Efficient Computer Experiment-Based Optimization through Variable Selection

Dachuan T. Shih
Conifer Health Solutions
2401 Internet Boulevard
Frisco, TX 75034 USA

Seoung Bum Kim
School of Industrial Management Engineering
Korea University
Seoul, Republic of Korea

Victoria C. P. Chen (correspondence author) and Jay M. Rosenberger
Department of Industrial and Manufacturing Systems Engineering
The University of Texas at Arlington
Campus Box 19017
Arlington, TX 76019-0017 USA
E-mail: vchen@uta.edu

Venkata L. Pilla
American Airlines
4333 Amon Carter Blvd.
MD 5358
Fort Worth, TX 76155 USA

COSMOS Technical Report 07-02

Abstract

A computer experiment-based optimization approach employs design of experiments and statistical modeling to represent a complex objective function that can only be evaluated pointwise by running a computer model. In large-scale applications, the number of variables is huge, and direct use of computer experiments would require an exceedingly large experimental design and, consequently, significant computational effort. If a large portion of the variables have little impact on the objective, then there is a need to eliminate these before performing the complete set of computer experiments. This is a variable selection task. The ideal variable selection method for this task should handle unknown nonlinear structure, should be computationally fast, and would be conducted after a small number of computer experiment runs, likely fewer runs (n) than the number of variables (p). Conventional variable selection techniques are based on assumed linear model forms and cannot be applied in this "large p and small n" problem. In this paper, we present a framework that adds a variable selection step prior to computer experiment-based optimization, and we consider data mining methods, using principal components analysis and multiple testing based on false discovery rate, that are appropriate for our variable selection task. An airline fleet assignment case study is used to illustrate our approach.

Keywords: Computer experiments, False discovery rate, Large-scale optimization, Regression trees, Variable selection.

1 Introduction

Complex systems are challenging to optimize. One practical approach is the design and analysis of computer experiments (DACE, for a review see Chen et al. 2006). In DACE, an experimental design is used to organize a set of computer experiment runs, so as to enable fitting of a statistical "metamodel" that approximates the complex system's output from the computer experiment. A mathematical programming technique then determines the system inputs that optimize the metamodel. In traditional DACE, the computer experiment runs a simulation (Kleijnen 2005, Sacks et al. 1989). DACE-based optimization has been used to solve multi-stage mathematical programming problems. This approach has been successfully applied for value function approximation in stochastic dynamic programming and Markov decision processes (Chen 1999, Chen et al. 1999, 2003, Tsai et al. 2004), where the computer experiment runs an optimization computer model that provides a point on the value function. More recently, a DACE-based approach has been developed for two-stage stochastic programming (Pilla 2006, Pilla et al. 2008).

The research in this paper was motivated generally by the high-dimensionality of many real world complex systems and, in particular, by the two-stage stochastic programming problem studied by Pilla et al. (2008). The optimization runs provide the data to build the metamodels, where a higher-dimensional input space requires more runs, and, consequently, more computational effort. If, in fact, an accurate metamodel of the output of the optimization requires all input dimensions, then there is nothing we can do to reduce the number of runs. However, if an accurate metamodel can be constructed using a subset of the input dimensions, then the experimental design can focus on this subset, and consequently

can require a smaller number of optimization runs. In the case of the Pilla (2006), only 42 out of 1264 input variables were included in a metamodel that achieved an R^2 of 99.459%. To identify unimportant variables, this paper adds a Data Mining (DM) Phase to conduct variable selection prior to DACE modeling.

The key is to identify a reasonable subset of the input variables. This is a variable selection task. In recent years, variable selection has received considerable attention in various areas for which data sets with tens thousands of variables are available, including signal/image processing, bioinformatics, process monitoring, and text mining (Jain et al. 2000, Guyon and Elisseeff 2003, Kim et al. 2008, Temiyasathit et al. 2009). The main objective of variable selection is to identify a subset of the variables that are most predictive or informative of a given response (or output) variable. Further, successful implementation of variable selection simplifies the entire modeling process and, thus, reduces computational and analytical efforts. Variable selection is particularly of interest when the number of candidate explanatory variables is large, and many redundant or irrelevant variables are thought to be present.

Conducting variable selection prior to DACE modeling still requires some number of optimization runs (n). In our approach, we severely limit n, so that our data set contains fewer runs than input variables (p). The contribution of our paper is twofold: (1) a general framework for large-scale optimization using data mining variable selection and DACE (DM-DACE), and (2) two new methods for variable selection in the case of large p and small n. Both variable selection methods employ a multiple testing procedure based on false discovery rate (FDR, Benjamini and Hochberg 1995). Our first version uses regression trees as a pre-processing step, prior to running the multiple testing procedure. The second version reverses the roles of the variables in the original testing procedure. Since DACE-based approaches for large-scale optimization already exist, this paper focuses on the DM Phase of the DM-DACE framework.

The rest of this paper is organized as follows. In the next section, our DM-DACE framework is presented, first with a *general* description, and then followed by two example implementations for large-scale optimization, one for stochastic programming and the other for stochastic dynamic programming. In Section 3, we describe the airline fleet assignment application that motivated this study. Section 4 briefly introduces the concept of principal component analysis, a widely used dimensionality reduction method. In Section 5, we present our new variable selection approaches using FDR. Section 6 describes the experimental results, followed by concluding remarks in Section 7.

2 Data Mining and DACE (DM-DACE) Framework

For modeling a performance function in optimization, such as an objective function or a value function, our DACE-based research has focused on the use of multivariate adaptive regression splines (MARS, Friedman 1991, Tsai and Chen 2005, Shih et al. 2006). In large-scale optimization applications, the number of variables in a DACE-based approach can initially be very large. Although one could simply attempt a MARS approximation over these high-dimensional spaces, typically, many variables have little effect on the performance measure. Thus, a data mining step to conduct variable selection is essential to reduce the

number of runs in the optimization computer experiment. The study presented in this paper tests the use of a multiple testing procedure based on false discovery rate for variable selection.

Figure 1 diagrams our DM-DACE approach. This is a two-phase approach, first conducting data mining for variable selection, and then conducting DACE for metamodeling. To conduct the DM Phase:

1. We start in Figure 1 at the top with all possible input variables ($\boldsymbol{x}$), where we require a method that can identify a point as feasible or infeasible. The feasible region is defined by the constraints of the optimization problem.

2. We then move to next box in Figure 1, where the feasibility checker is used in conjunction with design of experiments to select a *small* set of feasible design points. Here we assume an extremely large set of possible inputs and a design with significantly fewer points than inputs (i.e., large p and small n). The generation of a feasible design is in itself a research problem that will depend on a specific application.

3. The (feasible) experimental design is then used to conduct the optimization computer experiment, in which a computer model is run for each design point, and the output is recorded. For the DM Phase, it is noted that the computer model employed here could be representing functions that are related to the optimization, but do not directly require running the optimization. An example of this is discussed for stochastic dynamic programming in Section 2.2. For the stochastic programming example in Section 2.1, the computer model requires running the optimization for both the DM and DACE Phases.

4. Finally, in the bottom box in Figure 1, the DM Phase conducts variable selection to identify a subset of input variables for use in the DACE Phase.

Figure 1 about here.

Once the subset of important inputs has been identified, the process in Figure 1 repeats for the DACE Phase, with some slight variations. In step 1, some modification of the feasibility checker may be needed to handle the subset of input variables. In step 2, the training sample size of the experimental design will likely have more points than the number of input variables in the subset in order to obtain a sufficiently accurate metamodel. In step 3, an actual optimization computer model must be run, as opposed to the option of using related modeling mentioned above. Finally, step 4 changes to the DACE goal of building an accurate metamodel for the optimization computer model. This metamodel is then employed in the objective function of a larger optimization problem. More specifics are given in the next two subsections for stochastic programming and stochastic dynamic programming.

In the DACE Phase, because the set of design points must typically be larger than the number of inputs in the subset, if the DM Phase is not conducted first, then a unnecessarily large design may be generated, wasting computational effort in the optimization. By conducting the DM Phase prior to the DACE Phase, we can ensure the importance of the inputs included in the design, and construct a size-appropriate design.

The methods in Section 5 focus on variable selection in the DM Phase. Given the general setup in Figure 1, we require computationally efficient methods that can handle unknown nonlinear structure and a data set with large p and small n, where the points are spread over a feasible region, as opposed to the circular or rectangular regions of typical designs. Specifically, we consider principal component analysis and two new versions of a multiple testing procedure based on false discovery rate. The original version using false discovery rate was introduced by Benjamini and Hochberg (1995). Our first version uses regression trees as a pre-processing step, prior to running the multiple testing procedure. The second version reverse the roles of the variables in the original testing procedure.

In particular, this research problem and the resulting methods emerged as a consequence of an airline fleet assignment problem studied by Pilla (2006), Pilla et al. (2008) that uses two-stage stochastic programming. This application is the case study for our methods and is described in Section 3. While the fleet assignment problem is a specific application, the optimization involves stochastic and integer programming methods which are applicable to many real world decision-making problems. Thus, our methods additionally handle input variables that are mostly binary (0 or 1), representing integer solutions, but also have values in between, representing fractional solutions.

The results in Pilla (2006) state that the traditional Benders approach required about 3.26 days on a Dual 2.8-GHz Intel Xeon Workstation to solve the fleet assignment problem, while the DACE-based approach with 3562 design points for 1264 decision variables required 3.10 days, where 2.5 days of that time was spent executing the optimization computer experiment for all the experimental design points. The resulting optimal solution was not degraded, demonstrating the promise of the DACE-based approach. However, the requirement of several days to solve this optimization is not at all acceptable, which is why a two-stage fleet assignment model is not optimized in practice. The work of Sherali and Zhu (2008) which uses a traditional approach, corroborates this computational challenge, and it should be noted that they did not solve the problem to optimality because of this challenge. The traditional approach cannot easily overcome this computational intractability; however, there are two basic ways to reduce the computation of the DACE-based approach: (1) use parallel computing (which is not an option for the traditional approach) or (2) use a smaller experimental design. Our current paper considers the second direction for reducing computation since a reduction in the number of important decision variables translates to an ability to use a smaller experimental design. To justify the potential for dimension reduction, we analyzed the MARS approximation of Pilla et al. (2008) and found that it employed only 42 of the 1264 decision variables. Hence, for this problem, the DACE approach could be much more efficient without adversely affecting the solution.

A similar phenomenon was witnessed in a DACE-based study of precursor nitrogen oxides that lead to ozone pollution (Yang et al. 2007). This study employed a stochastic dynamic programming framework, and it was critical to reduce the state space dimensionality. Many other potential applications arise in large-scale problems that can be modeled via stochastic programming or stochastic dynamic programming, but cannot be solved due to the computational intractability of traditional solution approaches. Included among these are revenue management problems, environmental decision-making problems, large-scale logistics problems, airport operations, and health care. To motivate and elucidate our DM-DACE framework, we provide below a brief overview of two-stage stochastic programming

and continuous-state stochastic dynamic programming and describe the DM-DACE process for these types of optimization problems.

2.1 Two-Stage Stochastic Programming

A two-stage stochastic linear program can be generally formulated as:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{c}^T\boldsymbol{x} + E\left\{\min_{\boldsymbol{y}} \boldsymbol{g}^T\boldsymbol{y}\right\} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, \\ & T\boldsymbol{x} + W\boldsymbol{y} = \boldsymbol{h}, \\ & \boldsymbol{x} \geq 0, \boldsymbol{y} \geq 0, \end{aligned}$$

where $\boldsymbol{x} \in R^{n_1}$ is the first-stage decision vector with linear costs $\boldsymbol{c} \in R^{n_1}$, $\boldsymbol{y} \in R^{n_2}$ is the second-stage decision vector with linear costs $\boldsymbol{g} \in R^{n_2}$, A is the $m_1 \times n_1$ first-stage linear constraint matrix with right-hand-side $\boldsymbol{b} \in R^{m_1}$, and T and W are, respectively, $m_2 \times n_1$ and $m_2 \times n_2$ matrices specifying the second-stage linear constraints on $\boldsymbol{x}$ and $\boldsymbol{y}$ with right-hand-side $\boldsymbol{h} \in R^{m_2}$. The expectation is taken over stochastic variables that may appear in $\boldsymbol{g}$, T, W, or $\boldsymbol{h}$. For a given realization of the stochastic variables, call it $\boldsymbol{\omega}$, we can write the second-stage recourse function as

$$Q(\boldsymbol{x}, \boldsymbol{\omega}) = \min_{\boldsymbol{y}} \left\{\boldsymbol{g}(\boldsymbol{\omega})^T\boldsymbol{y} \mid W(\boldsymbol{\omega})\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{\omega}) - T(\boldsymbol{\omega})\boldsymbol{x}, \boldsymbol{y} \geq 0\right\}. \tag{1}$$

Here we can see that the second-stage decision depends directly on the first-stage decision. Then the expected second-stage recourse function is

$$Q(\boldsymbol{x}) = E[Q(\boldsymbol{x}, \boldsymbol{\omega})],$$

where the expectation is taken over scenario realizations $\boldsymbol{\omega}$. Finally, the first-stage decision is found by solving the deterministic linear program:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{c}^T\boldsymbol{x} + Q(\boldsymbol{x}) \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{g}, \\ & \boldsymbol{x} \geq 0. \end{aligned} \tag{2}$$

The difficulty lies in determining $Q(\boldsymbol{x})$. If the stochastic variables are continuous, then a large number of scenarios may be needed to estimate the expectation. The second-stage recourse function in equation (1) must be solved individually for each scenario $\boldsymbol{\omega}$. Thus, there is a high computational cost for "evaluating" $Q(\boldsymbol{x})$ at just one $\boldsymbol{x}$.

Traditional two-stage stochastic programming algorithms, such as Benders' approach or the L-shaped method (Birge and Louveaux 1997), use the following approach:

Restricted Master Problem (RMP) Step: Optimize the first-stage decisions ($\boldsymbol{x}$) based upon a relaxed representation of the second-stage recourse problem in equation (1).

Subproblem Step: Based upon the first-stage solution, optimize the second-stage recourse decisions ($\boldsymbol{y}$) over the scenarios. If the relaxed representation of the second-stage recourse problem is consistent with the second-stage solution, terminate the algorithm. Otherwise, revise the relaxed representation of the second-stage recourse problem and return to the RMP Step.

In solving the minimization in equation (2), each evaluation of $Q(\cdot)$ is computationally expensive. Consequently, the traditional iterative approximation methods can be very slow to converge for large-scale problems.

As an alternative to the traditional approach, a DACE-based algorithm for two-stage stochastic programming uses the following steps (Pilla 2006, Pilla et al. 2008):

DACE Step: Use design of experiments to represent potential first-stage solutions ($\boldsymbol{x}$ points). Pilla et al. (2008) describes a method for identifying feasible first-stage solutions for the airline fleet assignment case. For each experimental design point (which specifies a potential first-stage solution $\boldsymbol{x}$), optimize the second-stage recourse problem in equation (1) over the scenarios. Use a statistical model to approximate the expected second-stage recourse function $Q(\boldsymbol{x})$.

Optimization Step: Optimize the first-stage using the statistical model to represent the optimized second-stage recourse problem.

For the commercial airline fleet assignment case study in Pilla et al. (2008), the set of 6537 first-stage decision variables was first reduced to 1264 by eliminating clearly redundant variables via various constraints. Then 3562 feasible design points were used to conduct the optimization computer experiment over a 1264-dimensional decision space, and a MARS approximation was fit to approximate the expected second-stage recourse function $Q(\boldsymbol{x})$ of a commercial airline fleet assignment problem. As mentioned earlier, it was observed that the MARS function included only a small subset of the 1264 first-stage decision variables. If it was known in advance that, for example, 800 of the 1264 variables were not important, then the experimental design created over the 464-dimensional space of the remaining variables could use one-third as many design points and correspondingly reduce the computational effort. However, since it is not possible to know in advance which variables to eliminate, the DM Phase of our DM-DACE process uses a small ($n < p$) experimental design and conducts data mining for variable selection to identify which variables to eliminate prior to implementing the DACE Phase as developed by Pilla et al. (2008).

2.2 Continuous-State Stochastic Dynamic Programming

Dynamic programming is used to optimize decisions for a system that is changing over time (Bellman 1957). State variables represent the state of the system as it moves through time, and decision variables are controlled to optimize the system. Problems can involve discrete time stages or continuous time, finite or infinite time horizons, discrete and/or continuous state (or decision) variables, and stochasticity; and modern numerical solution methods fall under the research area of approximate dynamic programming (Powell 2007). For small-scale discrete-state problems, the optimal solution can be tabulated for all possible states. For large-scale or continuous-state problems, this is not possible. Chen et al. (1999) first introduced a statistical perspective of finite-horizon, continuous-state stochastic dynamic

programming, which seeks to minimize expected costs over T time stages, i.e., to solve

$$\begin{array}{ll} \min\limits_{\boldsymbol{u}_1,\ldots\boldsymbol{u}_T} & E\left\{\sum\limits_{t=1}^{T} c_t(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\epsilon}_t)\right\} \\ \text{s.t.} & \boldsymbol{x}_{t+1}=f_t(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\epsilon}_t),\ \text{for } t=1,\ldots,T-1 \text{ and} \\ & (\boldsymbol{x}_t,\boldsymbol{u}_t)\in\Gamma_t,\ \text{for } t=1,\ldots,T \end{array}$$

where $\boldsymbol{x}_t \in R^n$ is the state vector, $\boldsymbol{u}_t \in R^m$ is the decision vector, $c_t : R^{n+m+l} \to R^1$ is a known cost function for period t, $\Gamma_t \subset R^{n+m}$ is the set of constraints on $\boldsymbol{u}_t$ which depend on $\boldsymbol{x}_t$, and the expectation is taken over the random vector $\boldsymbol{\epsilon}_t \in R^l$, with known probability distribution. The known function f_t defines the state transition from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$ by $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\epsilon}_t)$. The future value function at time t is

$$\begin{array}{rll} V_t(\boldsymbol{x}_t) = & \min\limits_{\boldsymbol{u}_t,\ldots\boldsymbol{u}_T} & E\left\{\sum\limits_{\tau=t}^{T} c_\tau(\boldsymbol{x}_\tau,\boldsymbol{u}_\tau,\boldsymbol{\epsilon}_\tau)\right\} \\ & \text{s.t.} & \boldsymbol{x}_{\tau+1}=f_t(\boldsymbol{x}_\tau,\boldsymbol{u}_\tau,\boldsymbol{\epsilon}_\tau),\ \text{for } \tau=t,\ldots,T-1 \text{ and} \\ & & (\boldsymbol{x}_\tau,\boldsymbol{u}_\tau)\in\Gamma_\tau,\ \text{for } \tau=t,\ldots,T-1, \end{array}$$

for $t = 1, \ldots, T$, and can be written recursively as

$$V_t(\boldsymbol{x}_t) = \min_{\boldsymbol{u}_t} E\{c_t(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\epsilon}_t) + V_{t+1}(\boldsymbol{x}_{t+1})\}, \quad t=1,\ldots,T, \tag{3}$$

where $V_{T+1} \equiv 0$. If the expected value cannot be computed exactly, then a set of scenarios, as discussed for stochastic programming, may be used to estimate the expected value. In theory, a backward solution approach can be used, where the future value functions are obtained in backwards order from $V_T(\boldsymbol{x}_T)$ to $V_1(\boldsymbol{x}_1)$ using the recursion.

The statistical perspective of Chen et al. (1999) views the future value functions $V_t(\cdot)$ as unknown relationships with the state variables ($\boldsymbol{x}_t$). Following a DACE process in each time stage t, an experimental design is used to "discretize" the state space, the optimization is solved at the design points, and then a statistical model approximates the future value function for that stage. They studied applications with up to 9 state variables (which were the largest at that time). Cervellera et al. (2006) later used this same statistical concept to solve a 30-dimensional problem. Yang et al. (2007) and Yang et al. (2009) were first to utilize variable selection to reduce the state space dimensionality. Their air quality case study employed a highly computational air quality simulation model and involved over 500 state variables, but their solution method only required up to 25 state variables in each time stage. The computer model in their DM Phase did not directly conduct the optimizations to yield the future value functions $V_t(\cdot)$. Instead, they used the air quality simulation model to study relationships for the costs $c_t(\cdot)$ and state transitions $f_t(\cdot)$ as a function of state variables $\boldsymbol{x}_t$ and decision variables $\boldsymbol{u}_t$ with error variation $\boldsymbol{\epsilon}_t$. Their DM Phase used least-squares regression, which requires more observations than unknown parameters ($n > p$), to conduct variable selection. The variable selection methods for $n < p$ developed in this paper could reduce the number of expensive computer simulation runs and optimization runs, and consequently reduce the total computational effort of the DM-DACE process.

3 Airline Fleet Assignment Application

Optimization plays an important role in airline planning and operations, including revenue management (McGill and van Ryzin 1999), crew-scheduling (Gopalakrishnan and Johnson 2005), and fleet assignment (Sherali et al. 2006). Given the financial difficulties of airlines in recent years, further approaches to decrease costs and increase revenues are needed. The complexity of airline planning (crew, fleet, maintenance) requires airlines to fix their schedules far in advance of actual flight departures. Specifically, these plans must be locked down at least 45–60 days prior to departure, and many airlines publish their schedule 90 days prior. Since most travelers do not purchase their airline tickets that far in advance, there is large uncertainty in this planning. Ideally, an airline would like to match their capacities with their demands, and the significant body of airline optimization research continues to seek solutions. Most domestic airlines use a hub-and-spoke network in which nearly all flight legs either depart from or arrive at a small subset of stations. Consequently, planes can be rerouted relatively easily allowing for smooth transitions into new fleet assignments.

In this paper, we consider a formulation of the fleet assignment model that uses a two-stage stochastic programming framework along with the Boeing concept of demand driven dispatch. The complete optimization formulation from Pilla et al. (2008) is presented in the appendix. Airline fleet assignment models are used to assign aircraft to the scheduled flights in order to maximize profit (revenue − cost). Demand driven dispatch, introduced by Berge and Hopperstad (1993), relies on the concept of crew-compatible aircraft that have identical cockpits. This allows an airline to swap aircraft without swapping crews, so as to capture more revenue by matching capacities more closely with demands while avoiding disruption of the already complex problem of crew scheduling. For example, Boeing 757 and 767 models are crew-compatible, but a 767 can fly more passengers than a 757, so we can use these different capacities to better match demand. A family of crew-compatible aircraft consists of all aircraft types with a particular cockpit. The two-stage stochastic programming approach was studied by Sherali and Zhu (2008) and Pilla (2006), where the first stage occurs when the flight schedule is published (e.g., 90 days prior to departure) and the second stage occurs closer to departure (e.g., one to two weeks prior) when most of the demand has been realized. In the first stage, crew-compatible families are assigned to flights, and in the second stage the actual aircraft within the families are assigned to best match demand. The goal of the two-stage formulation is to assign crew-compatible families in the first stage, so as to maximize the demand capturing potential in the second stage, given a specific crew-compatible family assignment from the first stage, the stochastic program considers several demand realizations (scenarios) in the second stage, and the average over these realizations estimates the expected profit for that first-stage assignment. This expected profit function is known to be concave over the space of first-stage assignments.

Instead of employing a Benders' approach, Pilla (2006) and Pilla et al. (2008) developed a DACE approach to reduce the computation involved in conducting the optimization. Their DACE Phase uses first-stage constraints in a multi-step process to construct an experimental design within the feasible region, then builds a statistical model that approximates the expected profit function in the first stage of the stochastic program. An Optimization Phase then solves the two-stage problem using the DACE expected profit approximation instead of solving many second-stage subproblems in every iteration. This greatly speeds up the

optimization, compared to Benders', because the computation of the subproblems is shifted to the DACE Phase. Overall, Pilla (2006) found that the total computational effort, including the DACE Phase, was lower for his approach compared to a Benders' approach. However, further speedup may be achieved by conducting a DM Phase prior to the DACE Phase.

The input variables in the DM and DACE Phases for the fleet assignment application are the first-stage assignment variables, and there is one variable for every possible crew-compatible family and flight combination. For example, there is one variable that considers the assignment of crew-compatible family g with flight f, and this variable is 1 if family g is assigned to flight f, otherwise, it is 0. An optimal solution yields assignment variables that are only 0 or 1; however, most solution procedures make use of fractional solutions in continuous relaxations of the assignment problem. These fractional solutions can have a practical interpretation: if the assignment variable for family g and flight f is 0.4, then the airline would be assigning family g to flight f 40% of the time. The challenge for the DM Phase is to conduct meaningful variable selection for this unusual type of input variable that is quite common in real world optimization problems.

4 Principal Component Analysis

Principal component analysis (PCA) is a multivariate statistical method that extracts new variables, called principal components (PCs) through an orthogonal transformation of the original variables (Jolliffe 2002). PCA has been widely used in a variety of applications for dimensionality reduction (for example, Yeung and Ruzzo 2001, Jackson 1991, Wise et al. 1990). Let the random vector $\boldsymbol{X} = [X_1, X_2, \cdots, X_P]^T$. PCA relies on the eigenvalue-eigenvector decomposition of the covariance matrix of $\boldsymbol{X}$, $\boldsymbol{C_X}$. Denote the pairs of eigenvalues and eigenvectors of $\boldsymbol{C_X}$ be $(\lambda_1, \boldsymbol{E}_1), (\lambda_2, \boldsymbol{E}_2), \cdots, (\lambda_P, \boldsymbol{E}_P)$, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_P$. The transformation process extracts the first PC that maximizes the variance of $Z_1 = \boldsymbol{a}_1^T \boldsymbol{X}$ subject to $|\boldsymbol{a}_1| = 1$. Using some properties of linear algebra, a projection vector $\boldsymbol{a}_1$ is $\boldsymbol{E}_1$. Thus, the first PC is a linear combination of $\boldsymbol{X}$, $Z_1 = \boldsymbol{E}_1^T \boldsymbol{X}$ and captures the maximum variance of $\boldsymbol{X}$. Likewise, the second PC is obtained by the maximization of $Z_2 = \boldsymbol{a}_2^T \boldsymbol{X}$ subject to $|\boldsymbol{a}_2| = 1$ and Cov(Z_1,Z_2)=0. This process is repeated P times to extract P PCs. The extracted PCs are uncorrelated to each other.

PCA is efficient for reducing high-dimensional data if only the first few PCs are needed to represent most of the variability in the entire data. Moreover, the PCA process depends solely upon the covariance matrix of $\boldsymbol{X}$ and thus, does not require any distributional assumptions. However, PCA has several limitations. First, the extracted PCs cannot be readily interpreted because they are linear combinations of a large number of original variables. Second, the PCs may not produce the best predictability on the response variable (Y) since the transformation process of PCA relies solely on $\boldsymbol{X}$ and ignores Y information. Third, there is no objective way to determine the number of PCs to retain.

We would like to clarify the distinction between variable extraction and variable selection, although much of the literature fails to make a clear distinction. PCA is an example of variable extraction, while FDR-based multiple testing procedures are examples of variable selection. Variable extraction techniques create new variables based on transformations of the original variables to extract useful information for the model. Variable selection methods,

on the other hand, find the best subset of the given original variables that leads to the best prediction. Variable extraction tends to reduce more dimensionality than variable selection, but suffers from the lack of interpretability with respect to the original variables. The choice between the two depends upon the purpose of the application problem. For the airline fleet assignment problem, maintaining the physical interpretation of the original variables is as important as aggressive dimensionality reduction.

5 Proposed Approaches for Variable Selection

5.1 Multiple testing procedure controlling false discovery rate

In this paper, we propose to use a multiple testing procedure that controls false discovery rate (FDR) to reduce the dimensionality of the original data. We begin with a brief introduction of the multiple testing procedure. Suppose through some statistical modeling processes, we have a collection of hypothesis tests and the corresponding p-values $\{p_i\}_{i=1}^{P}$, where p_i is the p-value of testing the null hypothesis, and P is the number of variables. In the literature, it is standard to choose a p-value threshold τ and declare the variable X_i significant if and only if the corresponding p-value $p_i \leq \tau$. A common approach in multiple testing nowadays is the false discovery rate (FDR) procedure (Benjamini and Hochberg 1995) because it is well-known that we need to adjust the significance level when conducting multiple tests and that the conventional procedures that control family-wise error rate (FWER; e.g., Bonferroni) are too conservative to detect true significant variables. The FDR is defined as the expected proportion of false positives (falsely rejected hypotheses) among all the hypotheses rejected (Benjamini and Hochberg 1995). Many FDR studies have revealed that FDR-based procedures find as many significant hypotheses as possible while keeping a relatively small number of false positives (for example, Kim et al. 2006, Efron 2004, Storey and Tibshirani 2003).

The FDR-based procedure to identify significant variables is as follows: Consider a series of hypotheses, p-values, and ordered p-values, denoted H_i, p_i, and $p_{(i)}$, respectively.

- Step 1: Choose a fixed α, where $0 \leq \alpha \leq 1$.
- Step 2: Find $\hat{i} = \max\left[i : p_{(i)} \leq \frac{i}{m} \cdot \frac{\alpha}{\pi_0}\right]$, where $\pi_0 (= \frac{m_0}{m})$ denote the proportion of true H_i, m_0 is the number of true H_i, and m is the total number of hypotheses.
- Step 3: If $\hat{i} \geq 1$, $\Omega = \{\text{All rejected } H_i \text{ with } p_i \leq p_{(\hat{i})}\}$ with $\text{FDR}(\Omega) \leq \alpha$.

  If $\hat{i} = 0$, do not reject any hypothesis since $\Omega = \emptyset$.

In general, $\pi_0 = 1$ is the most conservative possible choice. Thus, we use $\pi_0 = 1$ in this paper. For more details of the choice of π_0, refer to Efron (2004) and Storey and Tibshirani (2003).

Th original FDR procedure proposed by Benjamini and Hochberg (1995) assumes that all hypotheses are independent. However, later work by Benjamini and Yekutieli (2001) revealed that the procedure still holds when the hypotheses are positively correlated. In their paper, they also proved that the procedure can handle any correlation structure by replacing π_0 in step 2 with $\sum_{i=1}^{P} \frac{1}{i} \approx log(P) + 0.5$.

5.2 FDR-based variable selection from regression trees

Generally, a conventional FDR procedure for variable selection requires a categorical response variable that separates the data into c groups, where c is the number of categories. For each input variable, we test for differences in the c samples, using a t-test or F-test. However, because the response variable generated by a computer experiment is continuous in most cases, we need to categorize the original response. A mean or median value of the response variable can be used to separate the response variable into two groups, high and low, if the response surface is monotonic. However, if the relationship between the response and the inputs is not monotonic, such that the separation by high and low values does not make sense, then alternate grouping strategies are needed. Many methods are available to categorize the continuous values (for example, Elomma and Rousu 2002, Fayyad and Irani 1992). However, no consensus exists about which of them best satisfies all conditions. In the present study, we propose a new and simple strategy that uses binary regression trees to partition the response observations into meaningful groups.

An algorithm constructing binary regression trees partitions the input space into two regions using the input variable and a splitting-point to fit a piecewise-constant model that achieves the best fit (Mitchell 1997). This partitioning process is repeated to one or both of these regions until a termination criterion has been reached. Based on the terminal nodes of regression trees, the response values can be separated into a certain number of disjoint groups. Then, an FDR procedure can be applied for variable selection using grouping indices. Note that for three or more groups, an analysis of variance (ANOVA) table is constructed for each input variable and its significance is tested using an F-test. This approach simultaneously takes advantage of regression trees and an FDR procedure. The possible drawback of this approach is that we may lose the original characteristics of a continuous response by grouping its values.

5.3 Reverse FDR

In order to maintain the continuous characteristic of a response variable in an FDR procedure, we propose a new FDR approach for variable selection, called reverse FDR. Each input variable now serves as a categorical variable to group the response values for testing (by a t-test or an ANOVA F-test). The main idea is to create a set of new variables, one corresponding to each original input, by grouping the response variable based on the categories of each input variable and conducting an FDR procedure on these new variables. This is analogous to the resampling technique because each new variable is re-sampled from the set of original response values based on the categories of each input variable. Our proposed approach is similar to the original FDR procedure, except that the hypothesis test is conducted on the continuous response grouped by each input variable, as opposed to testing the continuous input variables grouped by the response values. This process explains why the proposed approach is called reverse FDR.

The setting and procedure for reverse FDR is as follows:

- Step 1: For each input variable, divide the response variable into c groups based on the categories of the input variable.

- Step 2: For each input variable, conduct a statistical test (e.g., two-sample t test, ANOVA F-test) on its corresponding set of response variable groups, and record the p-value.
- Step 3: Use the p-values to conduct an original FDR procedure (Benjamini and Hochberg 1995) or a correlated version of the FDR procedure (Benjamini and Yekutieli 2001) that identifies which input variables are statistically significant.

If the response surface is known to be convex or concave, a common occurrence in optimization, then reverse FDR with $c = 3$ groups should be sufficient. Our motivating example, the fleet assignment application has concave nonlinearity, and the decision variables fall into one of three categories, 0, 1, or between 0 and 1. The reverse FDR approach is particularly appropriate for this case since the input variables are easily categorized and the response variable is continuous.

6 Experimental Results

For a real airline network with 50 stations and 2358 legs, the initial decision space involved 6537 decision variables across the aircraft types. Pilla et al. (2008) was able to reduce this to 1264 dimensions by combining aircraft types into crew-compatible families and using implicit equalities specified by the constraints (where a combination of constraints leads to perfect correlation between some decision variables). In their DACE Phase, the multi-step design of experiments process derived 141 initial extreme points, which were then expanded into 3562 design points in the feasible region. The second-stage subproblem is then solved for each of these design points. Among the 1264 decision variables, there are still many useless ones that could be identified via a DM Phase, enabling a much smaller set of design points. It should be noted that 1264 variables were reduced to 1061 prior to implementation of variable selection approaches. Those 203 (=1264-1061) were dropped because they possess uniform values for almost all 141 observations (i.e., at least 140 values out of 141 are the same).

Using only the subproblem solutions for the 141 initial extreme points, we studied five cases of variable selection: (1) none, (2) PCA,(3) FDR on three groups identified by regression trees (FDR level=0.01), (4) Reverse FDR (FDR level=0.01), and (5) Correlated version of reverse FDR (FDR level=0.01). The resulting numbers of variables selected are given in Table 1.

Table 1 about here.

For PCA, we utilized MATLAB (www.mathworks.com), and 140 PCs were identified whose eigenvalues were greater than zero. In other words, the extracted 140 PCs explain 100% of the variability of the entire data.

We used regression trees to group the response variables. Here we forced the regression tree to make three terminal nodes (Mitchell 1997). Although "three" is not the optimal number of terminal nodes in terms of a regression tree fit, the objective here is simply to categorize a non-monotonic response. Having found grouping information from the regression tree, we used the FDR procedure (FDR=0.01) and found 454 significant variables. Finally, the reverse FDR and the correlated version of reverse FDR methods with $c = 3$ groups

selected 326 and 256 significant variables, respectively. It should be noted that all three methods are computationally efficient, requiring only about 3 minutes.

Interpretation of the FDR results can be made as follows: For example, 326 variables were identified by reverse FDR, implying that there are, on average, 3 to 4 ($3.26 = 326 \times 0.01$) variables falsely identified as significant (false discoveries) out of the 326 variables identified as significant. A similar interpretation can be applied for the results from FDR with three groups from regression trees and the correlated version of reverse FDR. Note that all methods achieved significant dimensionality reduction.

Ideally, the next step in the DM-DACE framework is to conduct the DACE Phase with a reduced set of decision variables. This would require a feasible experimental design that focuses on the reduced set of variables, running the second-stage optimization for all the design points, then fitting a MARS approximation. The current paper studies the DM Phase, but the DACE Phase and the subsequent first-stage optimization tasks are topics for future work. It should be noted here that the optimization computer experiment will still require specification of all input variables, even though the experimental design is only specifying the reduced subset of variables. There is more than one way to handle this, and our future work will study this. In theory, since the eliminated variables are unimportant, it should be not matter how they are set; however, their values can still affect the important variables through the constraints.

For model fit comparison purposes, MARS using an automatic stopping rule (Tsai and Chen 2005) was fit over the 3562 design points from Pilla et al. (2008) using the variable sets identified by each of the five methods. MARS is a linear statistical model with a forward stepwise procedure that adds basis functions based on the fit to the data, and the automatic stopping rule terminates the forward procedure when the quality of fit no longer improves. Table 2 displays the resulting number of basis functions and coefficient of determination (R^2) for each of the five methods.

Table 2 about here.

A validation data set of 1600 points was generated to test the MARS approximations, and relative errors were computed using the formula $\frac{|y-\hat{f}|}{y}$, where y and $\hat{f}$ are the actual and fitted response values, respectively. In terms of accuracy, boxplots in Figure 2 show that the maximum relative errors of all methods are less than 1.75×10^{-4}, where the plus signs at the top of the boxplots indicate observations that are more than 1.5 times the interquartile range (the difference between the 75th percentile and 25th percentile of the sample) away from the top of the box. The MARS model constructed with 140 PCs produces a relatively wide interquartile range (box) compared to other methods. This indicates that the extracted 140 PCs do not necessarily lead to good prediction accuracy although they explain 100% of the variability of the entire input data. The models constructed with the variables selected from the two reverse FDR approaches give more compact interquartile ranges with fewer plus signs, and the tree/FDR result, although with many plus signs, is not significantly different from the two reverse FDR methods. Overall, we judge the two reverse FDR methods as providing a MARS approximation that is nearly as good as using all 1061 variables.

Figure 2 about here.

With regard to computational effort, reductions can be easily estimated without completing the DACE Phase and subsequent first-stage optimization. It was mentioned in Section 2 that the DACE-based approach of Pilla (2006) required 3.10 days of computation, where 2.5 days were spent conducting the optimization computer experiments. In our current study, only $n = 141$ of the original 3562 design points (4%) were employed for the DM Phase; hence, the computational effort for the DM Phase computer experiment runs can be estimated as 4% of 2.5 days, which is 2.4 hours. It should be noted that the data from these runs are also used in the DACE Phase, so these runs do not officially add to the total computational effort. The dimension reduction achieved by the correlated version of Reverse FDR was 76%; hence the computational effort for all computer experiment runs can be estimated as 24% of 2.5 days, which is 14.4 hours. This is a reduction of 1.9 days. Additional reduction will also be achieved in the generation of the design points, which originally required 4.5 hours in the DACE Phase. Since only 24% of the points need to be generated, this reduction can be estimated as 76% of 4.5 hours, which is 3.4 hours. The run times for the variable selection methods were about 3 minutes for the FDR based approaches and 19 minutes for PCA, and the run time for building the MARS model in the DACE Phase was about 11 minutes on a Dual 2.6-Ghz Athlon Workstation. These additional computations are cancelled out by other reductions in the DACE Phase due to the smaller experimental design, such as fewer feasibility checks. In summary, we can estimate the total reduction in computation to be more than 2 days.

A few additional remarks on the methods are given below.

Remark 1: A different choice of FDR level leads to a different number of variables selected. A higher level of FDR increases the number of selected variables, which yields larger power but produces more false positives. Similarly, a lower level of FDR decreases the number of false positives but deteriorates power.

Remark 2: For tree/FDR, different groupings from the regression tree can be explored. As mentioned we do not seek an optimal tree in this task; we merely want a data-driven grouping of the response values.

7 Conclusion

We have proposed computationally-efficient variable selection methods within a DM-DACE framework to expedite solving some large-scale complex optimization problems. The main contributions of the present study are the general DM-DACE framework and two new variable selection methods for the DM Phase that use a multiple testing procedure for controlling FDR. The first approach is a simple extension that uses regression trees to group the response values for the FDR-based variable selection procedure. This tree/FDR approach allows a more general representation of continuous response variable values than a mean or median. The second approach, called reverse FDR, is devised to fully utilize the original continuous response variable by switching the roles of the response and input variables. Reverse FDR was additionally developed with a version to handle correlated variables. Both approaches are designed to handle the "large p and small n" problem with an unknown nonlinear response structure.

To test the adequacy of selected variables from our proposed approaches, we applied them to a real airline fleet assignment problem. For comparison, we used relative errors from MARS models fit to the selected variable sets from the different methods. The results demonstrated that the variables selected by our proposed approaches yield nearly as good models as using all variables. Moreover, compared with PCA, one of the most widely used dimensionality reduction techniques, our proposed approaches provide more accurate and robust prediction results. This implies that our approaches adequately select the important variables (i.e., eliminate unnecessary variables). To the best of our knowledge, the present study is the first attempt to reduce computational effort in large-scale DACE-based optimization through appropriate variable selection approaches. Future work will study the impact of the DM Phase on the DACE Phase and subsequent Optimization Phase to solve the larger optimization.

8 Appendix: Airline Fleet Assignment Model Formulation

The optimization formulation from Pilla et al. (2008) is reproduced here for the readers' reference.

Let L be the set of flight legs (indexed by l). Let F denote the set of fleet types (indexed by f), and G be the set of crew-compatible families (indexed by g), which can be used for each of the legs $l \in L$. Since we assign crew-compatible families in the first stage, for each leg $l \in L$ and for each crew-compatible family type $g \in G$, let a binary variable x_{gl} be defined such that

$$x_{gl} = \begin{cases} 1 & \text{if crew-compatible family } g \text{ is assigned to flight leg } l, \\ 0 & \text{otherwise.} \end{cases}$$

In the second stage, we assign specific aircraft within the crew-compatible family. As such, for each leg $l \in L$, for each aircraft type $f \in F$, and for each scenario $\xi \in \Xi$, let a binary variable x^{ξ}_{fl} be defined such that

$$x^{\xi}_{fl} = \begin{cases} 1 & \text{if aircraft type } f \text{ is assigned to the leg } l \text{ in scenario } \xi, \\ 0 & \text{otherwise.} \end{cases}$$

Since a combined FAM and PMM model is used, let the decision variable z^{ξ}_i represent the number of booked passengers for itinerary-fare class i in scenario ξ.

For combined FAM and PMM, consider the following additional parameters:

- S = set of stations, indexed by s,
- I = set of itinerary-fare classes, indexed by i,
- V = set of nodes in the entire network, indexed by v,
- $f(v)$ = fleet type associated with node v,
- A_v = set of flights arriving at node v,

- D_v = set of flights departing at node v,
- M_f = number of aircraft of type f,
- f_i = fare for itinerary-fare class i,
- C_{fl} = cost if aircraft type f is assigned to flight leg l,
- $a_{v^+}^{\xi}$ = value of ground arc leaving node v for scenario ξ,
- $a_{v^-}^{\xi}$ = value of ground arc entering node v for scenario ξ,
- O_f = set of arcs that include the plane count hour for fleet type f, indexed by o,
- L_0 = set of flight legs in air at the plane count hour,
- Cap_f = capacity of aircraft type f,
- D_i^{ξ} = demand for itinerary-fare class i in scenario ξ.

The two-stage formulation can be represented as:

$$\max\ \theta\ = E\left[-\sum_{l\in L}\sum_{f\in F} C_{fl}(x_{fl}^{\xi}) + \sum_{i\in I} f_i z_i^{\xi}\right]$$

$$\text{s.t. } a_{v^-}^{\xi} + \sum_{l\in A_v} x_{f(v)l}^{\xi} - \sum_{l\in D_v} x_{f(v)l}^{\xi} - a_{v^+}^{\xi} = 0 \qquad \forall v\in V, \xi\in\Xi \tag{4}$$

$$\sum_{f\in g} x_{fl}^{\xi} = x_{gl} \qquad \forall l\in L, g\in G, \xi\in\Xi \tag{5}$$

$$\sum_{o\in O_f} a_o^{\xi} + \sum_{l\in L_0} x_{fl}^{\xi} \le M_f \qquad \forall f\in F, \xi\in\Xi \tag{6}$$

$$\sum_{i\ni I} z_i^{\xi} - \sum_{f\in F} Cap_f\ x_{fl}^{\xi} \le 0 \qquad \forall l\in L, \xi\in\Xi \tag{7}$$

$$0 \le z_i^{\xi} \le D_i^{\xi} \qquad \forall i\in I, \xi\in\Xi \tag{8}$$

$$x_{fl}^{\xi} \in \{0,1\} \qquad \forall l\in L, \xi\in\Xi \tag{9}$$

$$x_{gl} \in \{0,1\} \qquad \forall l\in L, g\in G$$

$$a_{v^+}^{\xi} \ge 0 \qquad \forall v\in V, \xi\in\Xi.$$

The objective is to maximize profit (revenue − cost) in the second stage by assigning aircraft within the crew-compatible allocation made in the first stage. The *block time* of a flight leg l is defined as the length of time from the moment the plane leaves the origin station until it arrives at the destination station. Let b_l be the scheduled block time for flight leg l. The cost for each flight leg is calculated as a function of block time and operating cost of a particular fleet type per block hour, and is given by:

$$C_{fl} = b_l * (\text{Operating cost per block hour})_f.$$

Constraints in set (4) represent the balance constraints needed to maintain the circulation of aircraft throughout the network. Cover constraints (5) guarantee that aircraft within the crew-compatible family (assigned in the first stage) are allocated. For formulating the plane count constraints (6), we need to count the number of aircraft of each fleet being used at a particular point of the day (generally when there are fewer planes in the air). As such the *ground arcs* that cross the time line at the plane count hour and the flights in air during that time are summed to assure that the total number of aircraft of a particular fleet type do not exceed the number available. Constraints (7) impose the seat capacity limits, i.e., the sum of all the booked passengers on different itineraries for a flight l should not exceed the capacity of the aircraft assigned and constraint (8) to meet the forecasted demand.

References

Bellman, R. E.: 1957, *Dynamic Programming*, Princeton University Press, Princeton.

Benjamini, Y. and Hochberg, Y.: 1995, Controlling the false discovery rate: a practical and powerful approach to multiple testing, *J. Roy. Statist. Soc. B* **57**, 289–300.

Benjamini, Y. and Yekutieli, D.: 2001, The control of the false discovery rate in multiple testing under dependency, *The Annals of Statistics* **29**, 1165–1188.

Berge, M. E. and Hopperstad, C. A.: 1993, Demand driven dispatch: A method of dynamic aircraft capacity assignment, models and algorithms, *Operations Research* **41**(1), 153–168.

Birge, J. R. and Louveaux, F.: 1997, *Introduction to Stochastic Programming*, Springer, New York, New York.

Cervellera, C., Chen, V. C. P. and Wen, A.: 2006, Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization, *European Journal of Operational Research* **171**, 1139–1151.

Chen, V. C. P.: 1999, Application of MARS and orthogonal arrays to inventory forecasting stochastic dynamic programs, *Computational Statistics and Data Analysis* **30**, 317–341.

Chen, V. C. P., Günther, D. and Johnson, E. L.: 2003, Solving for an optimal airline yield management policy via statistical learning, *Journal of the Royal Statistical Society* **Series C**(52 Part 1), 1–12.

Chen, V. C. P., Ruppert, D. and Shoemaker, C. A.: 1999, Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming, *Operations Research* **47**, 38–53.

Chen, V. C. P., Tsui, K.-L., Barton, R. R. and Meckesheimer, M.: 2006, Design, modeling, and applications of computer experiments, *IIE Transactions* **38**, 273–291.

Efron, B.: 2004, Large-scale simultaneous hypothesis testing: the choice of a null hypothesis, *J. Am. Statist. Assoc.* **99**, 99–104.

Elomma, T. and Rousu, J.: 2002, Fast minimum training error discretization, *Proceedings of the Ninetheenth International Conference on Machine Learning*, Sydney, Australia, pp. 131–138.

Fayyad, U. M. and Irani, K. B.: 1992, On the handling of continuous-valued attributes in decision tree generation, *Machine Learning* **8**(1), 82–102.

Friedman, J. H.: 1991, Multivariate adaptive regression splines (with discussion, *Annals of Statistics* **19**, 1–141.

Gopalakrishnan, B. and Johnson, E. L.: 2005, Airline crew scheduling: State-of-the-art, *Annals of Operations Research* **140**, 305–337.

Guyon, I. and Elisseeff, A.: 2003, An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**, 1157–1182.

Jackson, J. E.: 1991, *A User's Guide to Principal Components*, Wiley, New York, New York.

Jain, A. K., Duin, R. and Mao, J.: 2000, Statistical pattern recognition: A review, *The IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 4–37.

Jolliffe, I. T.: 2002, *Principal Components Analysis*, Springer, New York, New York.

Kim, S. B., Tsui, K. L. and Borodovsky, M.: 2006, Multiple hypothesis testing in large-scale contingency tables: inferring patterns of pair-wise amino acid association in β-sheets, *Journal of Bioinformatics Research and Applications* **2**, 193–217.

Kim, S. B., Wang, Z., Oraintara, S., Temiyasathit, C. and Wongsawat, Y.: 2008, Feature selection and classification of high-resolution nmr spectra in the complex wavelet transform domain, *Chemometrics and Intelligent Laboratory Systems* **90**(2), 161–168.

Kleijnen, J. P. C.: 2005, An overview of the design and analysis of simulation experiments for sensitivity analysis, *European Journal of Operational Research* **164**(2), 287–300.

McGill, J. and van Ryzin, G. J.: 1999, Revenue management: Research overview and prospects, *Transportation Science* **33**, 233–256.

Mitchell, T. M.: 1997, *Machine Learning*, McGraw-Hill, New York, NY.

Pilla, V. L.: 2006, *Robust Airline Fleet Assignment*, PhD thesis, University of Texas at Arlington.

Pilla, V. L., Rosenberger, J. M., Chen, V. C. P. and Smith, B.: 2008, A statistical computer experiments approach to airline fleet assignment, *IIE Transactions* **40**, 524–537.

Powell, W. B.: 2007, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Wiley, Hoboken, NJ.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P.: 1989, Design and analysis of computer experiments (with discussion), *Statistical Science* **4**, 409–423.

Sherali, H. D., Bish, E. K. and Zhu, X.: 2006, Airline fleet assignment concepts, models, and algorithms, *European Journal of Operational Research* **172**, 1–30.

Sherali, H. D. and Zhu, X.: 2008, Two-stage fleet assignment model considering stochastic passenger demands, *Operations Research* **56**(2), 383–399.

Shih, D. T., Chen, V. C. P. and Kim, S. B.: 2006, Convex version of multivariate adaptive regression splines, *Proceedings of the 2006 Industrial Engineering Research Conference*, Orlando, FL, USA.

Storey, J. D. and Tibshirani, R.: 2003, Statistical significance for genomewide studies, *Proc. Natl. Acad. Sci.* **100**, 9440–9445.

Temiyasathit, C., Kim, S. B. a. and Park, S. K.: 2009, Spatial prediction of ozone concentration profiles, *Computational Statistics & Data Analysis* **53**, 3892–3906.

Tsai, J. C. C. and Chen, V. C. P.: 2005, Flexible and robust implementations of multivariate adaptive regression splines within a wastewater treatment stochastic dynamic program, *Quality and Reliability Engineering International* **21**, 689–699.

Tsai, J. C. C., Chen, V. C. P., Beck, M. B. and Chen, J.: 2004, Stochastic dynamic programming formulation for a wastewater treatment decision-making framework, *Annals of Operations Research, Special Issue on Applied Optimization Under Uncertainty* **132**, 207–221.

Wise, B. M., Ricker, N. L., Veltkamp, D. F. and Kowalski, B. R.: 1990, A theoretical basis for the use of principal component models for monitoring multivariate processes, *Process Control and Quality* **1**, 1–41.

Yang, Z., Chen, V. C. P., Chang, M. E., Murphy, T. E. and Tsai, J. C. C.: 2007, Mining and modeling for a metropolitan atlanta ozone pollution decision-making framework, *IIE Transactions, Special Issue on Data Mining* **39**, 607–615.

Yang, Z., Chen, V. C. P., Chang, M. E., Sattler, M. L. and Wen, A.: 2009, A decision-making framework for ozone pollution control, *Operations Research* **57(2)**, 484–498.

Yeung, K. Y. and Ruzzo, W. L.: 2001, Principal component analysis for clustering gene expression data, *Bioinformatics* **17**, 763–774.

Table 1: Variable selection results for the fleet assignment application using $n = 141$.

Variable Selection Method	# of Variables Selected	Reduction Rate
None	1061	N/A
PCA	140	87%
FDR with three groups from regression trees	454	57%
Reverse FDR	326	69%
Correlated version of reverse FDR	256	76%

Table 2: MARS approximation results for the five cases in Table 1.

Variable Selection Method	# of Basis Functions	R^2
None	84	99.459%
PCA	118	99.013%
FDR with three groups from regression trees	78	99.230%
Reverse FDR	78	99.053%
Correlated version of reverse FDR	76	99.052%

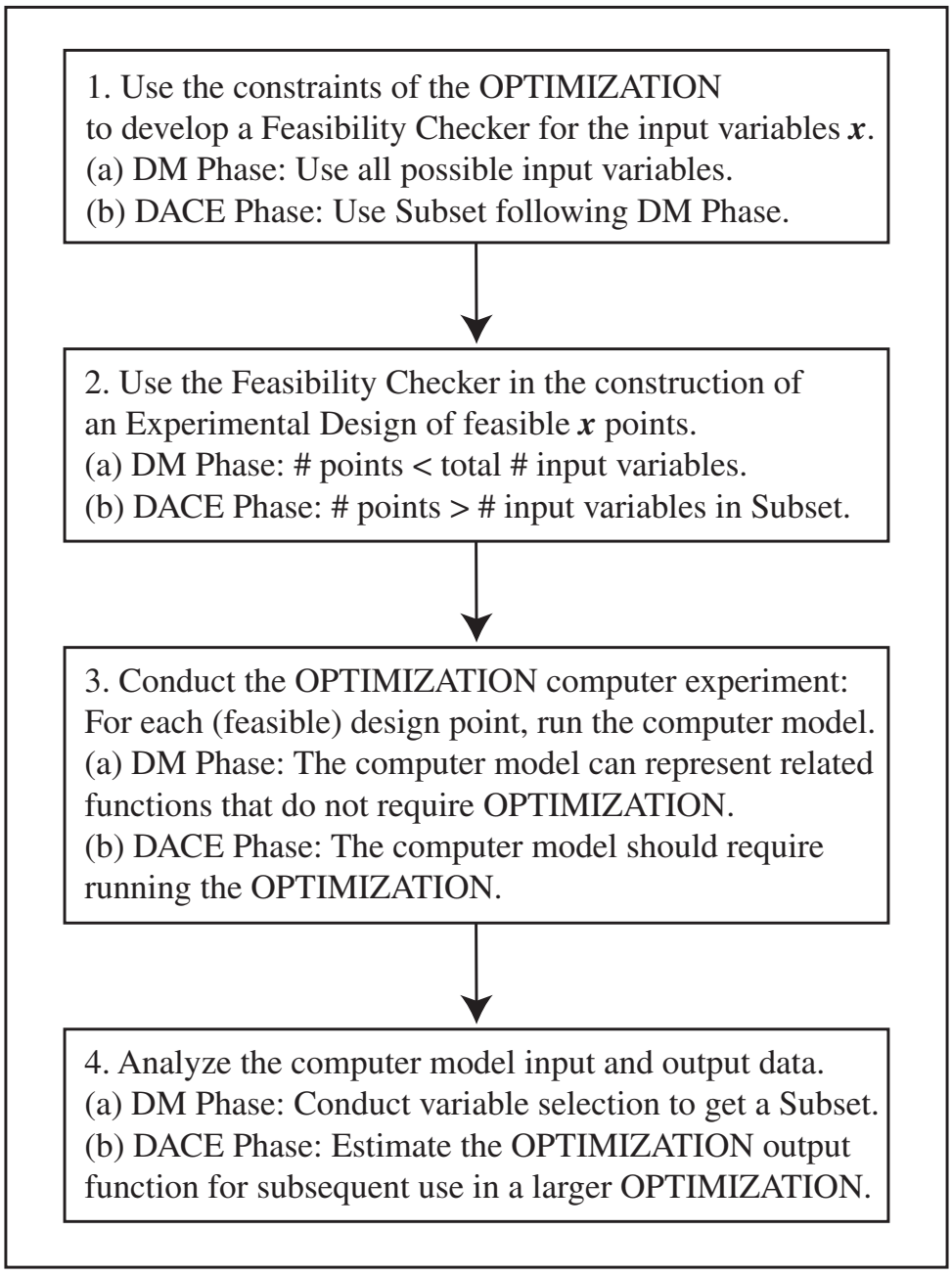

Figure 1: Flow chart for the DM-DACE framework.

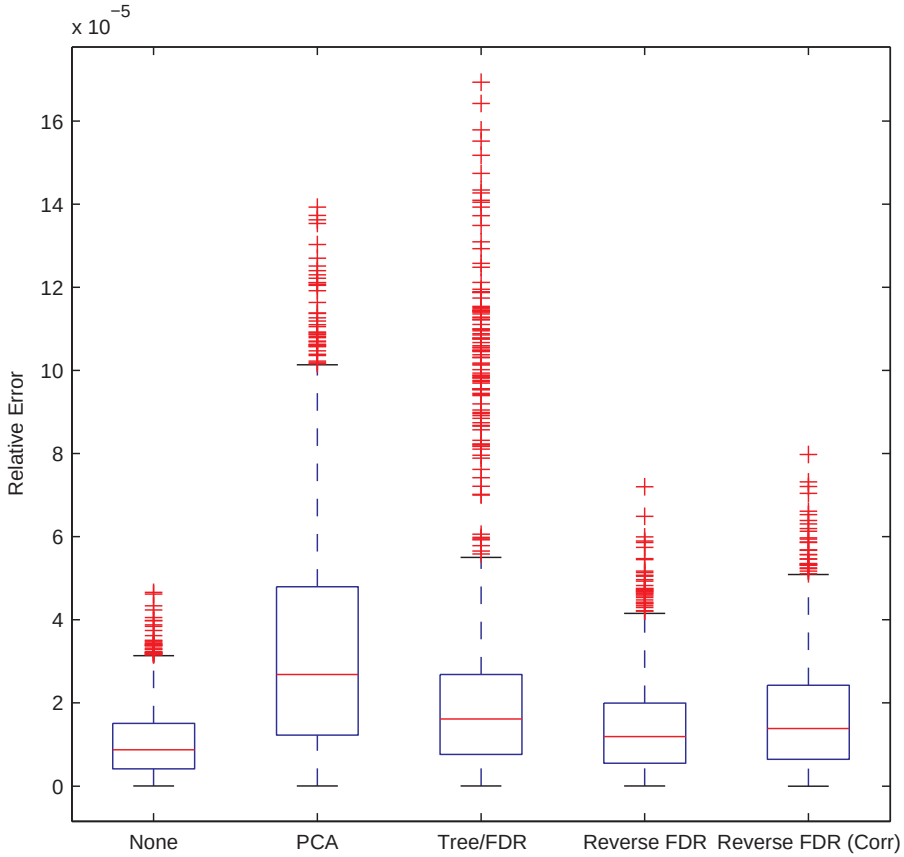

Figure 2: Boxplots of the relative errors from the five variable selection cases for the fleet assignment application.