MANAGING SURROGATE OBJECTIVES TO OPTIMIZE A HELICOPTER ROTOR DESIGN – Further Experiments AIAA MDO 98-4717

Andrew J. Booker and Paul D. Frank *
Mathematics & Engineering Analysis
Boeing Shared Service Group
Applied Research and Technology
Box 3707, M/S 7L-68, Seattle, WA 98124, <booker,frank@redwood.rt.cs.boeing.com>.

J. E. Dennis, Jr. and Douglas W. Moore †
Department of Computational and Applied Mathematics
Center for Research on Parallel Computation
Rice University
P. O. Box 1892, M/S 134, Houston TX 77005, <dennis,dougm@caam.rice.edu>.

David B. Serafini ‡
National Energy Research Scientific Computing Center
E.O. Lawrence Berkeley National Laboratory, MS 50B–2239
1 Cyclotron Road, Berkeley, CA 94720, <dbs@nersc.gov>.

Abstract

It is common engineering practice to use response surface approximations as surrogates for an expensive objective function in engineering design. The rationale is to reduce the number of detailed, costly analyses required during optimization. In earlier work, we developed a rigorous and effective scheme for managing the interplay between the use of surrogates in the optimization and scheduled progress checks with the expensive analysis so that the process converges to a solution of the original design problem. In this paper, we will report our latest numerical tests with a helicopter rotor design problem which has proved to be a fruitful laboratory for experimentation. The results given here support the use of an ANOVA decomposition on a DACE model to identify the most important optimization variables in an optimal design problem.

Key Words: Approximation Models, DACE Models, Surrogate Optimization, Nonlinear Programming, Nonlinear Optimization, Global Convergence, Pattern Search.
AMS(MOS) subject classification: 65K05, 49D37

Introduction

The use of optimization tools with computer simulations to drive engineering design underpins MDO. However, there still are many important problems for which existing methods are either unreliable ad hoc procedures or impractical. Legacy simulation codes that fail on some plausible inputs and run slowly on the rest are far from the clean, infinitely differentiable functions for which optimization specialists design and analyze powerful algorithms. A research collaboration, involving Boeing Applied Research & Technology and Rice University, has developed methods that interpose a computationally clean surrogate function between the optimizer and the simulation, to provide a reasonably accurate model of the true simulation and a function that the optimizer can evaluate quickly. The resulting methods have led to new theories and better practical solutions.

We present numerical results here for two versions of one of the target problems for our collaboration, the design of a lower vibration helicopter rotor blade.

†Professor, Member AIAA & Research Scientist. Research was supported by DOE FG03-95ER25257, NSF CRPC CCR-9120008, AFOSR-F49620-98-1-0267, The Boeing Company, and the REDI Foundation.
‡Boeing Associate Technical Fellows and Members AIAA.
§Staff Mathematician, Member AIAA

The two versions are a 31 variable problem and an 11 variable problem. The smaller problem was obtained from the larger problem by an analysis of variance decomposition, or ANOVA, [9, 12, 13] performed on a kriging interpolatory model of the objective function [15, 6].

The details of the ANOVA decomposition and the way the 11 variables were determined is covered elsewhere by Andrew Booker in this specially organized section. To us, ANOVA identifies the design parameters that have the greatest influence on the quality of the design. We hope to identify a few key variables that account for most of the variation in the objective function, then optimize solely with respect to those variables at reduced expense. In this paper, we present a method and use it to generate numerical results that encourage us in this hope.

In earlier work [4, 3], we have presented and proved convergence for a flexible framework, the Surrogate Management Framework for optimizing problems like this by searching on surrogates rather than the true objective function. We have presented experimental results, based on an implementation called MMF by Serafini. The idea is to use inexpensive surrogate values to find interim designs where the surrogate predicts improvement, and then occasionally to compute the expensive values of the true objective function to test those predictions.

The analysis ensures convergence independent of the approximation concepts [2] used to obtain the surrogates and independent of the accuracy of the surrogates - although more accurate surrogates certainly speed up convergence.

Mathematically, the problem we consider here is:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{B} \equiv \{x \mid a \leq x \leq b\}\ , \end{array} \qquad (1)$$

where $f : \Re^n \to \Re \cup \{\infty\}$, $a, b \in \Re^n$, and $a \leq b$ means that each coordinate satisfies $a_i \leq b_i$.

Problem (1) is an optimization problem with simple bound constraints. Most problems also include other types of constraints, and we are studying ways to handle general constraints. Our helicopter examples include an additional linear inequality constraint. We currently use such constraints either to eliminate one of the variables, or, risking failure in theory, to assign large function values to infeasible points.

Managing nonlinear constraints, especially equality constraints, is an issue that we have not settled yet. We believe that if the surrogate constraints and the true constraints match in values and derivatives, then trust region search methods for the surrogate problems present little analytic difficulty [1]. Although penalty function techniques place the burden of managing nonlinear constraints on the user, this may be necessary since more sophisticated approaches require Lagrange multiplier estimates, which involve derivatives not available for these problems. There are many interesting issues in modeling nonlinear constraints via surrogates, and we will address them in subsequent work.

We are interested in problems with the following properties:

1. The number of decision variables, x, is reasonably small, say $n \leq 100$.

2. It is impractical to accurately approximate the derivatives of f.

3. The routines that evaluate $f(x)$ may fail for some feasible x at the same cost as if a value had been obtained.

4. If x violates any of the bound constraints, then $f(x)$ may not be available.

5. The computation of $f(x)$ is very expensive and the values obtained have few correct digits.

Although the number of *optimization* variables is reasonably small, the *total* number of variables in the problem usually is large. Typically, $f(x)$ is expensive to evaluate because there are large numbers of ancillary or system variables that must be determined for each choice of x before $f(x)$ can be evaluated. For the helicopter rotor examples, x specifies a coupled set of partial differential equations that must be solved in order to obtain dependent system variables that are then used to evaluate $f(x)$. The coupling of PDEs via some iterative method, most often the notoriously unreliable successive substitution approach, explains the third property, since the iteration to resolve the system couplings may run for many iterations and not converge.

Problems with the second and third properties make quasi-Newton methods difficult to apply. Using finite difference gradients in practice depends for success very strongly on finding an effective finite difference step size. The fact that we may not be able to compute the function value at the step size selected compounds this difficulty. Automatic differentiation technology may one day offer actual derivatives in place of finite difference approximations, but the inaccuracy in computing f may still thwart quasi-Newton methods because they are badly affected by function inaccuracies [7].

Our last two properties explain why we allow the value of $f(x)$ to be infinite. In practice, none of our implementations perform calculations with infinity. Formally, we assign $f(x) = \infty$ either when x is infeasible or when the routine to evaluate $f(x)$ does not return a value. This amounts to assuming that any choice of decision variables at which the objective does not evaluate is not optimal. When the optimization method generates an infeasible vector x of decision variables, then we save ourselves the expense of trying to obtain a value for $f(x)$.

This is one place where optimization algorithms traditionally distinguish between types of constraints. It is common practice not to evaluate the objective at points that violate bound constraints, but it is also common practice to evaluate the objective at points that violate, for example, nonlinear equality constraints.

The results here use the MMF software [14], which can be applied regardless of the smoothness of f. Even table lookup components can be involved in the computation of f. For the convergence analysis [14], we need at least to have a continuous function, and the existence of a gradient is even better.

Our starting point for this research is the valuable survey of Barthelemy and Haftka [2], and our idea of replacing direct optimization, when it is impractical, with indirect optimization of surrogates constructed using approximation concepts.

The basic "one-shot" procedure is:

1. Choose a surrogate s for f by some approximation concept. Common approaches to constructing surrogates include:
   - Use simplified physics to obtain a less costly simulation s; or,
   - Evaluate f at selected design sites, x_1, x_2, $\ldots$, $x_d \in \mathcal{B}$ at which $f(x_i)$ is finite for each $i = 1, \ldots, d$, and then obtain s by interpolating or smoothing the function values thus obtained.
2. Minimize s on $\mathcal{B}$ to obtain x_s.
3. Compute $f(x_s)$ and determine if it improves upon the best x found to date, which may be some baseline x or one of the design sites if that approach is used.

The question is what to do if x_s is not good enough to use as a solution to (1)? A sensible modification is to use a sequence of models to identify interesting regions in which to build the expensive models. One aspect of this approach called "variable complexity modeling" has been systematically developed at the Virginia Tech MAD Center [5, 10].

Although the one-shot approach is easily implemented, subject to being able to construct the surrogate, difficulties arise when we try to use it repeatedly as part of an iterative procedure. Previous work [8] describes a rigorous, but complex, attempt to extend the surrogate approach to a practical iteration. Here we use a more general and elegant approach given in Serafini's thesis[14, 4].

Whether surrogates arise from simplified physics models or interpolatory models, they are always modified, or *recalibrated*, to interpolate known values of f. In the first case, the surrogate is the simplified physics model plus a term that interpolates the error in that model where the true objective function value is known. Likewise, when we use interpolatory models, we might build an underlying initial approximation using DACE models fitted to data sites from an experimental design and then calibrate this DACE model using polynomials. This distinction between the underlying initial model and the surrogate used by the framework is important.

Once we have updated the surrogate, we perform a fairly extensive search on the current surrogate problem to decide adaptively when and where to obtain additional values of f to compare to the values predicted by the current surrogate. We use all values that we obtain to improve the current approximation to f, thereby constructing a new surrogate. In the results given here, all such *calibration* points are in surrogate valleys, but we also could include some points where our adaptive strategy estimates that the surrogate is least reliable.

In the next section, we give the bare bones of the Surrogate Management Framework (SMF); more details are given in [4]. In section 3, we report numerical results of experiments on the MMF approach applied to optimal helicopter rotor design for the 11 variable subspace identified by ANOVA. We consider freezing the other 20 variables in two ways: we minimize f on the subspace in the 11 variables that contains the baseline x_0, and on the translation of that subspace to the best 31 variable point we found. One of our runs found the best minimizer we currently know for the helicopter rotor design problem.

Our results make a case that the 11 variable subspaces identified by the ANOVA decomposition provide a better solution than one is likely to get using all 31 variables. As in [4], the number of evaluations of the true objective needed to get a good design is gratifyingly small. On the other hand, it is clear that much more work is needed to realize the potential additional benefits of combining ANOVA with SMF.

The Surrogate Management Framework

The Surrogate Management Framework (SMF) is a set of strategies for using surrogates in both the **Search** and **Poll** steps of the generalized pattern search algorithm for bound constraints studied in [11]. For greater clarity, we have identified a separate **Evaluate/Calibrate** step. In what follows, we assume that we have a family of underlying approximating functions, an initial surrogate, and an algorithm to recalibrate the surrogate.

Let M_0 denote a rectangular mesh along the coordinate direction on $\mathcal{B} \equiv \{x \mid a \leq x \leq b\}$, and suppose that $x_0 \in M_0$ has been given. We use the notation $M_0/2$ to mean the refinement of M_0 that inserts a new point between each pair of points in M_0 adjacent along the coordinate directions.

SMF: Given s_0, an initial surrogate of f on $\mathcal{B}$, and $x_0 \in M_0$, let $X_0 \subset M_0$ contain x_0 and the $2n$ points adjacent to x_0 for which the differences between those points and x_0 are multiples of the coordinate vectors for $\Re^n$. As the algorithm generates $x_k \in M_k$, let $X_k \subset M_k$ be defined in the same way. For $k = 0, 1, \ldots$, do:

1. **Search**: Use any method to choose a trial set $T_k \subset M_k$. If $T_k \neq \emptyset$ is chosen, then it is required to contain at least one point at which $f(x)$ is not known. If $T_k = \emptyset$, then go to **Poll**.

2. **Evaluate/Calibrate**: Evaluate f on elements in T_k until either it is found that x_k minimizes f on T_k or until $x_{k+1} \in T_k$ is identified for which $f(x_{k+1}) < f(x_k)$. If such an x_{k+1} is found, then declare the **Search** successful. Recalibrate s_k with the new values of f computed at points in T_k.

3. If **Search** was successful, then set $s_{k+1} = s_k$, $M_{k+1} = M_k$, and increment k;
   else return to **Search** with the recalibrated s_k, but without incrementing k.

4. **Poll**:
   If x_k minimizes $f(x)$ for $x \in X_k$, then declare the **Poll** unsuccessful, set $x_{k+1} = x_k$, and set $M_{k+1} = M_k/2$;
   else declare the **Poll** successful, set x_{k+1} to a point in X_k at which $f(x_{k+1}) < f(x_k)$, and set $M_{k+1} = M_k$.
   Recalibrate s_k with the new values of f computed at points in X_k. Set $s_{k+1} = s_k$.
   Increment k.

Our framework exploits the convergence analysis for general pattern search methods given by Lewis and Torczon [11] in a novel way that allows great flexibility in the heuristics one can employ. Serafini presents the details of the convergence analysis in this thesis [14], but for completeness, we state the theorem here.

Theorem If f is continuously differentiable on the feasible region $\mathcal{B}$, then some limit point of the sequence $\{x_k\}$ produced by SMF for bound-constrained minimization is a stationary point for problem (1).

Experimental Results

In this section, we present some results on optimizing $f(x)$ in the 11 variables that are the most important, according to the ANOVA that Andrew Booker presents in this section. The results can be summarized as follows:

- Minimizing in the 11 variables identified by ANOVA leads to quick, efficient reduction of the objective for several different DACE models as initial surrogate.

- Setting the values of the 20 variables *not* identified by ANOVA to match corresponding values from the best known solution to the 31 variable problem makes the restriction to 11 variables particularly effective.

- Starting the 11 variable problem from a solution to the 31 variable problem offers suprising improvement, because resetting the surrogate makes search work better.

  And last but not least:

- It is better to be lucky than good.

The graphs below show objective function decline as a function of the number of function evaluations, for several different numerical experiments. In the first, the plot labeled "Good model" shows rapid initial decrease for ten evaluations or so, and then flattens out; this profile is typical. The "Good model" is one constructed according to DACE modelling techniques from a collection of 40 data sites. The plot "Better model" is based on an improvement of the initial DACE model, as Andrew Booker identified an outlier among the initial trials and removed it to achieve a smoother model in 39 data sites. As the chart shows, the "Better model" produces a better result.

The second graph shows, in part, the effect of starting points on the success of the search. The plots labeled "bad start" and "good start" differ only in that the latter starts with the design suggested by a full 31 variable run of MMF, where the former starts with the common starting point for all these calculations. It is noteworthy that, starting from a 31 variable optimum, the 11 variable method can still find improvement. We believe that this happens because the surrogates, after MMF compels them to interpolate dozens of points that are not chosen to improve the condition of the surrogate, become ill-conditioned and unreliable. The fresh start offered in the "good start" solution allows progress where none was apparent in the 31 variable problem.

The last of our conclusions, whose universal truth is undisputed, is supported by the third plot, "Lucky". That plot, which shows the progress of the best solution we found, and the best solution we have seen reported, was the result of the sort of mistake that is easily made in running a set of experiments like these. We built a model on the 11 variable subspace containing the best 31-variable solution. However, in the MMF run, we inadvertently restricted the objective function to the 11 variable subspace containing the 31-variable baseline point.

The run is interesting however, and we view it as showing how robust MMF is with respect to the fidelity of the model. Notice that in the graph, this run initially improves the objective more slowly than any of the other runs – all of which use consistent models. The lucky part is that this seems to have lead us into a fertile part of design space, by which time the model has been recalibrated numerous times using truth values computed in the correct subspace. The effect of the initial erroneous model is damped out as the iterations proceed.

References

[1] Natalia Alexandrov, J. E. Dennis Jr., Robert Michael Lewis, and Virginia Torczon. A trust region framework for managing the use of approximation models in optimization. To appear in *Structural Optimization.*

[2] Jean-Francois M. Barthelemy and Raphael T. Haftka. Approximation concepts for optimum structural design – a review. *Structural Optimization*, 5:129–144, 1993.

[3] A. J. Booker, J. E. Dennis, Jr., Paul D. Frank, David B. Serafini, and Virginia Torczon. Optimization using surrogate objectives on a helicopter test example. Submitted for publication, 1997.

[4] A. J. Booker, J. E. Dennis, Jr., Paul D. Frank, David B. Serafini, Virginia Torczon, and Michael Trosset. A rigorous framework for optimization of expensive functions by surrogates. In preparation, 1998.

[5] S. L. Burgee, A. A. Giunta, V. Balabanov, B. Grossman, W. H. Mason, R. Narducci, R. T. Haftka, and L. T. Watson. A coarse grained parallel variable–complexity multidisciplinary optimization program. Technical Report 95–20, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, October 1995.

[6] Cressie, N. The origins of kriging. *Mathematical Geology*, 22(2):239–252, 1990.

[7] J. E. Jr. Dennis and H. F. Walker. Inaccuracy in quasi-Newton methods: Local improvement theorems. *Mathematical Programming Study*, 22:70–85, 1984.

[8] J. E. Dennis, Jr. and Virginia Torczon. Managing approximation models in optimization. In Natalia Alexandrov and M.Y. Hussaini, editors, *Multidisciplinary Design Optimization: State-of-the-Art*, pages 330–347. SIAM, Philadelphia, 1997. Also available as Rice U. CAAM Tech. Report 95-19.

[9] B. Efron and C. Stein. The jacknife estimate of variance. *The Annals of Statistics*, 9(3):586–596, 1981.

[10] A. A. Giunta, V. Balabanov, D. Haim, B. Grossman, W. H. Mason, L. T. Watson, and R. T. Haftka. Wing design for a high-speed civil transport using a design of experiments methodology. In *Proceedings of the Sixth AIAA/NASA/USAF Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, 1996.* American Institute of Aeronautics and Astronautics, 1996. AIAA Paper 96-4001.

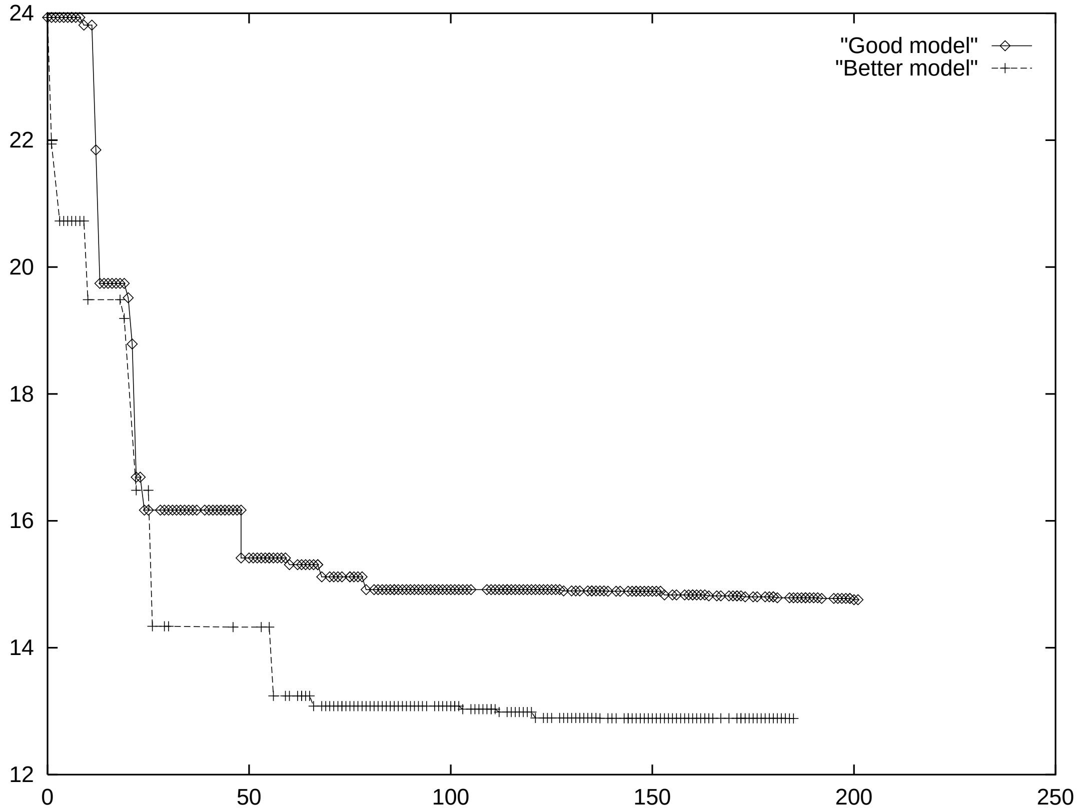
[11] Robert Michael Lewis and Virginia Torczon. Pattern search algorithms for bound constrained minimization. Technical Report 96–20, ICASE, NASA Langley Research Center, Hampton, VA 23681–0001, March 1996. To appear in *SIAM Journal on Optimization.*

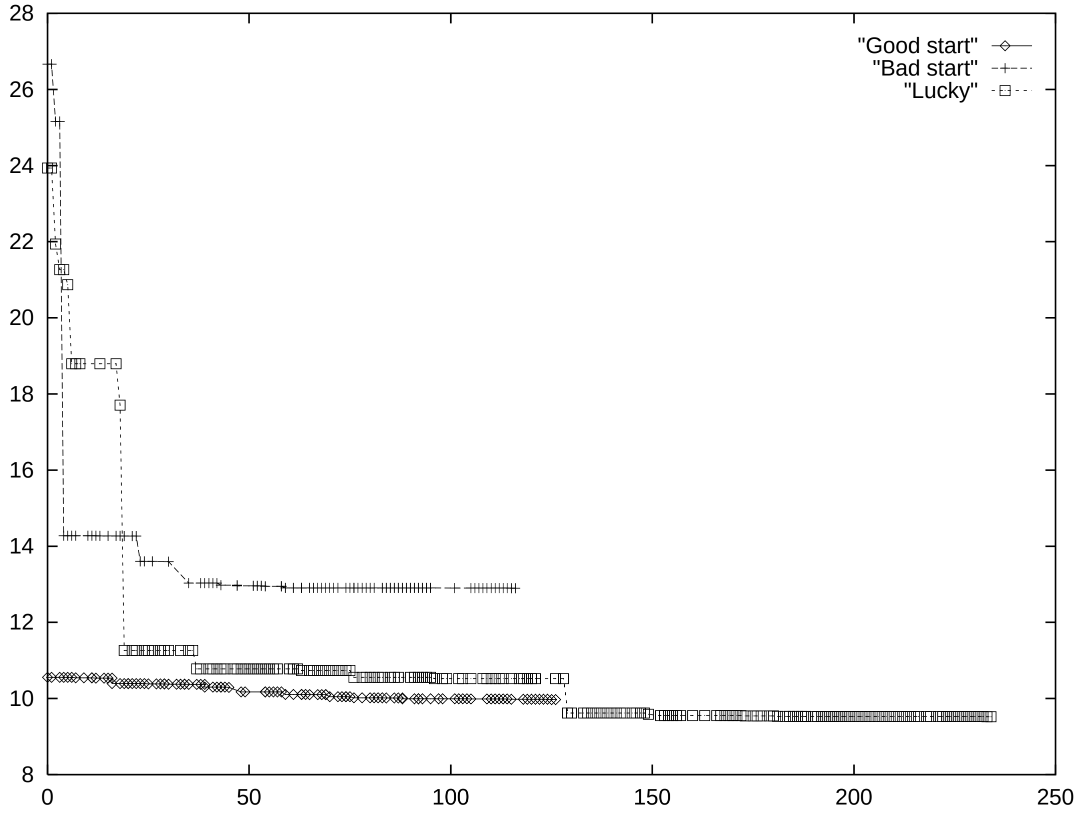
[12] A. B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452, 1992.

[13] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[14] David B. Serafini. *A Framework for Managing Models in Optimization for Computationally Expensive Functions*. PhD thesis, Rice University, Houston, TX, 1997.

[15] G. S. Watson. Smoothing and interpolation by kriging and with splines. *Mathematical Geology*, 16:601–615, 1984.


"Good model"
"Better model"
24
22
20
18
16
14
12
0
50
100
150
200
250


"Good start"
"Bad start"
"Lucky"
28
26
24
22
20
18
16
14
12
10
8
0
50
100
150
200
250