Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)

A Review of Two Industrial Deployments of Multi-criteria Decision-making Systems at General Electric

Raj Subbu, Piero Bonissone,

Srinivas Bollapragada, Kete Chalermkraivuth, Neil Eklund,

Naresh Iyer, Rasik Shah, Feng Xue, and Weizhong Yan

General Electric Global Research, One Research Circle, Niskayuna, NY 12309, USA

***Abstract*—Two industrial deployments of multi-criteria decision-making systems at General Electric are reviewed from the perspective of their multi-criteria decision-making component similarities and differences. The motivation is to present a framework for multi-criteria decision-making system development and deployment. The first deployment is a financial portfolio management system that integrates hybrid multi-objective optimization and interactive Pareto frontier decision-making techniques to optimally allocate financial assets while considering multiple measures of return and risk, and numerous regulatory constraints. The second deployment is a power plant management system that integrates predictive modeling based on neural networks, optimization based on multi-objective evolutionary algorithms, and automated decision-making based on Pareto frontier techniques. The integrated approach, embedded in a real-time plant optimization and control software environment dynamically optimizes emissions and efficiency while simultaneously meeting load demands and other operational constraints in a complex real-world power plant.**

***Index Terms* — Financial portfolio, hybrid optimization, industrial control, evolutionary algorithms, Pareto frontier, neural networks, multiple objectives, decision-making.**

I. Introduction

We review two industrial deployments of multi-criteria decision-making systems at General Electric from the perspective of their constituent components. The deployments highlighted represent extreme points of a framework for the structured development, deployment, and update of multi-criteria decision-making systems. The eight principal system components considered are:

- Runtime requirements
- Deployment architecture
- Response evaluation,
- Search method,
- Objectives and constraints complexity,
- Uncertainty management,
- Decision-making needs and method, and
- Update requirements for solution fidelity.

The first deployment [1] is a financial portfolio management system that integrates hybrid multi-objective optimization and interactive Pareto frontier decision-making techniques to optimally allocate financial assets while considering multiple measures of return and risk, and numerous regulatory constraints. The hybrid multi-objective optimization approach 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. The method combines 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-world portfolios with hundreds to thousands of assets, and is currently being used for investment decision-making in industry.

The second deployment [2] is a power plant management system that integrates predictive modeling based on neural networks, optimization based on multi-objective evolutionary algorithms, and automated decision-making based on Pareto frontier techniques. The predictive models are adaptive, and continually update themselves to reflect with high fidelity the gradually changing underlying system dynamics. The integrated approach, embedded in a real-time plant optimization and control software environment has been deployed to dynamically optimize emissions and efficiency while simultaneously meeting load demands and other operational constraints in a complex real-world power plant.

In each of these deployments, we leverage, represent, and incorporate domain knowledge for robust problem solving and decision-making (see [3] for a deeper exposition on this concept).

The rest of this paper is organized as follows: Section II presents background on topics relevant to this paper; Section III presents a comparative review of the multi-criteria decision-making components of these two deployed systems. We conclude in Section IV.

II. Background

In this section, we present background in multi-objective evolutionary algorithms, portfolio optimization, neural networks, and model-predictive optimization and control.

A. Multi-objective Optimization

Most real-world optimization problems have several, often conflicting objectives. Therefore, the optimum for a multi-objective problem is typically not a single solution—it is a set of solutions that trade-off between objectives. The Italian economist Vilfredo Pareto first generally formulated this concept in 1896 [4], 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*.

A review of mathematical programming-based optimization methods for multi-objective problems is presented in [5]. These techniques generally 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. 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]).

B. Multi-objective Portfolio Optimization

Modern Computational Finance has its historical roots in the pioneering portfolio theory of Markowitz [9]. 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. Varying the risk target and maximizing on the return measure obtains the well-known risk-return efficient frontier. 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. [10] 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.

C. Relevant Work in Evolutionary Portfolio Optimization

An early approach to evolutionary portfolio optimization was presented in [11]. Chang et al. [12] present a comparison of Tabu Search, Simulated Annealing, and Evolutionary Algorithms on the Markowitz mean-variance portfolio optimization problem. Ehrgott et al. [13] use neighborhood search, Tabu Search, Simulated Annealing, and a Genetic Algorithm for a portfolio optimization problem with objectives aggregated via user-specified utility functions. This, and the previous evolutionary portfolio optimization approaches have solved the inherently multi-objective 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. Streichert et al. [14] have proposed an evolutionary multi-objective 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 multi-objective optimization sense. Our problem formulation includes multiple linear and nonlinear measures of risk and return, and a large number of real-world linear allocation constraints including the basic knapsack additive requirement (i.e. sum of all fractional allocations must equal 1). This latter constraint characteristic is a key differentiator to earlier formulations such as in [13] wherein the knapsack additive requirement is the principal constraint with some additional constraints to bound allocation fractions. Recently, metaheuristic and hybrid evolutionary techniques have been applied to the portfolio optimization problem [15]-[16].

We have developed [1] a novel hybrid evolutionary multi-objective 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 have also developed 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. In contrast to prior approaches, importantly, our approach incorporates a systematic approach for portfolio decision-making, which has thus far been lacking in the literature.

D. Predictive Modeling

Predictive models are routinely used in a variety of business, industrial, and scientific applications. These models could be based on data-driven construction techniques, based on physics-based (e.g. lumped parameter models) construction techniques, or based on a combination of these techniques. Typically, data-driven modeling is applied for modeling complex systems for which the physics is not well understood, or for which a physics-based model is very difficult to develop and maintain at the accuracy levels required of the application. Physics-based models may as well have approximation issues due to simplification of the model structure and uncertainties involved in parameter estimations.

Neural network modeling is a well-known instance of data-driven predictive modeling, and utilizes computational structures/systems composed of many simple interconnected but parallel functional and gain elements [17]. Such structures are modeled on structures in biological nervous systems. Their computational paradigm is one of "parallel distributed processing," and such structures are capable of multivariable, nonlinear, non-parametric modeling. They are also universal approximators [18]. Such data-driven models are trainable using mathematically well-defined algorithms (e.g., learning algorithms). That is, such models may be developed by training them to accurately map system inputs onto system outputs based upon measured or existing process data. This training requires the presentation of a diverse set of several input-output data vector tuples to the training algorithm. The system's historical operational data is first subject to extensive cleansing and filtering to make it suitable for deriving input-output relationships. The trained models may then accurately represent the input-output behavior of the underlying processes. As a result, neural networks have been widely used for modeling, classification, and prediction across a wide spectrum of scientific and engineering applications. They also have a rich history of application in the modeling and control of dynamic systems [19].

E. Model-predictive Optimization and Control

Model-predictive control of industrial processes has a rich research and applications history (e.g. [20]-[24]), so the interested reader is referred to these articles for further study. In this approach, a forward-looking predictive system model is probed by an optimization algorithm to identify and deploy a control strategy or system inputs (setpoints) as a function of time and operational needs.

Prior work in the space of model-predictive control techniques that leverage neural networks for modeling and single-objective evolutionary algorithm methods for optimization that probe these models to identify an optimal input vector, appear in (e.g. [25]-[29]). When multiple objectives need to be considered, these approaches apply a linear or nonlinear aggregation function over the objectives.

We now review prior work in the space of model-predictive techniques that leverage neural networks for modeling and multi-objective evolutionary algorithms for optimization. This concept has been applied to the control of robot arms [30], to robot path planning [31], and to a simplified model of a single-input single-output nonlinear system [32]. Compared to these prior approaches, ours is a development and deployment of these concepts for a complex real-world application such as an industrial process or a power plant.

In our approach [2], predictive models are interfaced with an optimizer once it is determined that they are capable of faithfully predicting various system outputs, given a set of inputs. This determination may be accomplished by comparing predicted versus actual values during a validation process performed on the models. Various methods of optimization may be interfaced, e.g., evolutionary algorithms (EAs), which are optimization techniques that simulate natural evolutionary processes, or gradient-descent optimization techniques. The predictive models coupled with an optimizer may be used for realizing a process controller (e.g., by applying the optimizer to manipulate process inputs in a manner that is known to result in desired model and process outputs). For example, for a power plant, the inputs are the various controllable and ambient variables, and the outputs may include emissions characteristics such as NOx and CO, efficiency characteristics such as Heat Rate (inversely related to efficiency), and operational characteristics such as Load.

III. Multi-criteria Decision Making Framework Components Review

In this section, we review the two multi-criteria decision-making deployments at General Electric from the perspective of their constituent components. The deployments represent extreme points of a developing framework for multi-criteria decision-making systems. Such a framework allows us to decompose and reason about the core components necessary to develop and deploy multi-criteria decision-making systems. First, we present high-level remarks on system characteristics, followed by a more extensive discussion. Table 1 presents a comparative summary of the principal characteristics of these two deployed systems.

Table 1: Summary of multi-criteria decision-making component characteristics of the two deployed systems.

	Portfolio management system	**Power plant management system**
Performance requirements	*Off-board (batch)*	*On-board (real-time)*
Deployment architecture	*Centralized*	*Distributed*
Response evaluation	*Mathematical* description of responses	*Data-driven* computation of responses
Search method	*Hybrid* multi-objective optimization	*Evolutionary* multi-objective optimization
Objectives and constraints complexity	*Multiple* objectives, *multiple* constraints	*Two* objectives, *two* constraints
Uncertainty management	Uncertainty measures *implicitly* captured in objectives	*Explicit* externally driven uncertainty management
Decision-making needs and method	*Interactive* graphical with human in loop	*Automated* decision-making via constraints and weights
Update requirements for solution fidelity	*Implicit* via update of problem descriptors in database	*Explicit* via periodic retraining of data-driven models

A. Portfolio Management System

1) Runtime Requirements

Figure 1 shows the block diagram of the portfolio management system. The principal modules in the system address identification of the search space boundaries using linear programming, optimization using multi-objective evolutionary algorithms, strategic densification of the Pareto frontier, and finally interactive portfolio selection. The system is deployed as an *off-board* application in a front-end web-based interactive environment, and is geared for use by portfolio managers.

2) Deployment Architecture

This application uses a *centralized* architecture. A back-end database maintains information on available assets, portfolio characteristics, and a variety of asset performance indices. The database also stores the results of optimization and decision-making runs. Our portfolio management approach is currently in use at General Electric Asset Management, General Electric Insurance, and Genworth Financial.

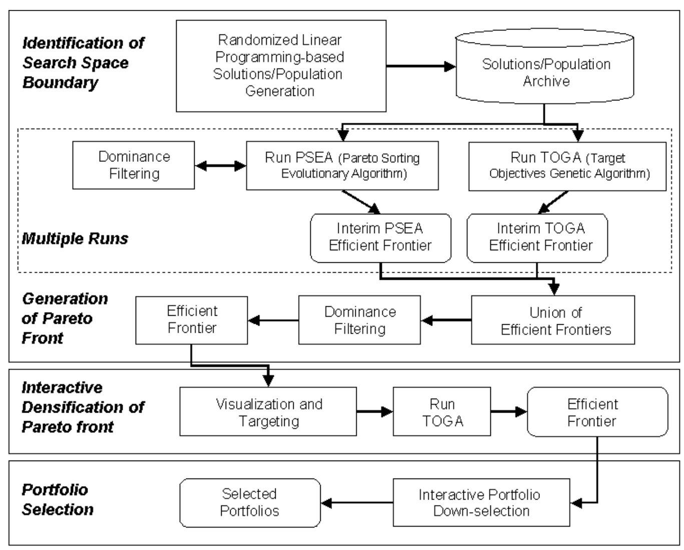

Figure 1: Block diagram of architecture.

3) Response Evaluation

In asset-liability management (ALM) applications, surplus variance is used as a measure of risk. We compute portfolio variance using an analytical method based on a multifactor risk framework [33]-[34]. 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. 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 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 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 [35]. These metrics define the 3-D optimization space. Now, we 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 [34]. 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.

4) Search Method

The dashed block in Figure 1 shows the search components of our approach. We 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 further enhance 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 is 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. Details on the PSEA and TOGA methods may be found in [1]. We used TOGA to fill potential "gaps" in the Pareto front. We initialize 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. 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. 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.

5) Objectives and Constraints Complexity

While the portfolio optimization problem has multiple linear and nonlinear measures of return and risk, the key challenge in solving the portfolio optimization problem is 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.

6) Uncertainty Management

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 via the capture of the inherent uncertainty in the asset allocation decisions, and are used to quantify various aspects of portfolio uncertainty. Therefore, uncertainty management is implicitly achieved via optimization over risk measures.

7) Decision Making

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. Since these objectives cannot be satisfied simultaneously, we need to accept tradeoffs. We incorporate the decision-maker's preferences in the return-risk tradeoff to perform our selection—the goal being the reduction of 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 augment 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 as shown in Figure 2 to visualize 2-D projections of the Pareto front—in this case the projections being (*Book-yield, Risk1*), (*Book-yield, Risk2*), (*Risk2, Risk1*), and (*Book-yield, Duration-weighted-market-Yield*). 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.

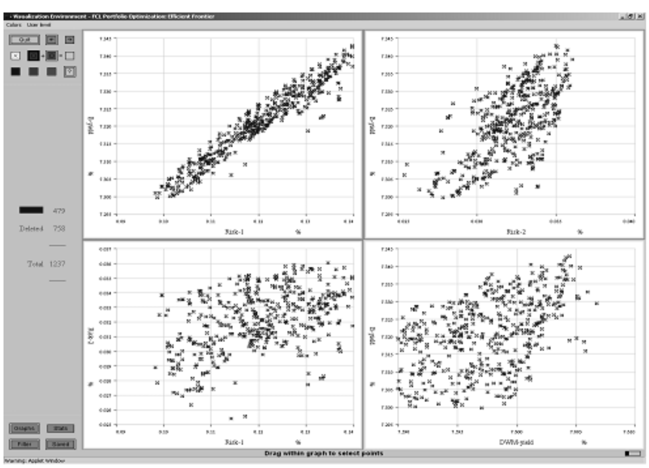

Figure 2: Four 2-D projections of the Pareto Front containing 479 interactively down-selected points.

The use of progressive preference elicitation provides a natural mechanism to identify a small number of the good solutions. Essentially, the interactive decision-making method allows the decision-maker to specify preferences for regions within portfolio performance space projections to select portfolios for further iterative down-selection. Pareto dominance in each performance space 2-D projection may further be utilized in this filtering. Selection preferences are interactively specified over the augmented 6-D performance space, rather than over the optimized 3-D Pareto front. In problems with a large number of objectives, a more formal preference elicitation method [36] may be applied in conjunction with our graphical methods.

8) Update Requirements

The portfolio management system has little or no internal update requirements for continued high-fidelity performance. Only inputs corresponding to defining and creating a portfolio optimization scenario are subject to change and update. Once a problem is setup, there is no requirement to frequently maintain or update the response computations, as they are all defined mathematically, and do not degrade as a function of time.

B. Power Plant Management System

1) Runtime Requirements

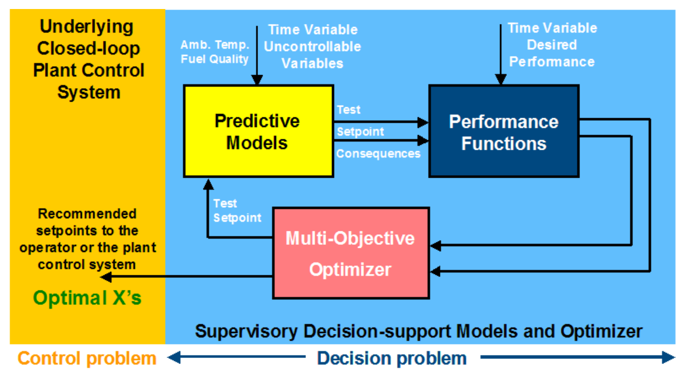

Figure 3: Architecture of model-predictive multi-objective optimization.

Figure 3 shows a high-level architecture of the power plant management system. Nonlinear neural-network models are used to represent mappings between the inputs space of control variables and time variable ambient uncontrollable variables, and the various outputs (objectives and constraints) of interest. First-principles-based methods and domain-knowledge are used to identify the relevant model inputs. The evolutionary multi-objective optimizer generates test inputs/setpoints and receives as feedback the corresponding output performance metrics after transformation by suitable objective (performance) functions. The multi-objective optimizer uses this feedback to generate and identify the Pareto-optimal set of input-output vector tuples that satisfy operational constraints. A decision function is superimposed on this Pareto-optimal set of input-output vector tuples to identify a deployable input-output vector, which is then dispatched to the underlying plant control system, or recommended to the operator for execution.

A Pareto-optimal front that jointly minimizes NOx and Heat Rate (inversely related to efficiency) for a 400MW target load demand in a 400MW power plant is shown in Figure 4. In this figure, the circles show the range of historical operating points from a NOx—Heat Rate perspective. The stars and inter-connecting line show the optimized Pareto frontier in the NOx—Heat Rate space. Each point not on this frontier is a sub-optimal operating point—the goal being the operation of the plant or process at a Pareto optimal point at all times. Moving the system operation from the interior of the decision space to the Pareto frontier results in a large operational savings opportunity.

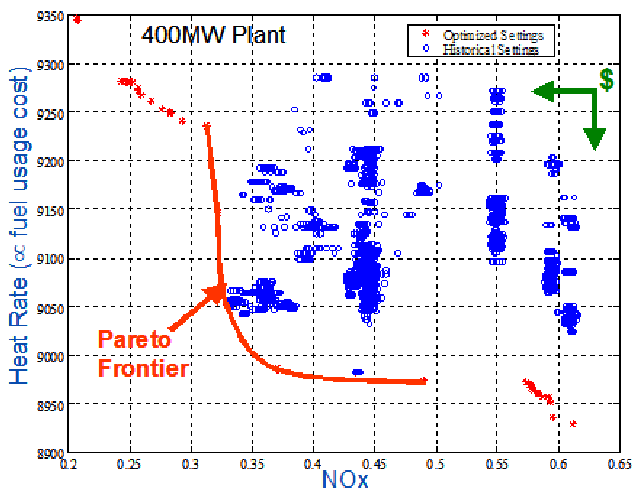

Figure 4: Pareto tradeoff between NOx emissions and Heat Rate (inversely related to efficiency) for a 400MW power plant. The knee of the frontier is shown connected based on the identified Pareto optimized setpoints.

Our integrated approach to plant management is embedded in a *real-time* plant optimization and control software environment. This has been deployed as an *on-board* application to dynamically optimize emissions and efficiency while simultaneously meeting load demands and other operational constraints in a complex real-world power plant. The plant management system front-end interfaces with the plant control system at the back-end.

2) Deployment Architecture

Multiple versions of this system could be developed and deployed for different plants, using a *distributed*, loosely connected architecture. The historical data of each plant would be used to train and customize each system. The global performance of all systems could be used to fine-tune the tradeoffs between efficiency and emissions.

3) Response Evaluation

Nonlinear neural-network models are used to represent mappings between the inputs space of control variables and time variable ambient uncontrollable variables, and the various outputs (objectives and constraints) of interest. For example, for a power plant, the inputs are the various controllable and ambient variables, and the outputs may include emissions characteristics such as NOx and CO, efficiency characteristics such as Heat Rate, and operational characteristics such as Load. First-principles-based methods and domain-knowledge are used to identify the relevant model inputs. This is an iterative process that requires a combination of domain knowledge and information analysis techniques to select the minimal set of inputs that can be used to suitably model each output of interest. Each of the models must meet a domain specified threshold in predictive performance. Typically, each such model must be capable of a highly robust forward prediction and a very low misprediction rate (about 0.1% to 3%). Validation of the prediction performance of the models may include metrics such as the Mean Absolute Error (MAE), which is the average of the absolute prediction errors, and serves as a measure similar to the standard deviation of prediction errors:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \qquad (5)$$

where y_i is an actual i^{th} sample value, and $\hat{y}_i$ is the predicted value of the i^{th} sample of the n samples tested. The MAE is a reliable bound on the most likely range for mispredictions; a model with a low MAE is therefore desirable.

4) Search Method

A multi-objective evolutionary optimizer is employed to identify the Pareto frontier among the objectives of interest. However, for any of the inputs generated by the optimizer, certain constraints may be violated by those inputs settings if they were to be deployed. If any constraint is violated, we apply penalty function-based adjustments to the objectives over which the optimization is performed, so that those input vectors that lead to constraint violation will be automatically discarded as part of the evolutionary search. As an example, if Load is a constraint, and NOx, and Heat Rate are objectives to be optimized, then if a certain input vector exceeds or falls below the desired Load by a certain amount the NOx and Heat Rate values associated with that solution are respectively adjusted by penalty function factors.

5) Objectives and Constraints Complexity

The power plant management system is based on the use of two objectives (NOx, Heat Rate) and two constraints (Load, CO) that require predictive models. There is a set of global range constraints applied to the inputs space, and those correspond to the physical limits of the input variables.

6) Uncertainty Management

In order to do model based optimization, it is necessary to have accurate and reliable models over the operational range of interest. For physics-based models, it is possible in principle to be accurate for any combination of parameters. However, empirical data-driven models are only known to be accurate in the neighborhood of the training data. While empirical models may by chance be accurate in regions of parameter space where there is no data, generally they are inaccurate in those regions. Either way, there is no way to know which is the case - we can have no confidence of any empirical model far from region of data it was built on. This is because nonlinear data-driven prediction models are generally very good at interpolation, i.e. they are very good predictors in inputs spaces where sufficient training data surrounds that space. Such predictive models can also tolerate moderate extrapolation, i.e. they can stretch to make predictions in inputs spaces that are proximal to regions of training data. However, they are poor in extrapolative prediction in inputs spaces that are far from where training data exists.

Given the above problem, a first step in applying the optimizer to the models is specification of the search space constraints (ranges for each of the input variables). The search constraints are typically computed based on the respective input variable ranges from the historical data, and specify the exploration ranges for the optimizer. However, implementing a search simply based on such inputs-space exploration ranges will almost certainly not ensure high confidence in the predictions. Thus, when optimizing a complex system with highly constrained controls based on an empirical model, it is necessary to arrive at solutions that are near existing data. One way to solve this problem is to obtain data that covers the entire operational space of interest. However, in most practical situations this is an infeasible goal as the dimensionality of the inputs space for the modeling is typically large, and this precludes the generation and gathering of such well-distributed data. Often times, even the cost of executing a set of select experiments that probe the extremes of the feasible inputs-space is exorbitant.

A solution to this practical problem is to restrict the search to areas close to the regions of the available historical data. Over time, deploying new setpoints that are close to the available historical data will push the envelope of the historical data, and lead to enhanced model-based prediction capabilities. An efficient method to enable such a restricted search is to scan the historical data for operating points that were deployed when the ambient conditions were "close" to the current operating conditions. For instance, if the current load demand is 350MW and the ambient temperature is 70^{o} F, then it would be appropriate to scan the historical database for setpoints that were deployed when operating conditions were close (within specifiable bounds of the current load demand and current ambient temperature), and use these setpoints as seed points to initiate a restricted search.

Additionally, model performance could be significantly improved through the use of a committee of models and intelligent fusion of their predictions. Fusing the outputs from an ensemble of models in an effective way can often boost overall model accuracy. This concept is further developed in [37] wherein we present a novel method called locally weighted fusion, which aggregates the results of multiple predictive models based on local accuracy measures of these models in the neighborhood of the probe point for which we want to make a prediction. This fusion method may be applied to develop highly accurate predictive models. The locally weighted fusion method boosts the predictive performance by 20~40% over the baseline single model approach for the various prediction targets. Relative to this approach, fusion strategies which apply averaging or globally weighting only produce a 2~6% performance boost over the baseline.

The use of approximate or meta models for fitness evaluation in evolutionary optimization (of computationally expensive problems) is increasing in importance. A survey of this topic has been published recently [38].

7) Decision Making

The multi-objective optimizer in conjunction with the predictive models and the decision function solve a decision problem as a function of time. Control of the transition of the plant or process state to achieve the recommendation is delegated to the underlying plant control system. In a supervisory mode of deployment, a recommendation is transmitted to an expert human operator who then programs the recommendations in the plant control system, while in an automated mode of deployment, the recommendations are directly transmitted to the plant control system. Such use cases necessitate the use of automated down-selection to a solution from the Pareto frontier, for execution. This down- selection is part of the multi-objective decision-making step.

The Pareto frontier in NOx—Heat Rate space identified from the multi-objective search is clipped by the systematic application of profit-based and operational-need constraints for each of NOx and Heat Rate. Next, a solution from this reduced frontier that is closest in inputs space to the current plant state is selected and transmitted to the plant control system. Such an approach minimizes the state deviations while achieving Pareto-optimal operation.

Figure 5 shows the performance gains that may be achieved in NOx emissions using this decision-making approach. When a decision-making function is used which simultaneously considers a tradeoff Pareto point at each instant, roughly 18% improvement in NOx emissions may be achieved (upper figure half). However, if the optimization favors a NOx minimization that satisfies a given Heat Rate constraint, more significant NOx emissions improvement is possible (lower figure half). Similarly, 1-2% improvements in Heat Rate are possible. In a typical power plant setting, such savings in NOx and Heat Rate are very significant and could lead to operational savings of hundreds of thousands to millions of dollars per year.

The decision-making approach further highlights the inherent flexibility of Pareto frontier techniques whereby the entire efficient set of solutions is first identified without regard to situation specific down-selection, and later a flexible decision function is superimposed to identify a deployable input set (or setpoint).

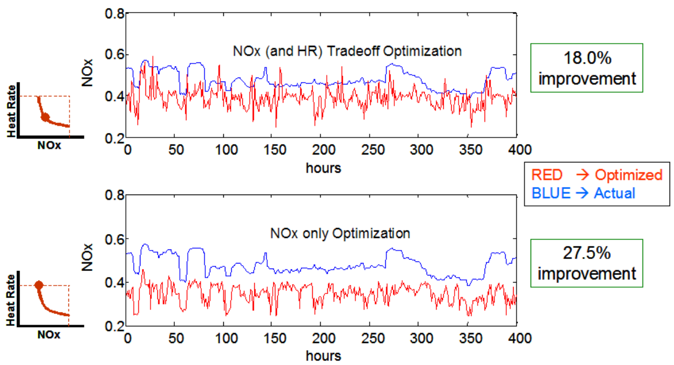

Figure 5: Pareto frontier tradeoff optimization of NOx and Heat Rate and NOx only optimization.

8) Update Requirements

In real-world applications such as the power plant management system, before we can use a model in a production environment we must address the model's entire life cycle, from its design and implementation, to its validation, tuning, production testing, use, monitoring and maintenance. By maintenance we mean all the steps required to keep the model vital (e.g., non-obsolete) and able to adapt to changes. Two reasons justify the focus on maintenance. Over the life cycle of the model, maintenance costs are the most expensive for software. Secondly, when dealing with mission-critical software we need to guarantee continuous operations or at least fast recovery from system failures or model obsolescence to avoid lost revenues and other business costs [39]. Models derived from combination of first principles, domain knowledge and field data need to be constantly monitored and updated to take into account changes that might occur in standard operating and maintenance procedures, in configuration upgrades, and in sensor failures and repairs. Furthermore, design assumptions regarding variability in process inputs and environmental variables must be monitored for significant changes. Due to this changing systems dynamics the underlying predictive models need to be monitored and continually updated via retraining. There will be situations in which the models should not be used, as they fall outside of the original design scope. There will also be situations in which the models should be retrained and retested before their respective new versions are re-deployed. In the first case, the monitoring system alerts the operator and terminates the use of the models for prediction and optimization. In the second case, the monitoring system alerts the operator indicating the detected changes occurring in the plant operation.

C. Degree of Difficulty Associated with the Multi-criteria Decision-making Components

In Table 2 below, we summarize the degree of difficulty associated with each of these system components corresponding to the two deployed systems. The degree of difficulty notionally represents both the computational and development complexities.

Table 2: Degree of difficulty of the multi-criteria decision-making components for the two deployed systems.

Degree of difficulty	**Portfolio Management System**	**Power Plant Management System**
Runtime requirements	Low-Medium	Medium-High
Deployment architecture	Low	Low-Medium
Response evaluation	Low	High
Search	High	Low
Objectives and constraints	High	Low
Uncertainty management	Low	High
Decision-making	High	Low
Update for solution fidelity	Low	Medium-High

IV. CONCLUSIONS

In the previous section, we described two applications with rather different requirements, which present interesting MCDM challenges. In the case of the Portfolio Management System, the key challenges were *searching* in a *high-dimensional space*, derived by very long chromosomes, and *performing decision in a high-dimensional performance space*, derived from multiple metrics used to evaluate risk and return. We addressed the first challenge by fusing multiple meta-heuristic search methods to generate a better-sampled Pareto surface. We solve the second challenge by using an interactive visualization system to enable the decision-maker to attach ordinal preferences to pairs of performance metrics. In the case of the Power Plant Management System the key issues were the *uncertainty embedded in the response evaluation,* and the *updating of the models* performing such evaluation. We addressed the first challenge by using multiple, diverse models and performing a customized fusion based on local weights. We solved the second challenge by using a dynamic learning method to update the neural networks that implemented such models.

There are many other combinations of MCDM requirements that pose equally interesting challenges, such as the autonomous updating of distributed models used for response evaluation; the concept of partial (or fuzzy) dominance of a solution point over another one, caused by the uncertainty or vagueness of their evaluations; the explicit uncertainty management/reduction of such evaluations, human factor issues that influence decision-making consistency and quality etc. Many of these challenges are further described in [40].

REFERENCES

[1] R. Subbu, P. Bonissone, N. Eklund, S. Bollapragada, and K. Chalermkraivuth, "Multiobjective Financial Portfolio Design: A Hybrid Evolutionary Approach," *Proceedings of the 2005 IEEE International Congress on Evolutionary Computation*, Sep. 2-5, 2005, Edinburgh, UK.

[2] R. Subbu, P. Bonissone, N. Eklund, W. Yan, N. Iyer, F. Xue, and R. Shah, "Management of Complex Dynamic Systems based on Model-predictive Multi-objective Optimization," *Proceedings of the 2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, July 12-14, 2006, La Coruña, Spain.

[3] P. Bonissone, R. Subbu, N. Eklund, and T. Kiehl, "Evolutionary algorithms + domain knowledge = Real-world evolutionary computation," *IEEE Transactions on Evolutionary Computation,* 10(3), 2006.

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

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

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

[10] 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*, September 2005.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] S. Haykin. *Neural Networks: A Comprehensive Foundation*. 2nd edition, Prentice Hall, 1998.

[18] G. Cybenko. "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals, and Systems*, 2:303-314, 1989.

[19] K. S. Narendra and K. Parthasarathy. "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Transactions on Neural Networks,* 1(1):4-27, 1990.

[20] D. W. Clarke. "Application of Generalized Predictive Control to Industrial Processes", *IEEE Control Systems Magazine,* 8(2):49-55, 1988.

[21] A. Draeger, S. Engell, and H. Ranke. "Model Predictive Control Using Neural Networks", *IEEE Control Systems Magazine,* 15(5):61-66, 1995.

[22] D. Soloway and P. J. Haley. "Neural Generalized Predictive Control: A Newton-Raphson Implementation", *Proceedings of the 1996 IEEE International Symposium on Intelligent Control,* pp 277-282, 1996.

[23] G. Prasad, E. Swidenbank, and B. W. Hogg. "A Neural Net Model-based Multivariable Long-range Predictive Control Strategy Applied in Thermal Power Plant Control", *IEEE Transactions on Energy Conversion,* 13(2):176-182, 1998.

[24] S. Piche, B. Sayyar-Rodsari, D. Johnson, and M. Gerules. "Nonlinear Model Predictive Control Using Neural Networks", *IEEE Control Systems Magazine*, 20(3):53-62, 2000.

[25] Y-H. Pao. "Process Monitoring and Optimization for Power Systems Applications", *Proceedings of the 1994 IEEE World Congress on Computational Intelligence,* pp 3697-3702, 1994.

[26] R. Cass and B. Radl. "Adaptive Process Optimization Using Functional-link Networks and Evolutionary Optimization", *Control Engineering Practice*, 4(11):1579-1584, 1996.

[27] C. L. Karr. "A Synergistic Architecture for Adaptive, Intelligent Control System Development", *Proceedings of the 2000 IEEE Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, pp 2986-2991, 2000.

[28] H. Ghezelayagh and K. Y. Lee. "Intelligent Predictive Control of a Power Plant with Evolutionary Programming Optimizer and Neuro-Fuzzy Identifier", *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp 1308-1313, 2002.

[29] S. Fan, X. Wang, and Y. Zhou. "Neural Network based Predictive Control for Active Power Filter", *Proceedings of the 2004 IEEE Annual Conference of the Industrial Electronics Society*, pp 822-826, 2004.

[30] N. Chaiyaratana and A. M. S. Zalzala. "Hybridisation of Neural Networks and Genetic Algorithms for Time-Optimal Control", *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pp 389-396, 1999.

[31] L-H. Chen and C-H. Chiang. "An Intelligent Control System Based on Multiobjective Genetic Algorithms and Fuzzy Neural Network", *Proceedings of the 2002 IEEE International Conference on Systems, Man, and Cybernetics,* 2002.

[32] K. Laabidi and F. Bouani. "Genetic Algorithms for Multiobjective Predictive Control", *Proceedings of the 2004 IEEE International Symposium on Control, Communications, and Signal Processing,* pp 149-152, 2004.

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

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

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

[36] 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.

[37] F. Xue, R. Subbu, and P. Bonissone, "Locally Weighted Fusion of Multiple Predictive Models," *Proceedings of the 2006 IEEE World Congress on Computational Intelligence*, July 16-21, 2006, Vancouver, BC, Canada.

[38] Y. Jin, "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation," *Soft Computing*, 9(1):3—12, 2005.

[39] A. Patterson, P. Bonissone, and M. Pavese, "Six Sigma Quality Applied Throughout the Lifecycle of an Automated Decision System", *Journal of Quality and Reliability International,* 21:275-292, 2005.

[40] P. Bonissone, "Multi-Criteria Decision-Making: The Intersection of Search, Preference Tradeoff, and Interaction Visualization Processes", *Proceedings of the First IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007)*, April 1-5, 2007.