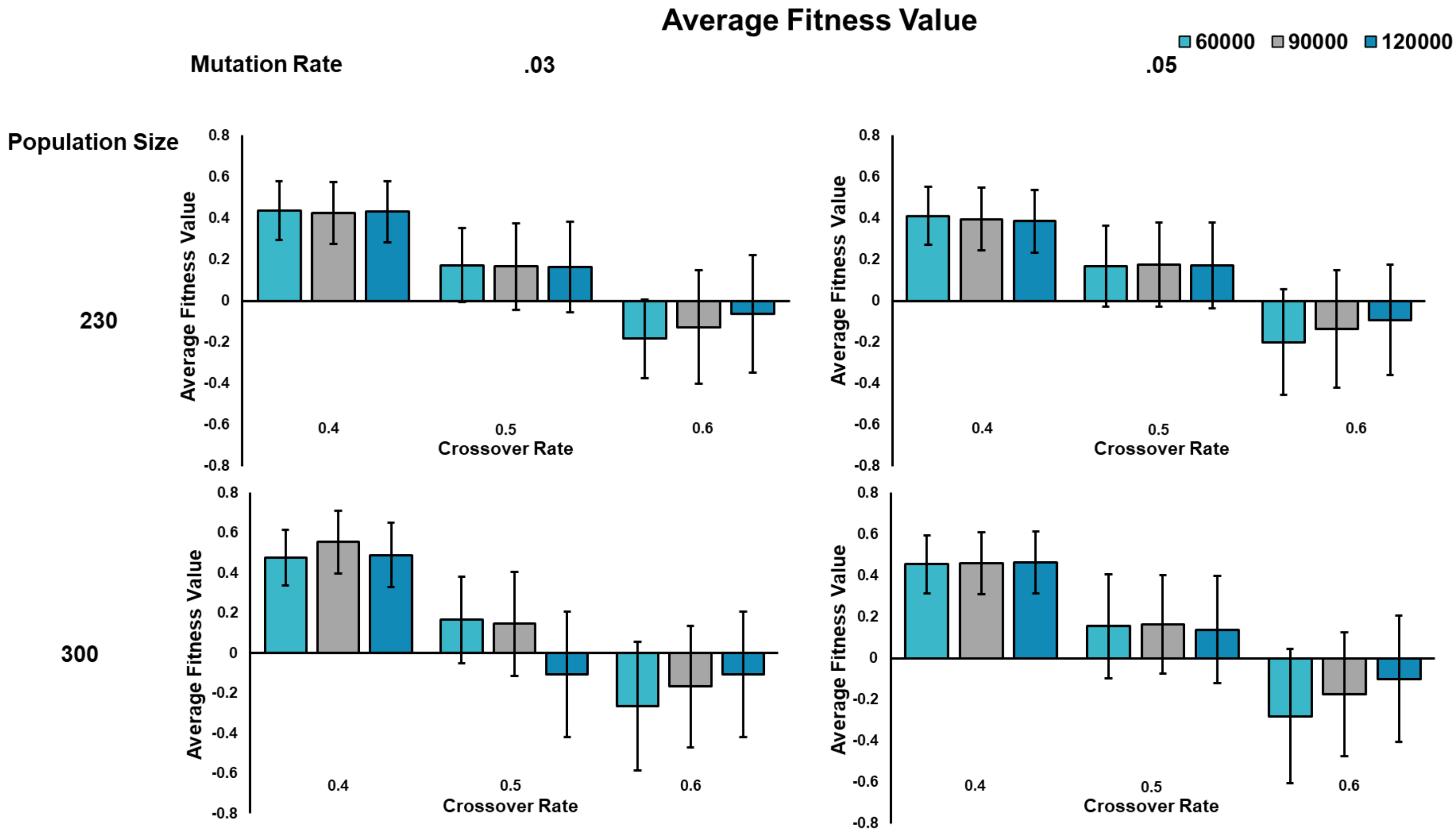

Figure 4: Average fitness values varies significantly as a function of crossover rate.
Fitness values were plotted against crossover rate for various mutation rates and population sizes. Top-left: mutation rate = .03/population size = 230. Top-right: mutation rate = .05/population size = 230. Bottom-left: mutation rate = .03/population size = 300. Bottom-right: mutation rate = .05/population size = 300

When discussing parameters, the values will be presented in the following format to maintain uniformity: {population, mutation rate, crossover rate, number of iterations}.

The average fitness values utilized were posited to be an effective means of comparison for the various parameters tested within the algorithm. However, several flaws within this metric were later ascertained. While the fitness value acted as an advantageous metric in completing calculations pertaining to the Chebyschev scalarization employed for updating solutions, it failed to account for several crucial factors in comparing parameters. First, the weighting of different objectives may have made a solution more attractive for a particular subproblem. Secondly, the average fitness value of the archive was delineated; in some cases, a larger set of solutions with more diversity that are marginally less fit would be preferred to more accurately approximate the efficient frontier. Finally, the fitness values exhibit significant discrepancies among runs of the algorithm, as noted by the large standard deviations indicated by the error bars. These standard deviations are seen even after the completion of 1000 experiments for each set of parameters. Therefore, while an algorithm with the parameters {300, .03, .4, 120,000} displays the greatest average fitness value of .4903, more nuanced details had to be analyzed before determining the optimal combination of parameters.

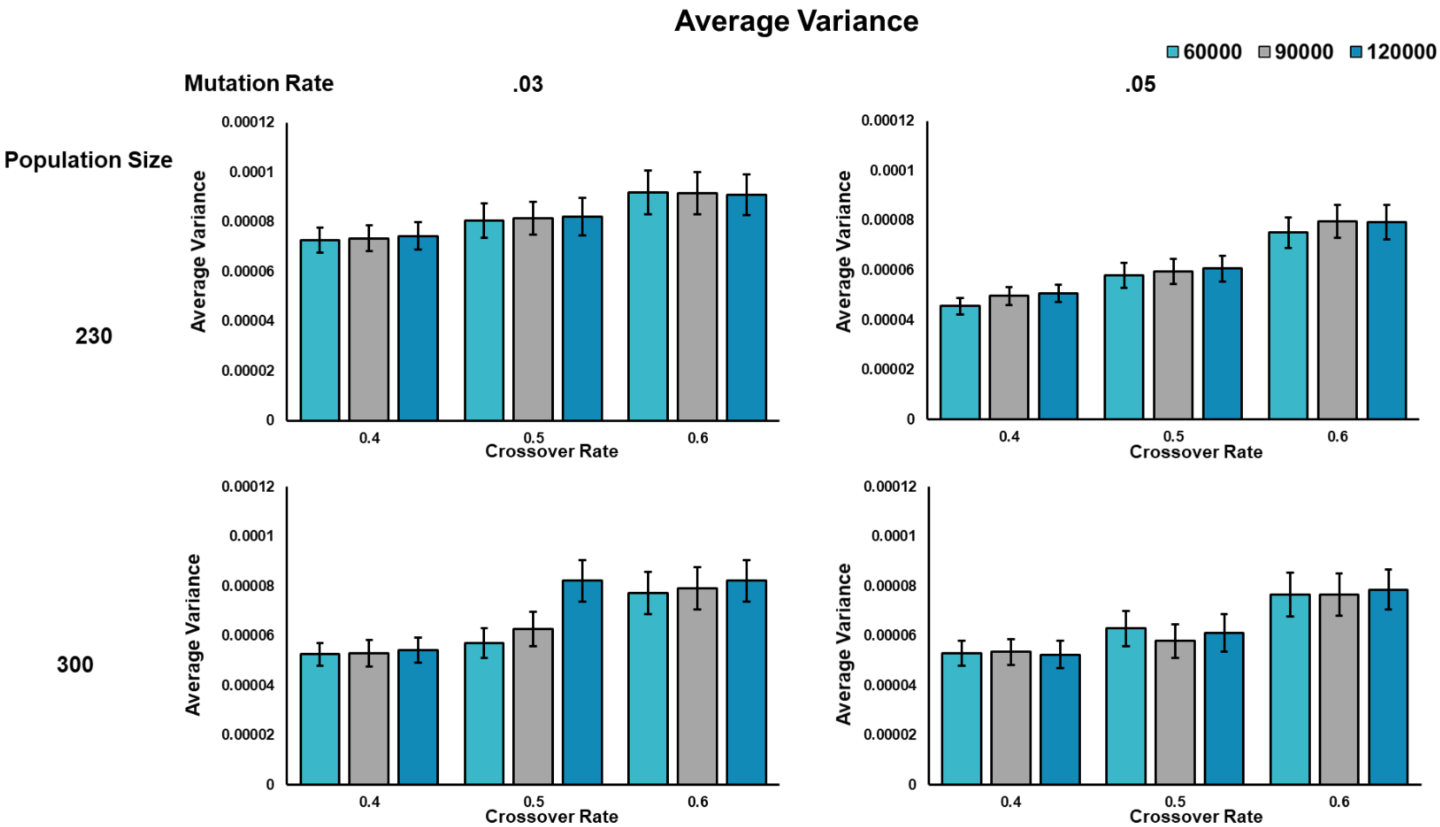

Figure 5: Average variance exhibits variation in relation to mutation rate.
Variance was plotted against crossover rate for various mutation rates and population sizes. Top-left: mutation rate = .03/population size = 230. Top-right: mutation rate = .05/population size = 230. Bottom-left: mutation rate = .03/population size = 300. Bottom-right: mutation rate = .05/population size = 300

It is worth nothing that while average return of the portfolios produced with the chosen parameters is discussed later in this section, the average variance of the solutions was more imperative to parameter selection. The model proposed in this study is not a profit-maximizing model: it seeks to minimize systematic risk for investors, sometimes at the expense of returns. Therefore, electing a model that features portfolios that minimize risk is crucial to the success of the second portion of this study.

The average variance values seen in Figure 5 show several palpable trends that illuminate the efficacy of speicifc sets of parameters. First, caeteris paribus, an upward trend is seen with an increase in crossover probability, as the average variance for crossover rates of .4 and .6 were

.00007825 and .00009565, respectively. A lower crossover probability allows for less volatile assets to be preserved as member of solutions. In addition, the smaller population size of 230 contained solutions with a generally lower average variance of .00008122 in comparison to the population size of 300 with an average variance of .00009276. A larger population exhibited a lower variance by containing an initial population with greater diversity before non-dominating solutions were selected.

Several anomalies were seen in this portion of the data. While one would anticipate that a greater number of iterations would allow the algorithm to converge towards solutions with lower variance, the results depict the opposite. While this could be attributed to the algorithm itself, it is most likely due to a poor weighting schematic of the conflicting objectives in fitness value calculations. Essentially, the algorithm would hold preference to either return or sector tracking in selecting solutions instead of variance. However, the solutions produced with the parameters {300, .05, .5, 90,000} contradict this trend with an average variance of .000082296.

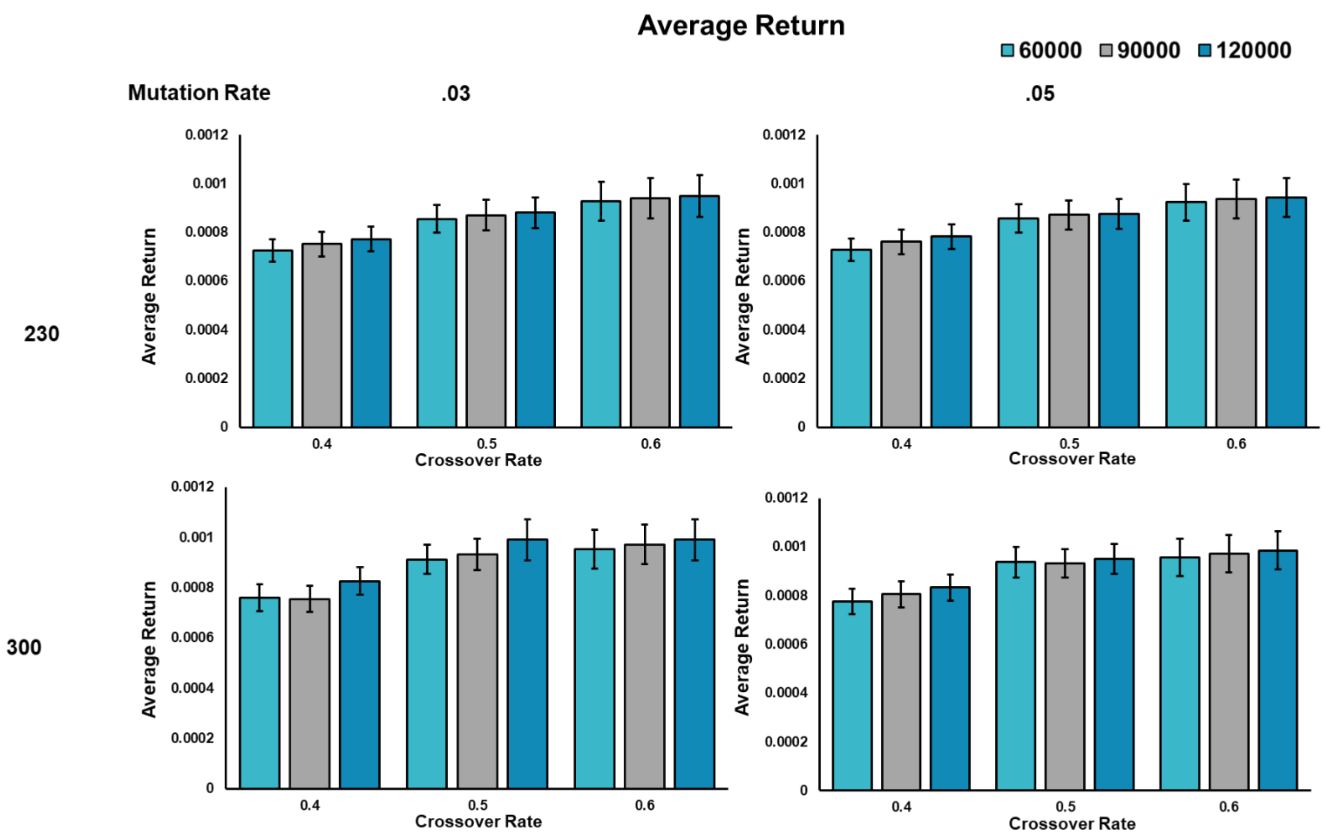

Figure 5 Average return exhibits positive correlation in relation to crossover rate.
Average return was plotted against crossover rate for various mutation rates and population sizes. Top-left: mutation rate = .03/population size = 230. Top-right: mutation rate = .05/population size = 230. Bottom-left: mutation rate = .03/population size = 300. Bottom-right: mutation rate = .05/population size = 300

The data conveyed several clear trends within the average return values. It must be understood that the return values portrayed here, and in later portions of the paper, are computed as logarithmic returns, often used to neglect the impact of compounding. To determine the annualized return, the values seen in this graph must be multiplied by 255, or the number of trading days in a year. As the crossover rate became larger and the number of iterations was increased, the return of the portfolios portrayed an upward trend. This also supports the earlier conjecture that the algorithm was promoting a different objective, in this case return, above variance. While it may be concerning that the algorithm was not emphasizing variance in a risk-minimizing model, the problem depicted here is frequent within evolutionary algorithms and was remedied by a change in scale factors for the second phase of testing [23]. The algorithm did not produce portfolios with returns that significantly deviated from the mean return, as is evident in the average standard deviation in return among all experiments of .0000842885.

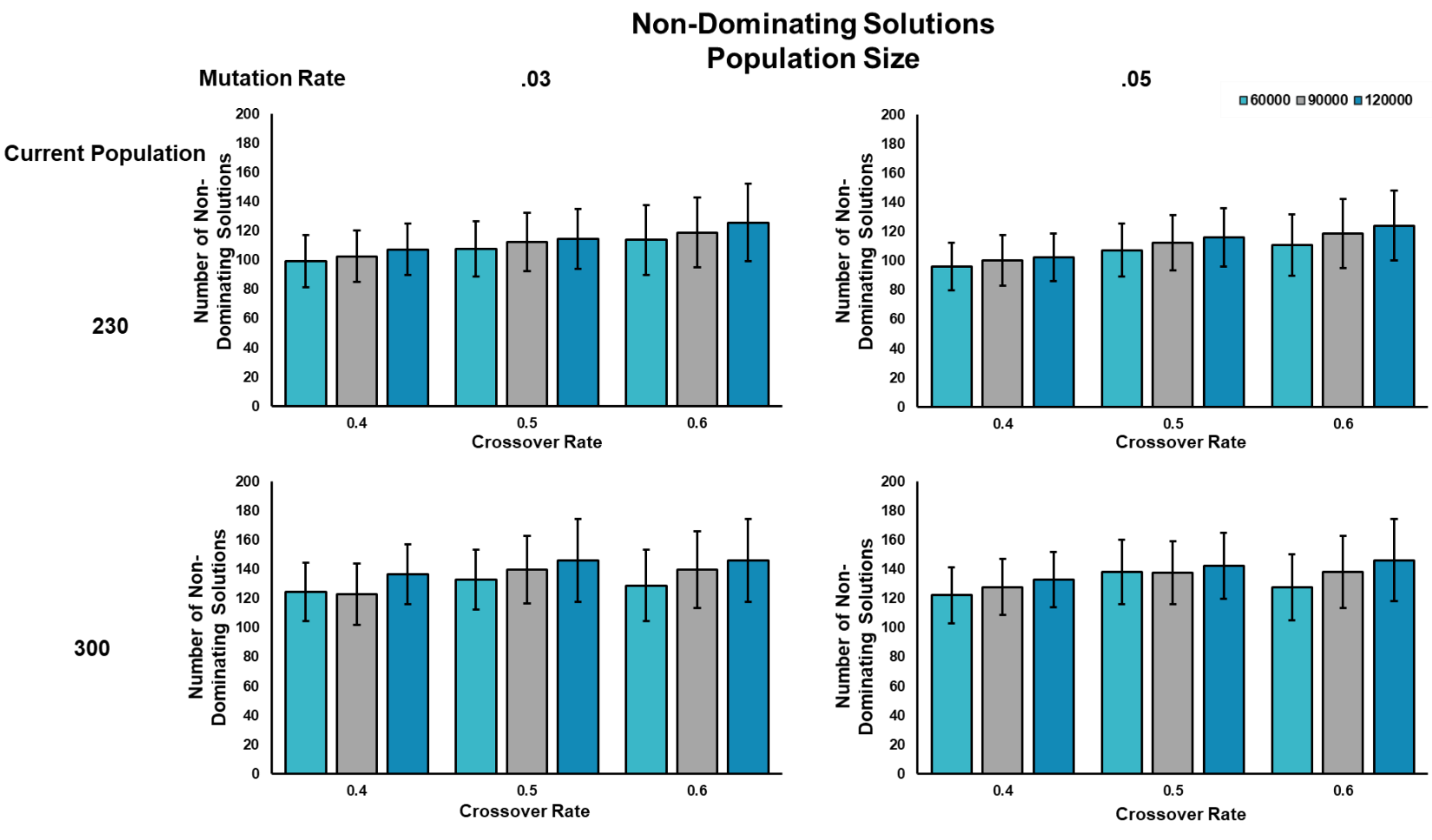

Figure 6: Number of non-dominating solutions shows variation with changes in crossover rate. Number of non-dominating solutions conveys the ability of the algorithm to converge at a population containing more diverse solutions and was plotted against crossover rate for various population sizes and mutation rates. Top-left: mutation rate = .03/population size = 230. Top-right: mutation rate = .05/population size = 230. Bottom-left: mutation rate = .03/population size = 300. Bottom-right: mutation rate = .05/population size = 300

The number of non-dominating solutions portrays the approximation of the efficient frontier created by the algorithm and the algorithm's ability to avoid saturation. Differences among the different population sizes is self-evident. A greater population size would be predisposed towards producing a set of solutions that are non-dominating in relation to one another. Two variables portrayed a clear trend with population size: crossover rate and iteration number. The greater the crossover rate and number of iterations, the greater the number of solutions found in the non-dominating solution set. A greater crossover rate would explore a larger portion of the potential solution set, while a greater number of iterations would incline the algorithm to seek out more solutions coinciding with the Pareto Front. It should be noted that the number of solutions was more consistent with a larger population size, as the average standard deviations for 230 and 300 members were 7.6925 and 8.4337, respectively. A compelling metric

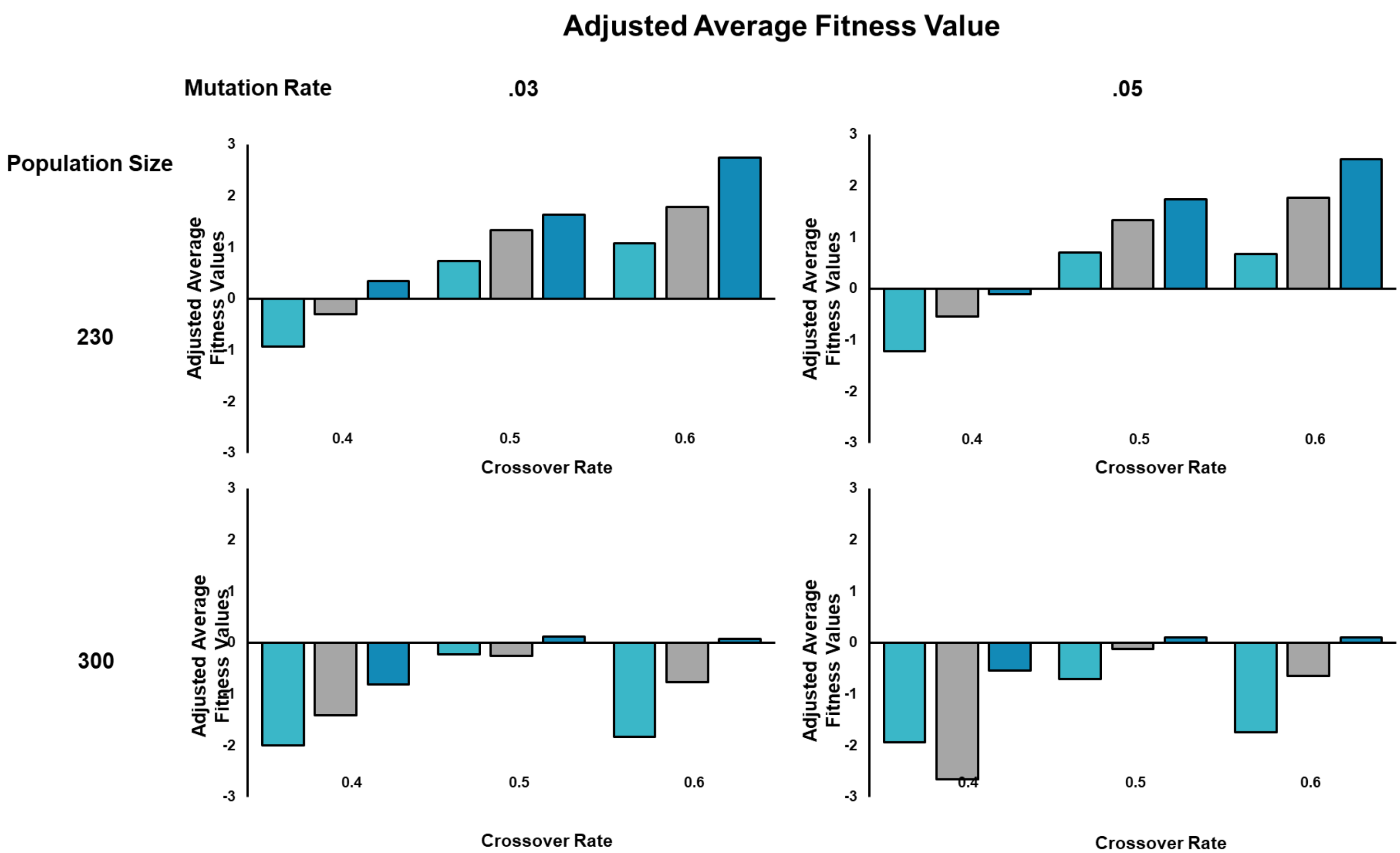

Figure 7: Adjusted fitness values illuminates differing efficacy of parameters.
Adjusted fitness values were calculated based on z-scores and were plotted against crossover rate for various mutation rates and population sizes. Top-left: mutation rate = .03/population size = 230. Top-right: mutation rate = .05/population size = 230. Bottom-left: mutation rate = .03/population size = 300. Bottom-right: mutation rate = .05/population size = 300

that was proposed was the percentage of population members that occupied the non-dominating solution set for the two population sizes, as they could not be compared directly. These values were .48 and .45, characterizing a population size of 230 as, to some extent, more effective at producing non-dominated solutions. This is most likely due to negligible differences among solutions within the 300-member population that made the larger population unnecessary.

The values for sector tracking are not displayed graphically in this portion of the paper. The sector tracking integrated well into the model, as the value of overweighting for sectors was .0408 on average within experimentation with the objective. as opposed to an average of .2563 in preliminary testing without the objective.

While a great deal of analysis was conducted on the parameter setting data, selecting parameters still proved problematic. In order to balance the conflicting objectives and choose parameter values solely based on the values for risk, return, and non-dominating solution size, an aggregation of z-scores was implemented, seen in Figure 7. The z-scores were taken based on the average for each respective metric on all experiments conducted, and the z-score for variance was negated, as a lower z-score would imply lower risk.

Based on these calculations, the following set of parameters was chosen for the second phase of experimentation due to its higher aggregate z-score: {230, .03, .6, 120,000}.

5.2 Objective Experimentation

While seven metrics were calculated for the 5000 experiments conducted with the three test groups, two posed the most utility in determining the efficacy of the objectives used. While the metrics used for parameter setting (average fitness value, average return, average variance, average sector tracking/behavioral risk, and average non-dominating population size) were still calculated in this portion of the study, they solely operated as benchmarks to ensure that the algorithm was converging correctly and were not necessarily a means of comparison.

However, when analyzing the performance of a portfolio, the time series immediately following the dataset with which portfolios are produced provides a realistic approach to examining an augmented or modified model. Therefore, future returns and variance were analyzed most thoroughly in this portion of the study.

The variance of the portfolios varied greatly among objectives. While the model with the Google Trends objective was posited to exhibit the lowest variance, it portrayed the greatest with an average variance of .0000669. While this was unanticipated, it can be attributed to the nature of the Standard & Poor's index; during the financial crisis assets moved indiscriminately upon the slightest shift in markets or information. However, the portfolios produced with the sector-weight objective demonstrated the smallest variance. This is consistent with earlier suppositions, as a greater variance would be exhibited by portfolios that contained assets concentrated among a few classes. However, variance is only a metric employed to select assets that are not expected to have shortfall in the future; while it is still significant, it is not as important as it is on the out-of-sample data set. This point will be explored further when discussing the future return values seen. While the variance of the Google Trends data was the greatest, it did have the lowest standard deviation in variance among experiments of .0000488896.

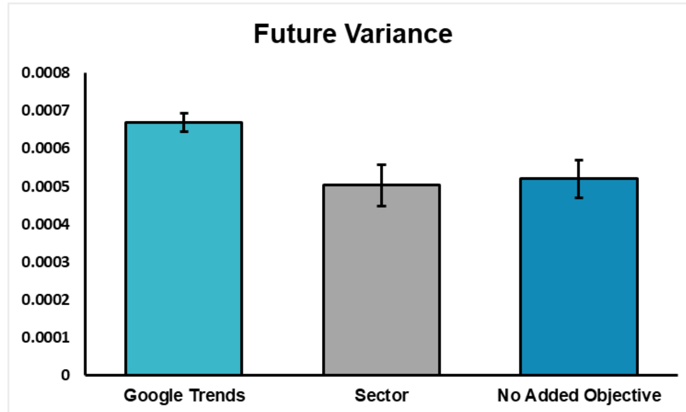

Figure 8: Projected variance elucidates efficacy of sector-weight objective in reducing risk. Tests were conducted on out-of-sample data set for 5000 iterations.

Contradicting expectations, the portfolio model incorporating Google Trends data had the greatest future return, which in this case was the lowest amount of losses, with an average return of -0.000435. In annualized terms, this is a loss of 11.08 percent in comparison to the losses of 13.82 percent seen with portfolios produced with the bi-objective model. As for the difference in variance, rebalancing in consistent intervals would allow one to have a portfolio that minimizes risk and would provide consistent returns over their investment horizon. While it is premature to characterize Google Trends data as a leading indicator, it does show some correlation to market movement that could provide utility in creating portfolios that are effective for recessions or market failures.

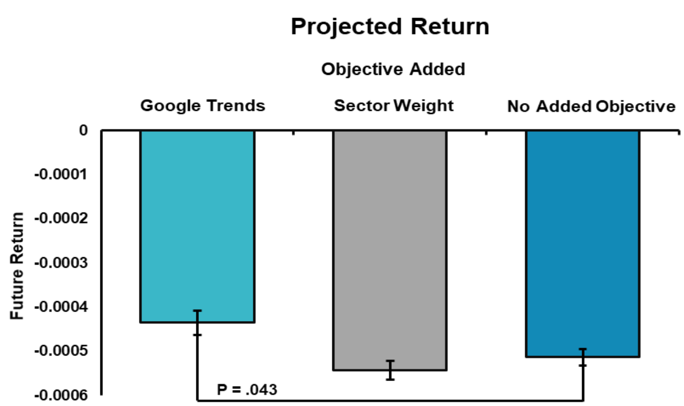

Figure 9: Projected losses portfolios illuminates efficacy of Google Trends in minimizing risk during market downturns. Tests were conducted on out-of-sample data set for 5000 iterations. P-value of .043: portrays statistical significance of difference

As can be seen in the data, the Google Trends data does display efficacy in establishing a portfolio that can reduce losses in a market downturn. The model augmented with the sector-weight objective failed to supply similar results. This can be attributed to a multitude of reasons, but primarily due to the market-weighted strategy employed that significantly constricts several sectors.

6 Conclusions

While the values of the sector weight and Google Trends data are not explored comprehensively in this paper, they are arbitrary. What was important to analyze was the impact that these objectives had on other factors including risk and return.

The challenge that arises with such data is that there is still an inherent tradeoff between the conflicting objectives, and no single objective or model can be declared as more effective. However, while variance is more important in the in-sample-data set towards selecting solutions, return is more important in the out-of-sample data set to ensure financial success. Currently, a sector-weight constraint endures as a more effective option for constraining specific sectors of the portfolio, as the sector-weight objective yielded greater losses than the bi-objective model. However, the Google Trends data objective shows great promise in portfolio optimization.

While the Google Trends data objective did not establish the risk-minimizing portfolio in computational terms, systematic risk is not something observed in measures such as covariance. Therefore, its results in producing greater future return portrays its relative efficacy. Finally, while the difference of 0.0107 in daily log returns may seem arbitrary, this is a difference of 2.74 percent in annual terms.

Future work can investigate the effect of different search query patterns as well as a more granular time scale that would be more sensitive to minor changes in search query patterns on the Google Trends objective. Additionally, to reduce concerns associated with biases in back testing, a neural network can be employed to determine the efficacy of Google Trends data more holistically.

This study proposes an extended model for portfolio optimization, showing the performance of Google Trends as a metric to be incorporated as an additional objective. As such, the economic community is implored to conduct further study on Google Trends within metaheuristics.

Works Cited

[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.