Multiobjective Financial Portfolio Design: A Hybrid Evolutionary Approach

Raj Subbu, Piero P. Bonissone, Neil Eklund, Srinivas Bollapragada, Kete Chalermkraivuth

***Abstract*—A principal challenge in modern computational finance is efficient portfolio design – portfolio optimization followed by decision-making. Optimization based on even the widely used Markowitz two-objective mean-variance approach becomes computationally challenging for real-life portfolios. Practical portfolio design introduces further complexity as it requires the optimization of multiple return and risk measures subject to a variety of risk and regulatory constraints. Further, some of these measures may be nonlinear and nonconvex, presenting a daunting challenge to conventional optimization approaches. We introduce a powerful hybrid multiobjective optimization approach that combines evolutionary computation with linear programming to simultaneously maximize these return measures, minimize these risk measures, and identify the efficient frontier of portfolios that satisfy all constraints. We also present a novel interactive graphical decision-making method that allows the decision-maker to quickly down-select to a small subset of efficient portfolios. The approach has been tested on real-life portfolios with hundreds to thousands of assets, and is currently being used for investment decision-making in industry.**

***Index Terms*—Evolutionary algorithms, linear programming, multiobjective decision-making, Pareto sorting, target objectives, portfolio optimization.**

I. Introduction

Modern Computational Finance has its historical roots in the pioneering portfolio theory of Markowitz [1]. This theory is based on the assumption that investors have an intrinsic desire to maximize return and minimize risk on investment. Mean or expected return is employed as a measure of return, and variance or standard deviation of return is employed as a measure of risk. This framework captures the risk-return tradeoff between a single linear return measure and a single convex nonlinear risk measure. The solution typically proceeds as a two-objective optimization problem where the return is maximized while the risk is constrained to be below a certain threshold. The well-known risk-return efficient frontier is obtained by varying the risk target and maximizing on the return measure. This framework however is unsuitable for practical portfolio design where it is important to consider measures beyond the mean and variance of returns, as portfolio managers are also concerned with measuring and optimizing the risk of losing all or most of a portfolio's value due to catastrophic events. In a normal situation, a portfolio's value fluctuates around its mean due to market volatility and other risk drivers. However, a portfolio may lose a significant amount of its value from a low-probability-high-impact event. A suitable measure—Value at Risk (*VaR*), which captures this risk aspect, is typically nonlinear but also nonconvex. Portfolio managers may also deal with an optimization problem that involves multiple return measures, as while some may be concerned with accounting incomes as well as economic returns, others may be concerned with long-term as well as short-term returns.

While return measures are typically linear, risk measures are typically nonlinear and often nonconvex. In a portfolio design problem with strictly linear objectives and constraints, a linear programming solution approach is the best fit. However, if one or more of the objectives are nonlinear, alternative approaches are required. For high-dimensional portfolio design problems with linear constraints where the return measure is linear, and the risk measure is nonlinear but convex, Chalermkraivuth et al. [2] have recently developed a novel *Sequential Linear Programming* algorithm for rapidly identifying the efficient frontier. The method proceeds by initially solving a relaxation of the problem without regard to risk, and later sequentially applying tighter linear constraints obtained by the linearization of the nonlinear convex risk function to generate the efficient risk-return frontier. At each sequential step, the return function is maximized subject to satisfaction of linear constraints. However, when one or more of the measures are nonconvex, an alternative optimization approach is required.

We present a new hybrid evolutionary multiobjective portfolio optimization algorithm that integrates evolutionary computation with linear programming for portfolio design problems with multiple measures of risk and return, where the measures may be nonlinear and nonconvex. We also introduce a novel interactive graphical decision-making method that allows a decision-maker to quickly down-select to a small subset of efficient portfolios via iterative constrained selections of portfolios represented as points projected in two-dimensional graphs over the combinations of the various return and risk measures utilized. Our portfolio design approach is implemented in a web-based environment that is currently in use at General Electric Asset Management, General Electric Insurance, and Genworth Financial.

The rest of this paper is organized as follows. In Section II, we present background on multiobjective optimization, and the multiobjective portfolio optimization problem. In Section III, we describe our hybrid evolutionary portfolio design (optimization and decision-making) architecture and present

The authors are with General Electric Global Research, Schenectady, NY 12309, USA. The ideas described in this paper are covered by several pending US Patents. The authors would like to acknowledge the helpful foundational and technical support of the following individuals towards the realization of the PROFITS portfolio optimization project: Michael Clark, Bernhard Scholz (both of who are with General Electric Global Research), John Deaton, Walter Neeves, and John Murdzek (all of who are with Genworth Financial). Special thanks to Naresh Iyer of General Electric Global Research for introducing us to interactive graphical tools for multi-objective decision-making.

results from a highly constrained real-life portfolio design case with over fifteen hundred search dimensions. We conclude in Section IV.

II. Multiobjective Portfolio Optimization

A. Multiobjective Optimization

Most real-world optimization problems have several, often conflicting objectives. Therefore, the optimum for a multiobjective problem is typically not a single solution—it is a set of solutions that trade-off between objectives. This concept was first generally formulated by the Italian economist Vilfredo Pareto in 1896 [3], and it bears his name today. A solution is Pareto optimal if (for a maximization problem) no increase in any criterion can be made without a simultaneous decrease in any other criterion. The set of all Pareto optimal points is known as the *Pareto frontier* or alternatively as the *efficient frontier*.

Traditional mathematical programming-based optimization methods for multiobjective problems abound [4]. However, these techniques require multiple executions to identify the Pareto frontier, and may in several cases be highly susceptible to the shape or continuity of the Pareto frontier, restricting their wide practical applicability. A recent mathematical programming-based multiobjective optimization technique that is able to identify nonconvex Pareto frontiers is presented in [5]. However, this approach does not guarantee identification of the true Pareto frontier, and only generates a good approximation to it, as is generally possible for most real-life applications.

Evolutionary algorithms have received much attention for use in numerous practical single objective optimization and learning applications. The area of evolutionary multi-objective optimization has grown considerably, starting with the pioneering work of Schaffer [6]. Since evolutionary algorithms inherently work with a population of solutions, they are naturally suited for extension into the multi-objective optimization problem domain, which requires the search for and maintenance of multiple solutions during the search. This characteristic allows finding an entire set of Pareto optimal solutions in a single execution of the algorithm. Additionally, evolutionary algorithms are less sensitive to the shape or continuity of the Pareto front than traditional mathematical programming-based techniques. In the past decade, the field of evolutionary multi-objective decision-making has been significantly energized, due in part to the multitude of immediate real-life applications in academia and industry. Several researchers have proposed evolutionary multi-objective optimization techniques, overviews, and their comparisons (e.g. [7]-[8]).

1) Relevant Work in Evolutionary Portfolio Optimization

An early approach to evolutionary portfolio optimization was presented in [9]. Chang et al. [10] present a comparison of Tabu Search, Simulated Annealing, and Evolutionary Algorithms on the Markowitz mean-variance portfolio optimization problem. Ehrgott et al. [11] use neighborhood search, Tabu Search, Simulated Annealing, and a Genetic Algorithm to optimize a mixed-integer (due to the constraints used) portfolio optimization problem with objectives aggregated via user-specified utility functions. This, and the previous evolutionary portfolio optimization approaches have solved the inherently multiobjective optimization problem using single objective optimization techniques. Elicitation of relative weights or utility functions to aggregate the multiple objectives can often be very difficult, and restricts flexibility to changing decision-maker preferences. Recently, Streichert et al. [12] have proposed an evolutionary multiobjective optimization approach to a Markowitz mean-variance portfolio optimization problem. Though in this approach the authors have attempted to solve a portfolio optimization problem with realistic constraints, they have optimized over only two objectives—one linear return objective, and one convex nonlinear risk objective. In contrast to these prior attempts, our approach is able to handle multiple measures of return and risk in a truly multiobjective optimization sense. Recently, metaheuristic and hybrid evolutionary techniques have been applied to the portfolio optimization problem [13]-[14]. In contrast to all these prior approaches, importantly, we present in addition a systematic approach for portfolio decision-making, which has thus far been lacking in the literature.

B. Multiobjective Portfolio Optimization Problem

The goal of portfolio optimization is to manage risk through diversification and obtain an optimal risk-return tradeoff. Risk measures play a crucial role in portfolio optimization. In order to characterize the investor's risk objectives and capture the potential risk-return tradeoffs, risk measures are used to quantify various aspects of portfolio risk.

For asset-liability management (ALM) applications, we use surplus variance as a measure of risk. We compute portfolio variance using an analytical method based on a multifactor risk framework [15]-[16]. In this framework, the value of a security can be characterized as a function of multiple underlying risk factors. The change in the value of a security can be approximated with the changes in the risk factor values and risk sensitivities to these risk factors. The portfolio variance equation can be derived analytically from the underlying value change function.

In asset-liability management applications, the portfolios have assets and liabilities that are affected by the changes in common risk factors. Since a majority of the assets are fixed-income securities, the dominant risk factors are interest rates. In ALM applications, in addition to maximizing return or minimizing risk, portfolio managers are constrained to match the characteristics of asset portfolios with those of the corresponding liabilities to preserve portfolio surplus due to interest rate changes. Therefore, the ALM portfolio optimization problem formulation has additional linear constraints that match the asset-liability characteristics when compared with the traditional Markowitz model. We use the following ALM portfolio optimization formulation:

Maximize	Portfolio Expected Return	
Minimize	Surplus Variance	
Minimize	Portfolio Value at Risk	
Subject to:	Duration mismatch ≤ target_1	(1)
	Convexity mismatch ≤ target_2; and	
	Linear portfolio investment constraints	

From the many available metrics, we decided to use *Book Yield*, *Variance*, and *Simplified Value at Risk* (*SVaR*) as the respective metrics for Portfolio Expected Return, Surplus Variance, and Portfolio Value at Risk. *Portfolio Book Yield* represents its accounting yield to maturity and is defined as:

$$BookYield_P = \frac{\sum_i BookValue_i x BookYield_i}{\sum_i BookValue_i} \tag{2}$$

Portfolio Variance is a measure of its variability and is defined as the second moment of its value change ΔV:

$$\sigma^2 = E\left[(\Delta V)^2\right] - E\left[(\Delta V)\right]^2 \tag{3}$$

Portfolio Simplified Value at Risk is a measure of the portfolio's catastrophic risk and is defined in detail in [17]. These metrics define the 3-D optimization space. Now, let us analyze its constraints. The change in the value ΔV of a security can be approximated by a second order Taylor series expansion given by:

$$\Delta V \approx \sum_{i=1}^{m} \left(\frac{\partial V}{\partial F_i}\right)\Delta F_i + \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\left(\frac{\partial^2 V}{\partial F_i \partial F_j}\right)\Delta F_i \Delta F_j \tag{4}$$

The first- and second-order partial derivatives in this equation are the risk sensitivities, i.e. the change in the security value with respect to the change in the risk factors F_i. These two terms are typically called *delta* and *gamma*, respectively [16]. For fixed-income securities, these measures are *duration* and *convexity*. The duration and convexity mismatches, which constrain our optimization space, are the absolute values of the differences between the effective durations and convexities of the assets and liabilities in the portfolio respectively. Though they are nonlinear (because of the absolute value function), the constraints can easily be made linear by replacing each of them with two new constraints that each ensure that the actual value of the mismatch is less than the target mismatch and greater than the negative of the target mismatch respectively. The other portfolio investment constraints include asset-sourcing constraints that impose a maximum limit on each asset class or security, overall portfolio credit quality, and other linear constraints.

III. Architecture and Results

A. *Multi-Criterion Decision Making*

As in many other real-world problems, optimal portfolio selection is characterized by multiple objectives, measuring different types of return and risk, which need to be optimized or at least satisfied simultaneously. The decision maker (DM) needs to search for the non-dominated solutions in this objective space, while aggregating his/her preferences over multiple criteria. For instance, the DM might want to find a portfolio that minimizes risk (measured by variance, value at risk, credit risk, etc.) and maximizes return (measured by book yield, market yield, duration-weighted market yield, etc.). Since these objectives cannot be satisfied simultaneously, we need to accept tradeoffs. We need to establish a search method driven by multi-objective functions to find non-dominated solutions, a multi-criteria tradeoff policy, and a structure for such a process. These activities are not independent.

We can distinguish among three models of decision-making, as characterized in [18]:

- *Perform multi-objective aggregation and decision before search.* This approach reduces the dimensionality of the problem by adding more ordering information. This is a well-know approach that could generate sub-optimal solutions when the solution space is not convex.
- *Perform search before multi-objective aggregation*. This approach postpones tradeoffs until large numbers of inferior, dominated solutions are eliminated and the efficient Pareto Front has been identified.
- *Iteratively integrate search and multi-objective aggregation and decision.* This third approach starts with a multi-criteria search that provides the DM with a preliminary idea of possible tradeoffs. The DM can then make multi-criteria decisions, limiting the search space dimensionality. A new search is then performed in this region of the solution space.

In our example, we will follow the third case, and cast the optimal portfolio selection as an iterative integration of search and multi-objective aggregation and decision. Figure 1 shows a block diagram of the architecture used in this approach.

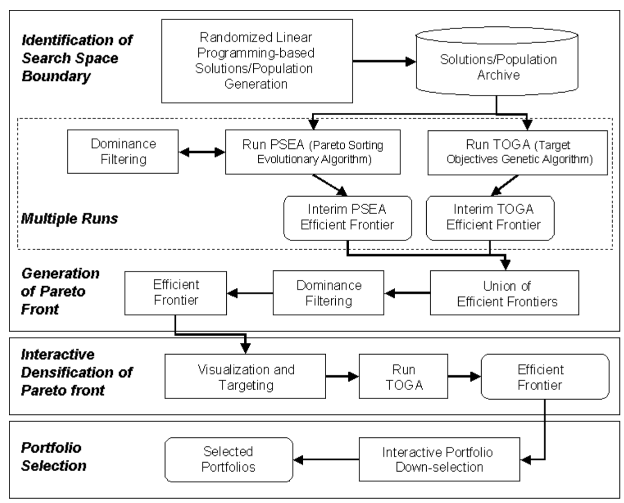

Fig 1. Block Diagram of architecture.

B. *Portfolio Optimization: Search for the Efficient Frontier*

The dashed block in Fig. 1 shows the search components of our approach. We decided to generate an initial Pareto front in a three-objective space defined by *Book Yield*, *Variance*, and *Simplified Value-at-Risk* (*SVaR*), using a Pareto Sorting Evolutionary Algorithm (PSEA). We initialized the PSEA with a Randomized Linear Programming (RLP) algorithm, which stochastically identifies a sample of the boundaries of the search space by solving thousands of randomized linear programs.

1) *Randomized Linear Programming (RLP)*

The key idea of the RLP is the generation of the initial population for the PSEA with potential solutions that would satisfy the constraints defined in the problem formulation. A linear programming algorithm is typically used to find the optimal decision vector $\mathbf{x}$ that solves the problem $\{\max \mathbf{w}^T\mathbf{x} : \mathbf{Ax} \le \mathbf{b}\}$, where $\mathbf{w}$ is a pre-specified coefficient or weight vector, $\mathbf{A}$ is a matrix of linear constraints and $\mathbf{b}$ is a vector of constants corresponding to the linear constraints. The space $\mathbf{Ax} \le \mathbf{b}$ is a convex polytope P. Due to the geometrical nature of the problem and the linear programming algorithm, given a weight vector $\mathbf{w}$, an extreme corner vertex of the feasible space defined by $\mathbf{Ax} \le \mathbf{b}$ is always identified as the optimal decision vector. If a different weight vector $\underline{\mathbf{w}}$ is used, the optimal decision vector will be some other extreme corner vertex $\underline{\mathbf{x}}$. We use this property to our advantage to sample the space of extreme corner vertices, by solving multiple linear programs with randomly specified weight vectors—the expectation being that a randomly specified weight vector will lead us to a random corner vertex that is optimal for that given random weight vector. This process of random weight vector specification and solution of the ensuing linear program is repeated thousands of times, depending on the search dimensionality of the portfolio optimization problem. These solutions are archived for future use by the evolutionary search algorithms described later. Our approach is similar in principle to the *pre-process phase*, proposed by Kubalik and Lazanski [19]. In our case, we utilize the RLP to stochastically sample the boundaries of the search space, so the evolutionary search within that space can proceed without concern of constraint satisfaction issues.

2) Pareto Sorting Evolutionary Algorithm (PSEA)

The Pareto Sorting Evolutionary Algorithm (PSEA) works with small, tractable population sizes and uses an *archive* that saves good non-dominated solutions found at any generation for future reference and use. In this way, the search algorithm navigates the search space using only a small population size, but commits any good solutions found to memory. However, there is no guarantee that a non-dominated solution found at a certain generation would also be globally non-dominated with respect to the existing solutions in the archive. Therefore, the final step of the PSEA is dominance filtering of the solutions in the archive to generate the global Pareto frontier.

Though the *archive* method is an efficient diversity preservation technique and is helpful for preservation of diversity across the entire Pareto frontier, it is not sufficient to ensure diversity in the populations during search. Diversity preservation in the populations themselves is an important aspect of an efficient search. In addition, diversity preservation in the populations has a significant impact on the diversity of the solutions in the archive. Diversity in populations may be enabled via three methods: *(i)* incorporation of the *crossover* operator, where two parent solutions are mathematically combined to yield two offspring solutions; *(ii)* incorporation of *new random solutions* in each generation of the search; and *(iii)* incorporation of a *non-crowding filter*, which is able to filter out solutions in a population that are heavily clustered and is able to reduce the propensity for clustering that causes reduced diversity. In our approach, we use all three methods, as exemplified by steps 5, 10, and 8 in Figure 2. This figure, which illustrates the PSEA operational process, is the reference for the step-by-step description of the PSEA below.

In Step 1, an initial population of cardinality n is created by randomly drawn solutions. In the context of our problem, the solutions constituting the initial population are randomly drawn from a *solutions archive* of feasible corner vertices of the linearly constrained convex problem-specific search space. Typically, the size of such a population would be in range of 50 to 100 solutions. In Step 2, the initial population is passed through a dominance filter, and a non-dominated sub-set of cardinality $(\mu < n)$ is identified (Step 3). This non-dominated sub-set is also committed to the *non-dominated solutions archive* in Step 4.

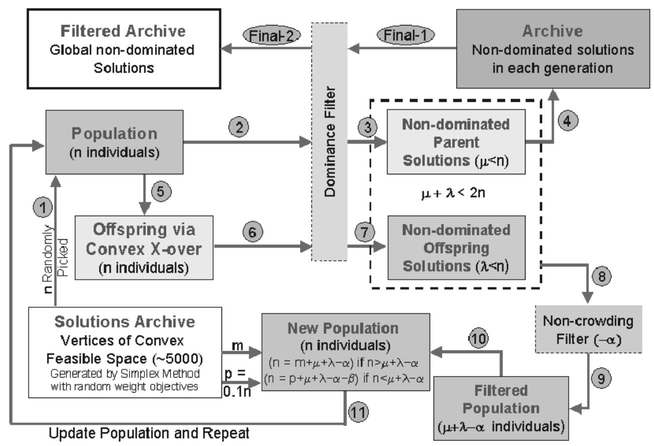

Fig 2. PSEA Block Diagram.

In Step 5, $n/2$ randomly matched pairs of parent solutions are combined to create n *offspring* solutions. In the context of portfolio risk optimization, since the feasible space is convex, and we already include a sample of the bounds of the convex feasible space via points from the solutions archive, interpolative convex combination of two parent solutions to yield two offspring solutions is a suitable crossover technique. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, the two offspring vectors generated via convex crossover are $w\boldsymbol{x} + (1-w)\boldsymbol{y}$, and $(1-w)\boldsymbol{x} + w\boldsymbol{y}$, where w is a real random number in the space [0 1]. The n offspring solutions are passed through a dominance filter in Step 6 that identifies the non-dominated sub-set of cardinality $(\lambda < n)$ in Step 7.

In Step 8, we combine the two non-dominated solution sub-sets of cardinality $(\mu+\lambda < 2n)$, and pass them through a *non-crowding filter*, that removes a smaller sub-set of α solutions that are tightly clustered. The result of this non-crowding operation is a reduced solution sub-set of cardinality $\mu+\lambda-\alpha$ in Step 9. This non-crowding filter identifies regions that are heavily clustered and drops solutions from those dense clusters. In Step 10, a new population of n individuals is created. If $n = \mu+\lambda-\alpha$, then the filtered population from Step 9 becomes the new improved population. In case $n > \mu+\lambda-\alpha$, m individuals from the solutions archive are randomly drawn such that $n = m+\mu+\lambda-\alpha$. The inclusion of these m individuals from the solutions archive injects new points into the population enhancing its diversity. In case $n < \mu+\lambda-\alpha$, we are faced with a problem of discarding some solutions from the set of $\mu+\lambda-\alpha$, and at the same time promoting diversity via the injection of new points. This is achieved in two sub-steps. In the first sub-step, *(n-p)* solutions are randomly selected from the $\mu+\lambda-\alpha$ solutions. p is a number that is one tenth the magnitude of n. In other words, β solutions are randomly discarded from the set of $\mu+\lambda-\alpha$ solutions. To adjust the cardinality of the new population to n, p solutions are randomly injected from the solutions archive. The inclusion of these p individuals from the solutions archive injects new points into the population enhancing its diversity.

In Step 11, the new population replaces the previous population, and the evolutionary process is continued until convergence is achieved, or allocated computational cycles are exhausted. Finally, at the conclusion of the evolutionary search, the non-dominated solutions archive is passed through a dominance filter (Final-1) to yield the global near-Pareto-optimal frontier (Final-2).

3) Fast Dominance Filter

Dominance filtering is a key component of our approach to multi-objective portfolio optimization. Given a set of M vectors to be partitioned into dominated and non-dominated subsets, and given N objectives, the computational complexity of the typical partitioning process is $O(N M^2)$. For large M and $N > 2$, the time required to partition the set of M vectors grows rapidly. Since the PSEA presented earlier is dependent on its ability to repeatedly and rapidly differentiate between the dominated and non-dominated solutions, speed in dominance filtering directly impacts the computational performance of the PSEA. The implementation of a fast dominance filtering relies on intelligently decomposing the set of solutions to be partitioned, and working systematically on smaller chunks of the set of solutions. Such a problem decomposition results in a significantly reduced problem complexity, since the cardinality (m) for these sets is typically much smaller than the cardinality of the entire solution set ($m << M$).

The operation of the fast dominance filter (FDF) is illustrated in Figure 3. The initial data set is depicted in Fig. 3a. The problem is first converted to a maximization problem by multiplying the value of any minimization dimension by –1. In the second step, each dimension is considered in turn, the remaining dimensions are split into a number of bins (increasing with each dimension to some maximum), and the point with the maximum value of the dimension under consideration within each bin is found (Figure 3b, 3c). This set of points is compared to the remaining points and all dominated points are removed from further consideration. Note that, as the algorithm progresses, fewer and fewer points are compared, improving the computational performance. In the third step (Fig. 3d), each point is considered in turn, and points dominated by the point under consideration are removed. The resulting set of points is non-dominated. The increase in speed of the FDF varies of course based on the coarseness of the binning method and the distribution of the raw data. If the number of points in each bin is consistently smaller than M, very good speedups are possible in the dominance filtering. The maximum speedups therefore arise when the raw data are evenly distributed in the objectives space.

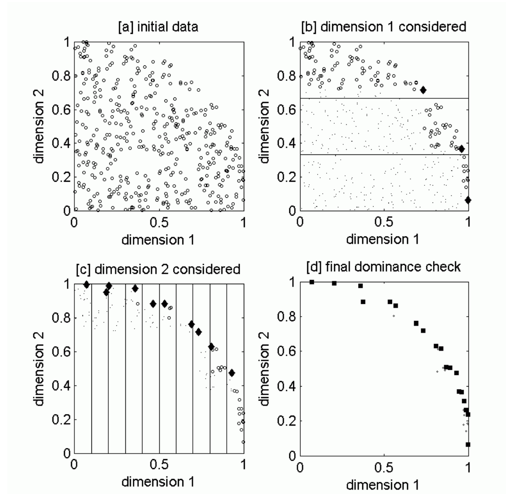

Figure 3. Fast Dominance Filtering. Lines delineate bins, filled diamonds are maximum values within that bin along the considered dimension; open circles are non-dominated points (at each step); dots are dominated points.

4) Target Objective Genetic Algorithm (TOGA)

The search for optimal portfolios is augmented with a Target Objective Genetic Algorithm (TOGA) [20], a non-Pareto, non-aggregating function approach to multi-objective optimization that borrows concepts from goal programming [21] and Schaffer's Vector Evaluated Genetic Algorithm [6]. Unlike the PSEA, which is driven by the concept of dominance, TOGA tries to find solutions that are as close as possible to a pre-defined target for one or more criterion.

Given k objectives, TOGA requires the user to provide a set of c target vectors, $\overline{T}^c$, for k-1 objectives, where c is the number of optimal points the user wishes to find in one run of the algorithm. In the absence of domain knowledge to the contrary, taking all combinations of points of interest for each objective (considered separately) is the recommended method for developing $\overline{T}^c$. In addition, k-1 scaling factors, w, are required (which can typically be determined empirically). During each generation, the objective values for each chromosome in the population are evaluated once, then the fitness "from the perspective" of each target vector is calculated based on the objective values using (for a maximization problem):

$$F^c = f_1(\bar{x}) - w_1(f_2(\bar{x}) - T_1^c)^2 - w_2(f_3(\bar{x}) - T_2^c)^2 - \ldots - w_{(k-1)}(f_k(\bar{x}) - T_{(k-1)}^c)^2 \qquad (5)$$

where:

F^c is the fitness from the perspective of target vector c

$f_k(\bar{x})$ is the objective value of objective k

$w_{(k-1)}$ is a scaling factor for objective k

$T_{(k-1)}^c$ is the target value for objective k in target combination c

Given F^c, selection (with replacement) of a small, even number (e.g., 4, 6, or 8) of chromosomes is performed for each of c target combinations using a rank based procedure, with the probability of selecting the i^{th} chromosome (when the chromosomes are sorted according to fitness), P_i according to:

$$P_i = \frac{q(1-q)^{r-1}}{1-(1-q)^n} \tag{6}$$

where:

q is the probability of selecting the fittest individual

r is the rank of chromosome (1 is best)

n is the population size

Crossover is performed separately on each of the c subpopulations formed. It is recommended that q be kept quite low (much less than 0.1), to promote exchange of genetic material from different locations on the optimal front (i.e., so that universally good building blocks developed in one region can spread rapidly to other regions). However, because of the low likelihood of preserving the best chromosome from each target combination, elitism is a critical feature of TOGA. During each generation the chromosome with the maximum F^c for each of the c combinations is passed on, unmodified, to the next generation. The method of mutation is not specified for TOGA (mutation should occur, but the particulars of implementation are not dictated by the approach). The selection and elitism strategies are the key elements in TOGA. By selecting from the perspective of each $\overline{T}^c$, subpopulations specializing in performance at that particular point on the optimal front are developed. However, diversity is maintained by having multiple $\overline{T}^c$, and by elitism. Moreover, by sharing good genetic building blocks developed on different regions of the optimal front, substantial efficiency is gained when compared to individual goal programming runs, which have to start from scratch.

TOGA has several advantages as an optimization method. First, the mechanics of the TOGA process do not require much computing power, particularly when compared to Pareto based approaches. TOGA takes advantage of *a priori* knowledge of the objective space to efficiently evolve solutions. Knowing where to search, and driving the population toward these points can effectively maximize efficiency at these points of interest. The use of disparate target points for subpopulation formation insures diversity in the population overall, allows TOGA to generate multiple optimal points during each run, and allows these points to be generated on either a concave or convex Pareto front. Moreover, because TOGA shares good building blocks developed during a run throughout a population, substantial computing time is saved when compared to individual runs of an aggregating GA approach.

However, TOGA also has a several limitations. It is most efficient when the researcher has some domain knowledge and is able to select good combinations of objectives, and the resulting solutions (as with goal programming) may not be Pareto-optimal, depending on the target points chosen, but are optimal solutions at that particular combination of objectives. Another limitation is that some experimentation is typically required to determine good scaling factors for the objectives.

5) *Summarizing the Search.*

Our search for the efficient frontier consists in initializing the population with the RLP, generating an interim Pareto front with PSEA and TOGA, completing gaps in the Pareto front with TOGA, and storing the result in a repository. After many runs, we filter the repository with an efficient dominance filter and generate the first efficient frontier, as shown in Fig. 1. This richly sampled Pareto front is analyzed for possible gaps, and augmented with a last run of TOGA, leading to the generation of the second efficient frontier (second box in Fig. 1). Each point in this front represents a non-dominated solution, i.e. a viable portfolio. At this point we need to incorporate the decision maker's ***preferences*** in the return-risk tradeoff (third box in Fig. 1). Our goal is to reduce the large number of viable solutions into a much smaller subset that could then be further analyzed for a final portfolio selection.

C. *Portfolio Selection: Exercising Progressive Preferences*

1) *Process Description*

In our example, we used a portfolio consisting of 1500+ decision components, each of which represents a percentage of the capital to be invested in a specific financial instrument. Our goal was to maximize Book Yield (*B-yield*), while minimizing *Variance* (*Risk1*) and *Simplified Value at Risk* (*Risk2*). Using the search approach described in III.B.5, we obtained a front with 1,237 points. To better exercise our preferences, we augmented this space with three additional metrics: *Market Yield* (*M-yield), Dollar-duration Weighted Market Yield* (*DWM-yield),* and *transaction cost (Delta*). For each of the 1,237 points, these three metrics are computed. Now, we needed to down-select the non-dominated points based on iterative constraints and tradeoffs performed in this augmented space.

In order to provide a progressive preference, we needed to understand and accept the effects of the tradeoffs already executed. To support this interaction we used a graphical tool - licensed from Aetion Technologies LLC - which shows 2-D projections of the multi-dimensional Pareto Front. In this case, there are *C(6,2)*=15 possible 2-D projections of the Pareto front. However, not all of them will be used, since some of these projections represent higher priorities in the decision-making process and will have a stronger effect in the down selection process.

2) *Intersection of Constraints*

The first step consists in finding the "sweet spot" in which all desirable constraints are satisfied. For example, we can establish the *minimal amount of return* (*B-yield*) and the *maximum amounts of risk* (*Risk1 and Risk2*) that the decision maker is willing to accept. We can also draw similar hyper-planes along the other three dimensions. Since the intersection of constraints is associative, the result is independent of the order of execution.

Fig. 4 illustrates four 2-D projections of the Pareto front: (*B-yield, Risk1*), (*B-yield, Risk2*), (*Risk2, Risk1*), and (*B-yield, DWM-Yield*). The first two projections show tradeoffs between return and risk (using different metrics), while the third and fourth show relationship between risk metrics and between return metrics. The figure shows the result of applying these constraints, where we are left with 479 points (from the original 1,237).

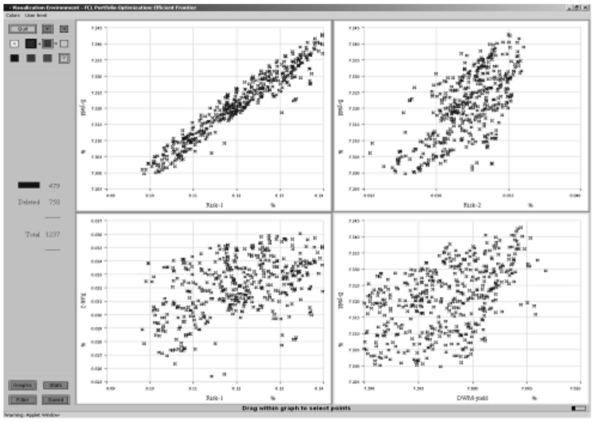

Fig 4. Four 2-D projections of the Pareto Front containing 479 points. The 2-D projections are: (*B-yield, Risk1*), (*B-yield, Risk2*), (*Risk2, Risk1*), and (*B-yield, DWM-Yield*).

3) Application of Ordinal Preferences

At this point, all remaining aggregations will reflect non-associative tradeoffs, and therefore the order of their application will determine the final results. Rather than establishing a *cardinal* preference order, which is usually difficult to elicit, we propose the use of an *ordinal* order, which is more natural and intuitive for the decision maker. Therefore, our preferences are expressed by the order in which we visit and execute *limited, local tradeoffs* in each of the available 2-D projections of the Pareto front.

In our example, the first 2-D projection visited is the (*B-yield, Risk1*). We use a *local dominance filter* (which could be extended by some tolerance parameter) [22] to reduce the 479 points down to 55 points, each of which is non-dominated in this projection. We noted that the *order* in which these local tradeoffs are performed is *relevant* to the result. By starting in the (*B-yield, Risk1*) projection we are stating that, after satisfying all independent constraints, *we want the best performance of the portfolio in this subspace* (trying to maximize Book Yield and minimize Variability).

After imposing this ordinal preference, we zoom into the resulting 55 points and we perform the *next most important tradeoff*, generating a Pareto Front in the (*B-yield, Risk2*) region. This process yields 10 points, as illustrated in Fig. 5. At his point we could zoom into this new subset of 10 points and use another metrics, e.g., transaction cost (*Delta)*, to select the lowest cost portfolio (shown in red) in Fig. 6.

We used this example to illustrate the complex interactions between the decision maker and the MCDM system. In many real-world applications, it is not practical or possible to elicit the DM's preference all at once – for an a priori or a posteriori aggregation. Rather, the DM needs to understand the available space of options, the shape of the Pareto front, and the costs/benefits of the available tradeoffs. The use of progressive preference elicitation provides a natural selection mechanism to identify a small number of good solutions.

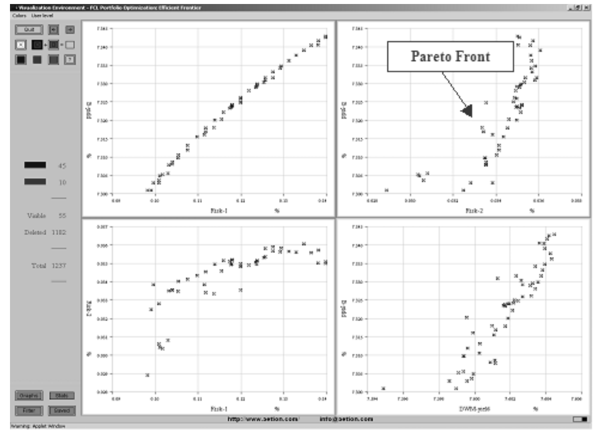

Fig 5. Four 2-D projections of the Pareto Front containing 55 points: 45 dominated and 10 non-dominated. The 2-D projections are: (*B-yield, Risk1*), (*B-yield, Risk2*), (*Risk2, Risk1*), and (*B-yield, DWM-Yield*).

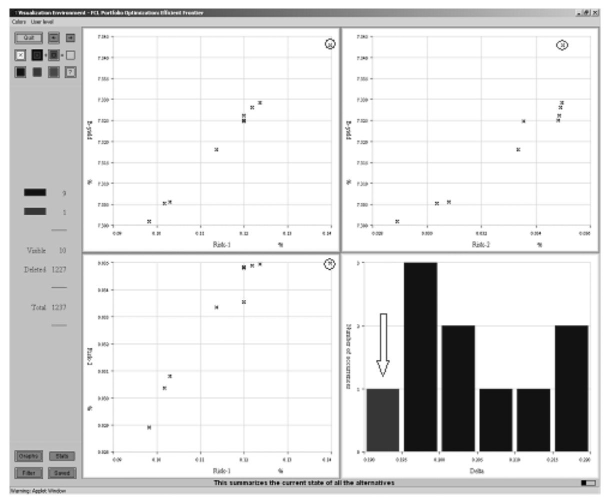

Fig. 6. Four 2-D projections of the Pareto Front containing 10 points. The 2-D projections are: (*B-yield, Risk1*), (*B-yield, Risk2*), (*Risk2, Risk1*), and (*Count, Delta*).

IV. Conclusions

We have described the problem of optimal allocation of available financial resources to a diversified portfolio of long and short-term financial assets in accordance with risk, liability, and regulatory constraints. This problem involves over fifteen hundred financial assets, and investment decisions of several billion dollars. The return and risk measures are complex linear or nonlinear functions of a variety of market and asset factors.

Given the explicit need for customization and hybridization in methods for portfolio optimization, it is not expected that existing multi-objective optimization algorithms could be applied without extensive modifications. Specifically, the requirement to optimize while satisfying a large number of linear constraints excludes the ready application of prior evolutionary multi-objective optimization approaches. This aspect was a principal motivation to develop the novel hybrid techniques described in this paper.

We have developed and tested a Pareto Sorting Evolutionary Algorithm (PSEA) that is able to robustly identify the Pareto front of optimal portfolios defined over a space of returns and risks. Our algorithm uses a secondary storage and maintains the diversity of the population by using a convex crossover operator, incorporating new random solutions in each generation of the search; and using a non-crowding filter. Given the reliance of the PSEA on the continuous identification of non-dominated points, we developed a Fast Dominance filter to implement this function very efficiently.

The key challenge in solving the portfolio optimization problem was presented by the large number of linear allocation constraints. The feasible space defined by these constraints is a high dimensional real-valued space (1500+ dimensions), and is a highly compact convex polytope, making for an enormously challenging constraint satisfaction problem. We leveraged our knowledge on the geometrical nature of the feasible space by designing a Randomized Linear Programming algorithm that robustly samples the boundary vertices of the convex feasible space. These extremity samples are seeded in the initial population of the PSEA and are exclusively used by the evolutionary multi-objective algorithm to generate interior points (via interpolative convex crossover) that are always geometrically feasible.

We further enhanced the quality of the Pareto Front by using a Target Objectives Genetic Algorithm (TOGA), a non-Pareto non-aggregating function approach to multi-objective optimization. Unlike the PSEA, which was driven by the concept of dominance, the TOGA finds solutions that are *as close as possible* to a pre-defined target for one or more criterion. We used this to fill potential "gaps" in the Pareto front.

Finally, we incorporate the decision-maker's *preferences* in the return-risk tradeoff to perform our selection. The goal was to reduce thousands of non-dominated solutions into a much smaller subset (of ~10 points), which could be further analyzed for a final portfolio selection. After obtaining a 3-D Pareto front, we augmented this space with three additional metrics, to reflect additional constraints for use in the tradeoff process. This augmented 6-D space was used for the down-selection problem. To incorporate progressive ordinal preferences, we used a graphical tool to visualize 2-D projections of the Pareto front. After applying a set of constraints to further refine the best region, we used an *ordinal* preference, defined by the order in which we visited and executed *limited, local tradeoffs* in each of the available 2-D projections of the Pareto front. In this approach the decision-maker can understand the available space of options and the costs/benefits of the available tradeoffs. The use of progressive preference elicitation provides a natural mechanism to identify a small number of the good solutions. However, in problems with a large number of objectives, a more formal preference elicitation method [23] may be applied in conjunction with our graphical methods.

Due to space restrictions, and in interest of presenting the entire portfolio design architecture, we have not included extensive experimental results in this paper. A forthcoming publication will cover in detail aspects of convergence quality, diversity quality etc. of the portfolio optimization methods presented in this paper.

References

[1] H. Markowitz, "Portfolio Selection," *J. of Finance*, 7, pp. 77—91,1952.

[2] K. C. Chalermkraivuth, S. Bollapragada, M. C. Clark, J. Deaton, L. Kiaer, J. P. Murdzek, W. Neeves, B. J. Scholz, and D. Toledano, "GE Asset Management, Genworth Financial, and GE Insurance Implement a Novel Sequential Linear Programming Algorithm for Portfolio Optimization," *INFORMS Interfaces*, (to appear in September 2005).

[3] V. Tarascio, *Pareto's Methodological Approach to Economics*, University of North Carolina Press, Chapel Hill, VA, 1968.

[4] K. M. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic, Boston, MA, 1998.

[5] R. V. Tappeta, J. E. Renaud, and J. F. Rodriguez, "An Interactive Multiobjective Optimization Design Strategy for Decision Based Multidisciplinary Design," *Engineering Optimization*, 34(5), pp. 523—544, 2002.

[6] J. D. Schaffer, *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, PhD thesis, Vanderbilt Univ., Nashville, TN, 1984.

[7] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, "Evolutionary Algorithm MOP Approaches," *Evolutionary Algorithms for Solving Multi-Objective Problems*, pp. 59-99. Kluwer Acad., 2002.

[8] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparitive Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, 3(4), pp. 257—271, 1999.

[9] S. Arnone, A. Loraschi, and A. Tettamanzi, "A Genetic Approach to Portfolio Selection," *Neural Network World, Int. Jour. Neural and Mass-Parallel Computing and Inform. Syst*. 3(6), pp. 597—604, 1993.

[10] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, "Heuristics for Cardinality Constrained Portfolio Optimization," *Computers and Operations Research*, 27(13), pp. 1271—1302, 2000.

[11] M. Ehrgott, K. Klamroth, and C. Schwehm, "An MCDM Approach to Portfolio Optimization," *European Journal of Operational Research*, 155, pp. 752—770, 2004.

[12] F. Streichert, H. Ulmer, and A. Zell, "Evaluating a Hybrid Encoding and Three Crossover Operators on the Constrained Portfolio Selection Problem," In *Proc. IEEE International Conference on Evolutionary Computation*, pp. 932—939, 2004.

[13] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto Ant Colony Optimization: A Metaheurstic Approach to Multiobjective Portfolio Selection", *Annals of Operations Research*, 131(1-4), pp. 79—99, 2004.

[14] F. Schlottmann and D. Seese, "A Hybrid Heuristic Approach to Discrete Multi-objective Optimization of Credit Portfolios", *Computational Statistics and Data Analysis*, 47(2), pp. 373—399, 2004.

[15] G. Fong and O. A. Vasicek, "A Multidimensional Framework for Risk Analysis," *Financial Analysts Journal*, July/August 1997.

[16] J. C. Hull, *Options, Futures & Other Derivatives*, Prentice Hall, 2000.

[17] K. C. Chalermkraivuth, 2004. Analytical Approaches for Multifactor Risk Measurements, GE Tech. Inform. 2004GRC184, Niskayuna, NY.

[18] J. Horn. " Multicriterion Decision Making", *Handbook of Evolutionary Computation*, (T. Back, D. B. Fogel, and Z. Michalewicz, eds.), pages F1.9:1-F1.9:15, IOP Publ. Ltd and Oxford University Press, 1997.

[19] J. Kubalik and J. Lazansky, (1999). "Genetic Algorithms and their Tuning", In Dubois Daniel M., editor(s), *Computing Anticipatory Systems*, pp. 217-229, Liege, Belgium, American Institute Physics, 1999

[20] N. Eklund. *Multiobjective Visible Spectrum Optimization: A Genetic Algorithm Approach*. Ph.D. Thesis, Rensselaer (RPI), Troy, NY, 2002.

[21] K. Deb. "Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms", Proc. 1999 Congress on Evolutionary Computation, pp. 77-84 Washington, DC, 1999.

[22] N. S. Iyer, "A family of dominance rules for multiattribute decision making under uncertainty," *IEEE Trans. Systems, Man, and Cybernetics, Part A*, vol. 33, pp. 441-450, 2003.

[23] D. Cvetkovic and I. C. Parmee, "Preferences and their Application in Evolutionary Multiobjective Optimisation", *IEEE Transactions on Evolutionary Computation*, 6(1), pp. 42—57, 2002.