Optimization using Surrogates for Engineering Design ©

Mark Abramson [*] Charles Audet [†] J.E. Dennis Jr. [‡]

December 29, 2002

Preface

The goal of these lectures is to acquaint the audience with some approaches to a class of nasty optimization problems involving nonconvex nonlinear extended-valued functions. Such functions arise often in multidisciplinary optimization (MDO). The first three lectures are meant to set the context for applying our algorithms. The context determines the form of the algorithms, and to present this context requires a bit more than just a short list of assumptions. Briefly though, the objective function and constraints depend not only on the optimization variables, but also on some ancillary variables such as the solutions of some coupled systems by stand-alone solvers for partial differential equations, table look-ups, and other nonsmooth simulation codes. This has important algorithmic implications. First, the function and constraint values may be very expensive. Second, the functions may be nondifferentiable and discontinuous. In fact, they are often treated as extended valued since a function call may not return a value even if all the specified constraints are satisfied.

The approach we treat in these lectures has been successful for some real problems in engineering design. We hope to convince engineers and mathematicians alike that not only are the algorithms given here useful, but the mathematics involved is interesting and relevant. We hope to convince mathematicians that good applied problems produce good mathematics, and that contrary to what they may have heard, they will suffer no loss of virtue as a direct result of considering them.

[*] Computational and Applied Mathematics Department, Rice University - MS 134, 6100 South Main Street, Houston, Texas, 77005-1892 (abramson@caam.rice.edu)

[†] Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal (Québec), H3C 3A7 Canada (charles.audet@gerard.ca)

[‡] Computational and Applied Mathematics Department, Rice University - MS 134, 6100 South Main Street, Houston, Texas, 77005-1892 (dennis@caam.rice.edu)

The first lecture sets the context for the engineering applications, and then we give detail on the mathematical infrastructure for the application framework. In Lecture 4, we give more detail about applying the methods. For the most part, the material is taken from the published papers referred to in place. We owe much to our collaborators for these obvious contributions, but even more, we acknowledge the motivation our Boeing collaborators provided.

In order to aid the reader, we combine the notes for Lectures 1&4 with the bibliography and we use hyperref. This makes the references and the equation numbers consistent and allows Acrobat to facilitate navigation for the online reader.

Lecture 1

1 Properties of the Target Class of Problems

- **General statement of the problem**

$$\min_{x \in X \subset \Re^n} f(x) \text{ s.t. } C(x) \leq 0 \,, \tag{1}$$

where $f : X \subset \Re^n \to \Re \cup \{\infty\}$ and $C : X \subset \Re^n \to (\Re \cup \{\infty\})^m$ are functions with $C = (c_1, \ldots, c_m)^T$, and $X = \{x \in \Re^n : a \leq Ax \leq b\}$ for $a, b \in \Re^n \cup \{\pm\infty\}$ with $a < b$, and A a matrix of rational numbers. The total feasible region is denoted by Ω.

- **Open and closed constraints**
  - A constraint is *closed* at a point if it is satisfied at that point. Some constraints are specified to be closed at every iteration, and they are called *closed constraints*; $x_k \in X$ is a typical example.
  - *Open* constraints at a point are constraints that are not satisfied at the point. Constraints not required to be satisfied until the solution is reached are called *open constraints*; general constraints $C(x) \leq 0$ are usually open.

- **Continuous, discrete, and categorical variables**
  - Continuous variables must always take values from $\Re^n$.
  - Discrete variables are subject to an open constraint that they belong to some discrete subset of $\Re^n$. This enables the use of branch-and-bound algorithms that proceed through allowing continuous relaxations of some of the discrete variables while holding others to specific discrete values.
  - Categorical variables are subject to a closed constraint that they belong to some discrete subset. Thus, there is often no point in representing categorical variables in terms of real numbers. In practice, a component of x is categorical if it must belong to the feasible discrete set or else the problem functions cannot be evaluated.

- **User desideratum**
  - Ease of use, ”global” solutions, robust performance
    * A global optimizer is a feasible point at which the objective function value is as low as it ever is at any other feasible point. A local optimizer is a feasible point at which the objective is as low as it ever is for all feasible points near to the point in question. The operative word is “near” since this implies a topology on the domain, or at least on the set of feasible points. The definition of a topology for a

problem with mixed categorical and continuous variables is crucial since this gives meaning to the points the algorithm can be expected to find.

The issue of finding global solutions is an interesting one. A fundamental reference is [75] where it is shown that it is impossible for an implementable algorithm to be guarenteed to solve all general nonconvex global optimization problems. Of course, this does not mean that we may not find a global optimizer in a particular case of interest.

The study of global optimization algorithms is useful in part because it is the nearest model problem to what the user really wants - the "right" local optimizer. By designing to achieve this ideal outcome, we may increase chances that we will succeed for a particular problem, and at least we offer the user assurance that we tried to find the best local optimizer we could.

Sometimes it is confusing to beginners in the subject that many algorithms can be shown to converge globally (i.e., from any starting point) to a local optimizer. Thus, to say an algorithm is *globally convergent* is not to say that it always finds a global optimizer.

* Users often put efficiency down the list of priorities when compared to reliability in finding some better point and trying to find a global optimizer. This is true even for many expensive problems. Of course, once it is demonstrated that an algorithm will find a solution reliably, then the user predictably will begin to press for efficiency.

– Optimization under uncertainty: This is the latest new thing in optimization. It is meant to ensure that the optimum is robust with respect to uncertainty in f, C and x. The end result is an optimization problem with perturbed f, C. It is pretty easy to think of sources of uncertainty, the difficult thing is to quantify them.

A way to think about uncertainty in x is

$$\min_{x \in X \subset \Re^n} F_{\mathcal{R}}(x) \text{ s.t. } C(x) \leq 0 \text{ , where } \int_{\mathcal{R}(x)} d\mu = 1 \text{ and } F_{\mathcal{R}}(x) \doteq \int_{\mathcal{R}(x)} f(y) d\mu. \quad (2)$$

Here, we think of $\mathcal{R}(x)$ as the subset containing x together with some probability distribution pertaining to the member of $\mathcal{R}(x)$ that might actually result if we specify that x is to be used. Then we optimize the "average" $f(x)$ over over $\mathcal{R}(x)$ to get a robust estimate of the likely value of the objective function in the actual design if x is specified. Take the case of a structural member whose $diameter$ is a design variable (actually, there are good reasons why the reciprocal of the diameter is often used!). In this case, if I optimize the design and order a load of structural members all of a certain diameter, I am unlikely to get a grossly different diameter, but I am likely to have no two of exactly the same diameter. The idea of robust design is to make decisions that won't lead to less than expected performance because of this. Clearly there are important issues related to the extent to which we are confident of the relevance of f as an objective function, but that is a separate issue to some extent.

But now consider that the structure is being designed to have optimal performance in some context, like climatic conditions, loads, etc. Notice that these conditions might be treated as certain but varying, or they may be treated as uncertain.

For example, an airplane might be designed with performance at its cruise altitude and speed as the major part of the objective function, but either in the objective function or constraints or both, there will also be some consideration of performance during takeoff and landing. These considerations could be modelled as deterministic. Uncertainty comes into play when one tries to robustify against not flying at exactly the specified altitude and speed during a stage of the flight, or any of the other myriad things the reader can easily conjure.

– Statement of multiobjective problem and Pareto surface: Virtually all real optimization problems have more than one objective and they conflict with each other. Call them $f_1, f_2, \cdots f_p$. A standard approach is to set goals for all but one objective and to write goal satisfaction as a constraint. The set of vectors of function values $(f_1(x), \cdots f_p(x))$ at the solutions for the various choices of goals as the goals vary is called the Pareto surface or efficient frontier. Thus, a point on the Pareto surface has the property that to improve any function value with a feasible move, at least one other function has to degrade. A standard procedure is to construct part of the Pareto surface in lower dimensions by studying the ”trade off” between a pair of competing objectives. For example how much more would the wing weigh if we made it 1% more stiff? Multiobjective optimization is one of the really important areas of optimization for applications. There is a nice relationship between multiobjective and robust optimization. One can view uncertainty as an objective along with the usual performance objective, but my impression is that it is more often treated via constraints.

2 Multidisciplinary Design Optimization (MDO)

- **Multidisciplinary Analysis (MDA)** - Why is it hard?

  – Formulate as a system of nonlinear equations
  The idea is to build a solver for coupled systems of disciplines by linking single discipline ”blackbox” legacy solvers together instead of recoding a single tightly coupled solver.
  Given a decision vector $x \in \Re^{n_d}$, find $u(x) \in \Re^{n_s}$ by solving $F(x, u) = 0$, where $F(x, \cdot) : \Re^{n_s} \rightarrow \Re^{n_s}$, but $F(x, u)$ is computed in blocks of rows. In other words, you may not be able to compute $F_1(x, u)$ without computing $F_2(x, u), \ldots F_{n_1}(x, u)$ as well. For simplicity, we let $F_i(x, u)$ denote the entire block of rows of $F(x, u)$ that are computed all together. Figure 1 shows the case of aeroelastic analysis where $u = (M_1, M_2)^T$, and x represents the nominal wing geometry. Think of x, which is suppressed in the figure, as parameterizing each of the single discipline boxes, which are indicated by the dashed lines. There certainly are other parameters that characterize

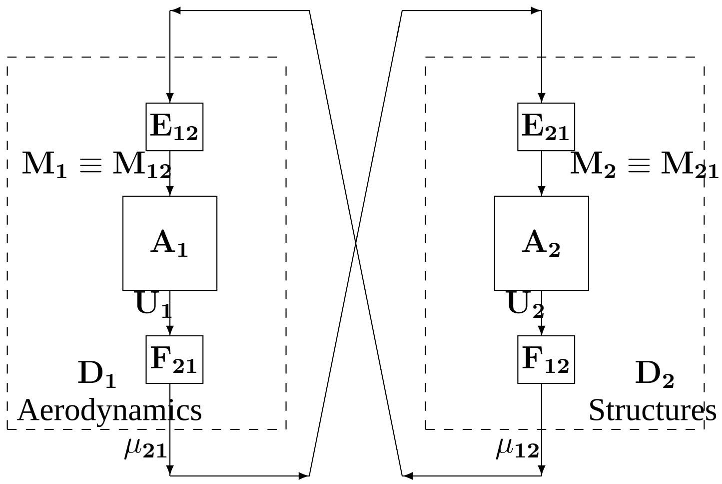


Figure 1: Aeroelastic System

the so-called "design point", like the viscosity of the air, angle of attack of the wing, and speed. We have completed the MDA for a given decision vector x when we have found $u(x)$ such that the following hold for $u = (M_1, M_2)$.

$$F_1(x, M_1, M_2) = E_{21} \circ F_{21} \circ A_1(M_1) - M_2, \tag{3}$$
$$F_2(x, M_1, M_2) = E_{12} \circ F_{12} \circ A_2(M_2) - M_1. \tag{4}$$

and A_1 is the the flow solver (maybe finite difference) and A_2 denotes the structures code (maybe finite element). The E, F routines are interface routines that do such bookkeeping as converting pressures to loads and deflections to shape. Figure 2 provides a more detailed idea of what the E and F boxes look like in that there is probably one required for each of the other disciplines connected to A_i. These interface boxes are the source of many of the difficulties with MDO.

– Why is successive replacements is used in MDA and not Newton? When faced with a system of nonlinear equations to solve, we automatically think of Newton's method. The problem is the availability of derivatives of F with respect to u. Each finite difference step would require a single discipline solve, and the number of correct digits in the difference might be small anyway. Still, one suspects that sparsity is there to be exploited, and the number of variables involved may be too large to use Broyden-type updating methods. The most common approach is for the user to apply intuition to a sequencing of the single discipline solvers and use successive replacements, i.e., fix $u_2, \ldots u_{n_s}$, solve the first equation block for a new u_1. Now fix all but u_2 with the new u_1, and use F_2 to get a new u_2, etc, and iterate. If convergence is a problem, then reorder the equation and unknowns and try again.

- **Formulations of MDO** using aeroelastic design example

  – MDF or black box or closed equations or control theory method (different fields call it different names. Different engineering disciplines are often different worlds.): This is

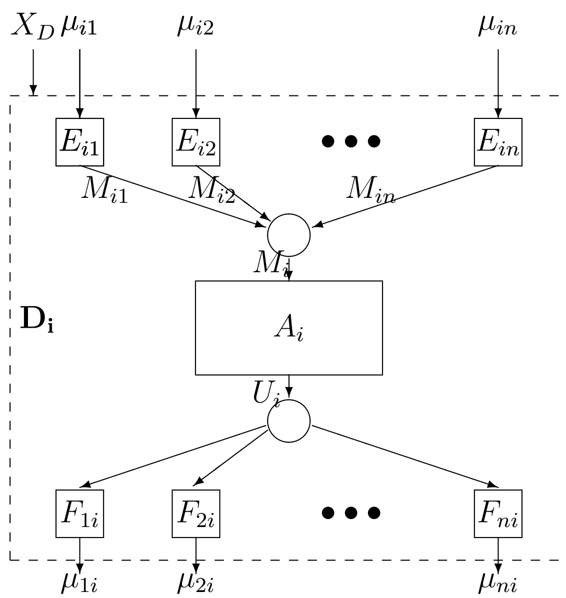


Figure 2: Generic Discipline

the traditional approach in which a complete MDA for $u(x)$ is done for each choice of the design variables x. Thus, the optimization problem to be solved is a function only of the design variables x, and it looks like (1). An important point is that the objective function $f(x)$ is almost surely of the form $f(x) = \bar{f}(x, u(x))$. The point being that the state or system variables u are needed to evaluate the design objective function. Thus, given x, we solve an MDA problem for u, and then we have the arguments to evaluate $\bar{f}, \bar{C}$. Notice that if there are bounds on u, then they become bounds on $u(x)$, and hence they are really nonlinear constraints. If bounds on $u(x)$ must be closed, (for example bounds on u may represent safety considerations - like operating temperatures of some chemical process), then we may have a nasty difficulty for the optimization algorithm.

- Advantage is optimization variables are the design variables (generally $n_d << n_s$ and every iterate can be made feasible. Thus, when we stop, we only need to restore feasibility in $C(x)$, often called the side constraints.
- Disadvantage is computational effort to complete an MDA for every function value. Even finite difference derivatives with respect to x require an MDA for each forward difference quotient.

– All-at-once (AAO) or Simultaneous Analysis and Design (SAND) or open equations or nonlinear programming method: This is the ideal in many ways. It doesn't always work. There is some analytic and practical evidence that nonsmooth behavior in the solution $u(x)$ gets smoothed as it passes through $f(x, u(x))$ and so the blackbox method may work when the AAO method doesn't. However, when the AAO approach does

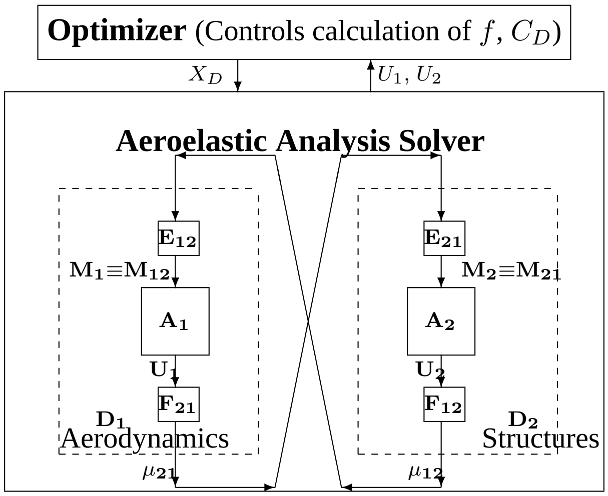


Figure 3: Multidisciplinary Feasible (MDF) Method

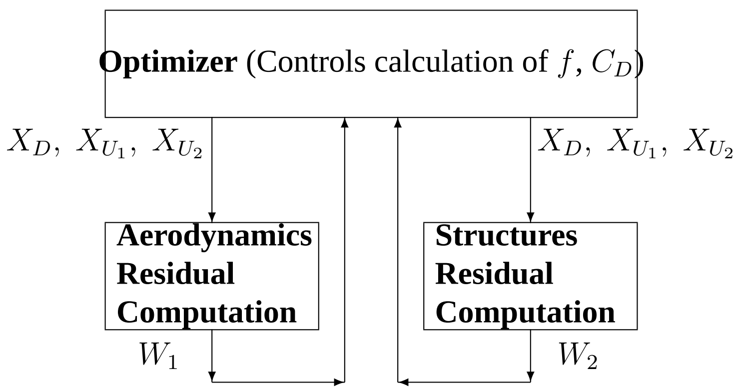


Figure 4: All-At-Once or AAO Formulation

work, it seems most efficient. One solves the AAO problem as:

$$\min_{x,u} f(x,u) \ \text{ s.t. } \ x \in X,\ F(x,u) = 0,\ \text{and}\ C(x) \leq 0\ . \tag{5}$$

Notice that this requires that we “open” the single discipline solvers in order to be able to evaluate their internal residuals (remember, U is produced by solving an internal system of equations). This is expensive, and generally it is resisted by the single discipline specialists.

* Advantage is less expensive function values and derivatives as well as a generally less nonlinear optimization problem
* Disadvantage is many more variables (the design variables + the ancillary variables u, as well as the inability to use legacy disciplinary solvers. Feasible point algorithms are not compatible

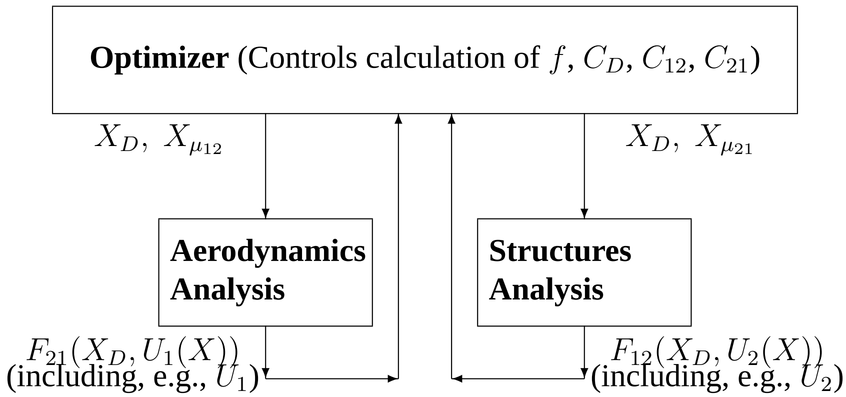


Figure 5: Individual Discipline Feasible or IDF Formulation

– Individual Discipline Feasible (IDF) or In-Between methods: Generally, a small percentage of the state variables from one discipline's solver are needed as input to other solvers. Suppose we reflect this by writing $u = (z, u)$, and z is the smaller subvector of the solver variables u actually needed as input to various solvers. We can now pose the MDO problem as:

$$\min_{x,z} \bar{f}(x, z, u(z, x)) \text{ s.t. } x \in X, \;\; C(x) \leq 0, \;\; (z, u(z, x)) - u(x) = 0. \tag{6}$$

  * Advantage is ability to use legacy disciplinary solvers as they are and the optimization problem is nearer the size of the MDF than the AAO problem. Seems to be more efficient than the blackbox.
  * Disadvantage is that the derivatives needed are like those needed for MDA, and feasible point algorithms are not compatible

Alexandrov and Lewis seem to have rediscovered this formulation, although there may be some difference we missed, and they do not compare it to IDF. They call it DAO or "distributed analysis optimization", a nicer name than IDF [3]. Alexandrov and Kodiyalam give computational evidence that this is an effective formulation [2].

Lecture 4

4 Surrogates for Expensive Functions

The earlier lectures set up the mathematical underpinnings of the surrogate management framework (SMF) for solving the design optimization problem (1). Our purpose here is to give an overview of how surrogates can be constructed for the problems we have seen in the previous two Lectures and to provide a test case attesting to the value of the SMF.

Many problems in engineering design are posed in the MDF, or multidisciplinary feasible, MDO format. The function values require calls to expensive solver codes often formed by linking single discipline solvers. The linking of these solvers may be unsuccessful and expensive anyway, and in general not very many digits of the results are correct. But, engineers are resourceful people, and they do not have the luxury of throwing up their hands just because a problem is intractable. A common approach they take is to build relatively inexpensive approximations to use as optimization surrogates for the expensive functions. A convenient straw man for this approach is the following:

The straw man surrogate (SMS) approach:

1. Choose surrogates s_f and S_C based on either:
   - less expensive solvers based on simplified physical models; or
   - approximations obtained by evaluating f, C at selected design sites, $x_1, \ldots, x_d$, and interpolating or smoothing the function values.
2. Solve the surrogate optimization problem to obtain x_s.

$$\begin{array}{rc} \min & s_f(x) \\ \text{s.t.} & S_C(x) \leq 0 \end{array} \tag{7}$$

   Every user has their favorite approach for this part. Good choices are some sort of randomized search or standard nonlinear programming algorithms. Several point might be returned as putatively better designs. It is pointless to solve (7) to much accuracy except in the unlikely event that we know the surrogates to be accurate approximations, as for example when quasi-Newton approximations are used as surrogates. Since we are more interested in using (7) as a way to identify basins of local optimality for the real design problem, we will global search techniques on the inexpensive surrogate problem.
3. Compute $f(x_s), C(x_s)$ on at least one of the putative new designs to determine if any is an improvement over the previous best x.

What if no improvement was found? In the traditional quasi-Newton approach where the surrogates match at least the gradient of the problem functions at the iterate, improvement can be guaranteed by a line search or trust region variant of (7) [33]. Basically, this is the result of using a surrogate that is guaranteed to be first order accurate in the distance from the current iterate. Such methods are not our subject.

We are concerned with the case where the SMS did not find a better design, the accuracy of the surrogate is suspect at best, and it can not be rigorously controlled by restricting the move in design space. There are several sensible approaches in the literature, e.g., [16, 47]. Another interesting approach is [81], which is intended for the case when one is to do the best one can on some fixed budget of function evaluations.

Our main topic thoughout is a rigorous framework for the SMS, which we call the surrogate management framework (SMF). Therefore, so early in our presentation of the material, we can not give a detailed definition. Still, for the purposes of understanding the case study of this Lecture, we must give the reader some correct, if vague, notion of how the SMF fits with the SMS. To this end, we say that the SMS approach above functions as a speculative search procedure on a mesh of certain resolution imposed on design space. We continue to iterate the SMS as long as it finds an improved design, and we use the incident function values to recalibrate the surrogates at each iteration. When SMS fails, we fall back on a rigidly defined, but possibly expensive, local polling step to determine if we may refine the mesh before we recalibrate the surrogates using any function values computed in the process and reinvoke the SMS on the finer mesh.

The remainder of this Lecture will be to give a brief and incomplete sample of ways of building and using surrogates in optimization.

4.1 Building surrogates

In this section, we will look at some of the effective ways of constructing approximations to either all or part of the objective and constraints and then use those approximations to build optimization surrogates. For example, in space mapping, one optimizes a surrogate that is a composition of two functions.

A difficult MDO issue is how to construct surrogates for the linked problem from surrogates for the individual disciplines. The obvious approach, to carry out an MDA using the individual discipline surrogates, raises many questions. For example, how does one match a set of individual discipline approximate solvers of varying fidelity to obtain a surrogate? Surely some disciplines have more influence than others on the problem, but is it sensible to expend more effort on those approximations when they will be linked with what are clearly poor approximations to other disciplinary solvers? We will not treat this issue because despite its importance, it is not central to our goals. More work is needed on such questions. This entire field is in its early days.

It will be helpful to use some terminology coined by the space mapping community. If there are two models of a process, one very accurate but too expensive to use directly with an optimization routine, and one much less expensive and less faithful, then we will call them the *fine* and *coarse* models respectively. Thus, a coarse model $\hat{f}$ might act as a surrogate, but it may be a step in building a surrogate as we will see below.

Another piece of terminology we will use is *surface* to denote a function (it may be vector valued) trained to fit or to smooth fine model data. Thus we lump kriging, response surfaces, polynomial interpolants, neural nets, etc, all as surfaces. That we do so is a function of the birdseye view we aim for; it is not a technical judgement. Neural nets have the nice property of being set up to interpolate to multidimensional input and output data. MUST CHECK THE LITERATURE FOR INCREMENTAL TRAINING AS NEW TRAINING PAIRS ARE ADDED

Since it is helpful to discuss surfaces fitting approaches in context, we will begin with a brief catalog of data site selection techniques.

4.1.1 Choosing data sites for surfaces

The first issue in building a surface is how to choose data sites on which to base the surface fit. Here are some ways in use:

- by statistical approaches based on the underlying functional form and the domain of interest. To this end, one might assume with dubious justification that the underlying form to be fit is a quadratic. One then asks where the data sites should be chosen in some rectangle containing the domain of interest in order to reproduce the quadratic. The assumptions on the underlying form and on the rectangular region are a stiff price to pay for making the results of the experimental design literature available for choosing the data sites. The quadratic assumption used to obtain the data sites need not be taken seriously if the resulting sites are well scattered, but the rectangular region is more problematic. The objective and open constraints may not be defined throughout such a region, but also our methods will only consider designs that satisfy any simple linear constraints and so it is unfortunate to be asked to do expensive calculations to little purpose. Thus, it would be ideal to have schemes that would adapt to a given polygonal region with external sites near the boundary. More work is needed on such questions.

- by judiciously scattering points to fill space. One might eschew any assumptions about the functional form one is trying to fit, and instead choose the data sites to be well scattered in the domain of interest. Statisticians have again led the way in schemes for choosing space filling sites, and again, there are difficulties when the domain of interest is not a rectangle. More work is needed on such questions. Latin hypercube sampling (LHS) [66, 74], orthogonal arrays ($\mathcal{OA}$) [67] and $\mathcal{OA}$-based LHS [76] are prototypes of this approach, which also is used in quasi-Monte Carlo integration. Some surveys are [70, 10, 61].

- by enforcing a poisedness condition on the geometry of the points, which forces convergence of the surfaces to the underlying function. A notable application of this approach is in the work of Powell [57] and Conn, Schienberg, Toint [22]. One can think of this as the numerical analysts' approach as opposed to the statisticians' approach given above. Fitting surfaces to multidimensional data often leads to a numerically ill-conditioned problem. In one dimension, it is well known that any distinct set of data sites gives a nonsingular coefficient matrix for determining the parameters in a linear fitting problem from a family of basis functions. One can show easily that choosing distinct multidimensional data sites does not guard against an exactly singular coefficient matrix needing to be inverted to solve for the fitting parameters. The poisedness conditions are meant to ensure well-conditioned fitting problems to determine the initial surface and to correct it as new data comes available. This leads to an elegant theory, though the number of points needed to apply that theory is rather larger than one would probably use to solve the problem. But such quibbles aside, the methods are very effective when they are properly implemented.

4.1.2 Surrogate forms

The most straightforward kind of coarse model directly approximates the fine model and acts as an autonomous surrogate, but regardless of how they are combined to form surrogates:

- some polynomial models and response surfaces use experimental design to select data sites.
- kriging, neural nets, least degree polynomials use space filling designs.
- Hermite surfaces use data on the underlying function and its gradient. These surfaces are well suited to trust-region surrogate management techniques [1]. GET SOME OF NATALIA'S NEWER REFERENCES

There are clever ways to design surfaces to correct a coarse model and to be combined with a coarse model to act as a surrogate in optimization (for more detail see [31]):

- surfaces from data on the difference between the fine and coarse models. The surrogate is the coarse model + the surface
- surfaces from data on the quotient of the fine and coarse models. The surrogate is the surface $\times$ the coarse model
- space mapping surfaces from fine model parameters to coarse model parameters. The surrogate is the coarse model applied to the image of the fine model parameters under the space mapping surface (see the next section)

4.1.3 Space mapping

The idea of a space map is very appealing, and it seems to have proved its worth in electrical engineering. John Bandler, an electrical engineer and entrepreneur from McMaster University in Ontario, seems to have originated the idea, and he has collaborators applying it in several variations. John appears to concentrate on the global space mapping approach, while Kaj Madsen and his student Jacob Søndergaard are developing the local space mapping approach. Local space mapping is a quasi-Newton alternative to the GPS based surrogate management framework we discuss here, and the global approach is a kind of SSM. This might tip you off to the simpler conceptual nature of the global procedure, and so it will be the basis of our explanation.

Assume that $\hat{f} : \hat{X} \subset \Re^m \to \Re$ is the coarse model and $f : X \subset \Re^n \to \Re$ is the fine model. The assumption that underlies global space mapping is that there is a mapping of the fine model parameters to the coarse model parameters, $P : X \to \hat{X}$, with the result that $f \approx \hat{f} \circ P$ well on X. One chooses data sites in X, gathers the corresponding data on P, obtains the corresponding surface P_s, and the optimization surrogate is $\hat{f} \circ P_s$.

The terminology is sensible, but it does temp the reader to a misconception that P depends only on $\hat{X}$ and X. It also depends on the particular coarse and fine models, and so $\hat{P}$ might be better notation. This will become more clear later.

IS THE FOLLOWING INTUITIVE EXPLANATION USEFUL OR PATRONIZING AND EXTRANEOUS?

A geographically appropriate motivation for the underlying assumption can be seen from the analogy to rowing a dragonship to a particular port. Take as a coarse model the strategy that has the Viking helmsman head the bow directly towards the desired spot. The coarse model says that the ship will go where the bow is pointed, which neglects wind and tide.

In space mapping, there is a useful distinction made between the full output of the two models, the tracks they predict for the ships, and the objective, which is to minimize the distance between the track of the ship and the desired target. Each model returns the predicted track of the ship, but the optimization problem is to find the course input to steer that will cause the track of the ship to lead to the mouth of the channel. Wind and current are not in $\hat{X}$, which only contains headings to steer.

But, the Danes did not reach the new world by using the coarse model as surrogate. Instead, the helmsman considers not only the desired course, but also the additional input of wind and wave and puts the bow on a compensatory course - he has applied a space mapping from the input vector of desired course, wind, and wave to obtain the single coarse model parameter, steered course, and then he applies the coarse model and steers the resulting course. The fine model can be thought of as the actual physical experiment, i.e., if you steer a certain course with a certain wind and a certain sea, then you will actually trace a (different) course. The objective in this case would be to arrive at the desired port, i.e., to minimize the distance between the track of the ship and the mouth of the channel into port. If the ship goes where it should, then the helmsman has a good space mapping.

At the risk of overdoing this analogy, one could also say that if the helmsman makes a decision based on relevant experience and lashes the tiller before going below, that is like global space mapping. On the other hand if he constantly readjusts his tiller based on his observations of the ship's track, then it is like a locally linear space mapping.

But, the Danes did not reach the new world by rowing. If the ship is under sail, then both models must be different. The coarse model must take wind direction into account to obtain a more complex piecewise linear arc incorporating tacking and wearing to steer rather than a straight line. Now the fine model for sailing must tell us where we will end up by following a given polygonal course in given sea with given wind direction and speed.

Let us abandon the analogy and return to the formalities and the crucial question of how one obtains the data $P(x)$ for a given $x \in X$. Each such value requires an expensive fine model evaluation $f(x)$ as well as the solution of a cheap problem like $\min_{\hat{x} \in \hat{X}} \|\hat{f}(\hat{x}) - f(x)\|$ for the value $P(x) = \hat{x}$. Bandler and his collaborators have used neural nets and polynomials to define P_s. Madsen and Søndergaard also update local linear approximations to improve the coarse model and circumvent the underlying assumption. It is likely that they will succeed in providing a rigorous basis for their approach. One clear difficulty is that the optimization subproblem for obtaining $P(x)$ may have multiple solutions. This is one of the research topics Madsen and Søndergaard are investigating. They have satisfactory regularizations, but they want to do even better. SPRINKLE THIS PARAGRAPH LIBERALLY WITH REFERENCES FROM THE BIT ISSUE.

4.1.4 Orthogonal Arrays and DACE Surfaces

We will use kriging or DACE surrogates of the objective function in the numerical studies that follow. The constraints are simple inequalities and bounds, and no surrogates were necessary.

These approaches may be based on whatever data one has, or they may choose data sites in a couple of ways. In LHS, the domain is an n-dimensional cube and the vectors that determine the data sites have each component chosen from equally spaced values. In $\mathcal{OA}$s, the components are assigned from l distinct values.

An $\mathcal{OA}$ is said to be of strength σ if every subset of σ components has the same number of occurrences of each of the l^σ choices of component values. LHS designs are of strength 1. The design sites at which the objective function f was evaluated to provide data for the initial surrogate was an $\mathcal{OA}$ of *strength 2*. Efficient code for generating $\mathcal{OA}$ designs is available from STATLIB(http://lib.stat.cmu.edu).

$\mathcal{OA}$s were devised for rectangular regions. In the helicopter rotor blade design examples, the mass constraint induces a nonrectangular feasible region. One might try various strategies for adapting $\mathcal{OA}$ designs to this region, e.g.

- Generate a design with d points in the rectangle defined by the variable bounds, then alter the design so that the d points satisfy the mass constraint.
- Generate a design with many points in the rectangle defined by the variable bounds, then discard the points that are outside a slightly expanded mass constraint boundary.

We were able to ignore this restriction in the case study because the constraints were not active during the solution process. This is fortunate because it avoids confounding ways of dealing with oddly shaped regions with the overall problem of using a surrogate.

After the design sites have been chosen and the objective function f has been evaluated there, the initial surrogate s can be constructed. This surrogate is intended to be an approximation of f throughout the region of interest that is inexpensive to evaluate. It is recalibrated as new function values are obtained in the course of solving the optimization problem. We use a flexible set of functions from which surrogates can be selected to avoid overly restrictive assumptions on the structure of f.

The surrogates for these tests came from the family of functions defined by the kriging procedures discussed in the DACE literature. The kriging parameterization is defined by means and covariances of function values, and it may seem more intuitive than other fixed degree polynomial approximations in the response surface literature. For some choices of covariance function, kriging is equivalent to spline interpolation, a correspondence that has been discussed in the geostatistics literature [83].

The surfaces we use can be written

$$s(x) = \sum_{i=1}^{d} w_i(x) f(x_i) \; .$$

Thus, prediction function values at any x are weighted averages of the values of f at the data sites. The weight $w_i(x)$ depends on how far the ith data site is from x.

The correlations needed to define weights were gotten by attempting global optimization of the (log) likelihood function. An interesting point here is that since this parameter identification problem for the surrogate involved only the surrogate form, very accurate objective and derivative values for them are available. That is why were were able to use the multistart Newton algorithm in [68]. Of course, this is a nontrivial computation, but, it does not approach the cost of applying a derivative-free algorithm directly to the actual design problem.

One technical difficulty with using surfaces as surrogates is that one always has to invert a matrix to obtain coefficients. Kriging calculations require inversion of the matrix of estimated correlations between function values at the design sites. The initial matrix usually is well-conditioned if the initial data sites are chosen by any of the methods suggested. However, if we are successful in our optimization procedure, then we will be called upon to recalibrate the surface using data at points that cluster as they converge to an optimizer. Thus the process of recalibration generally causes subsequent matrices to be ill conditioned with respect to inversion. In the case of kriging correlation matrices, Andrew Booker has effective new procedures, but for these results, the problem was dealt with by adding a small number (10^{-6}) to the diagonal of the correlation matrix. With this addition, the approximating functions do not exactly interpolate the observed function values; however, they retain their flexibility and predict observations very closely. GET REFERENCES FROM ANDREW ONCE THEY ARE APPROVED FOR RELEASE.

Finally, Booker has found that performing a functional analysis of variance [39, 67, 70] on the surrogate function s is a useful way of identifying lower-dimensional subspaces in which most of the variation in s resides. This ANOVA technique, which can in principle be applied to any square-integrable function, decomposes s into *main effects* (contributions of individual variables to variation in s) and *interaction effects* (contributions of combinations of variables to variation in s). The hope is that one can identify a few key variables that account for most of the variation in f, then solve 7 on the lower dimensional subspace defined by those variables. The point is that derivative-free search methods generally reduce to some sort of sampling techniques. As the dimension increases, the number of function values needed to make even a moderately dense sample goes up exponentially. Here, not surprisingly, when we restricted the search to the indicated 11 dimensional subspace, we found a better minimizer in fewer function values than in the 31 variable case.

4.2 The surrogate management framework

The SMF is just a reprise of the filter pattern search method with the strawman algorithm playing the role of a SEARCH routine. Thus, the algorithm steps become:

Search step: Use surrogates s_f, S_C to find some promising candidates where f, C will be evaluated.

Poll step: Polling is done on the actual (barrier) functions, which guarantees convergence of the algorithm (through the sequence of unsuccessful poll steps).

Update s_f, S_C with the new evaluations of f, C.

Surrogate Management Framework

Given initial surrogates s_f, s_C and $p_0 \in M_0$, a grid on $\Re^n$,
let $P_0 \subset M_0$ be p_0 and the points of M_0 adjacent to x_0.

For $k = 0, 1, \cdots$, do

1. Search on s_f, s_C to find an unfiltered $x_{k+1} \in M_k$ and then set $M_{k+1} = M_k$ and update the surrogates;
2. Else if p_k is the only unfiltered point in P_k; Then set $x_{k+1} = x_k$ and $M_{k+1} = M_k/2$ and update the surrogates; Else return to 1.

Thus, the algorithmic framework is very simple, and surprisingly effective.

4.3 The Helicopter Rotor Blade Design Problem - A Case Study

In order to show how effective surrogate optimization can be, we consider structural design of helicopter rotor blades to minimize vibration transmittal to the hub.

As indicated in Figure 6, the design variables consist of up to five structural parameters for each span segment. The variations on this problem that we have considered have 11, 31, and 56 variables. As described below the objective function is a weighted sum of various harmonics of forces and moments. The analysis code used is Tech01 [72].

Tech01 is an MDA code. The disciplines include dynamic structures, aerodynamics, wake modeling, and controls. The run time for a Tech01 fixed-wake analysis is roughly 20 minutes on

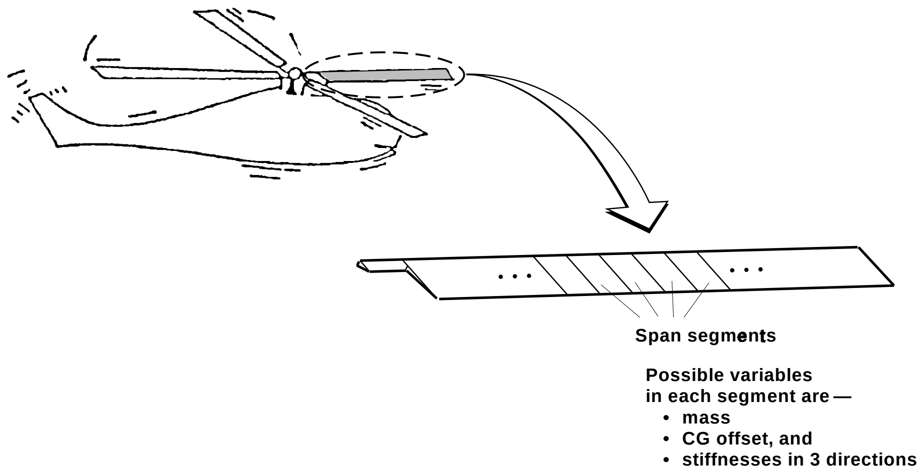


Figure 6: Rotor blade design variables

a mid-level workstation. However, the run time can increase to several days on the same machine if wake updating is used for greater physical fidelity. Our main focus is on the use of approximations to the analysis code results as objective function surrogates for optimization, so to save computational time, the test results discussed here use fixed-wake analyses.

A more detailed statement of the optimization problem is

$$\begin{array}{ll} \text{minimize} & f(x) = \sum_{i=1}^{nh} w_i \dfrac{|h_i(x)|}{|h_i(x_B) + 1|} \\ \text{with respect to} & x \in \Re^n \\ \text{subject to} & \text{xu}_j \geq x_j \geq \text{xl}_j,\ j = 1, \ldots, n \\ & \text{cu}_k \geq c_k(x) \geq \text{cl}_k,\ k = 1, \ldots, ncon. \end{array} \tag{8}$$

In the above equation, the $h_i,\ i = 1, \ldots, nh$, are output responses from Tech01. The subscript i is an index that maps from the response function vector elements to forces in three directions, moments in three directions, and harmonic numbers for each force and moment. In addition, the indexing may span several flight conditions, such as hover and forward flight at various speeds. Normalization to account for the different physical units of the responses is accomplished by including $h_i(x_B)$ in the denominator of the objective function terms, where x_B is the *baseline* design.

The objective function components are weighted by factors w_i. The limits xu_j, xl_j, cu_k, and cl_k are upper and lower bounds on the variables and constraints, respectively. The constraints $c_k(x)$ can be quantities such as required rotor horsepower, centrifugal force, autorotational inertia, snow load, and limits on total mass. Aside from the bounds on the independent variables, the only constraint in the examples considered here is total mass. Since the masses are a subset of the design variables, the mass constraint is a linear constraint involving a subset of the variables. Thus, it is independent of the analysis results, and does not require consideration of issues involving the construction of surrogate approximations of constraint functions.

Example	Variables	Objective	Constraints
31 Variable	10 masses, 10 centers of gravity, 11 stiffnesses in a single direction	weighted sum of 1st and 2nd harmonics for two flight conditions	upper bound on sum of masses

Table 1: Rotor blade design examples

The helicopter rotor blade design problem is summarized in Table 1. Note that this problem has upper and lower bounds on all the variables.

We now summarize the performance of several optimization methods when applied to the helicopter rotor blade design problem described in Section 4.3. We remind the reader that this problem has a linear inequality constraint which we treat by declining to evaluate $f(x)$ for any infeasible x, which is how we will treat linear and bound constraints in all our algorithms. The optimization methods that we considered are the following:

- **MMF**: This is Serafini's [71] implementation of the surrogate management framework. An initial approximation was constructed from 59 successful function evaluations using the DACEPAC software package [10, 11]. The initial iterate was a baseline solution provided by Boeing. SEARCH evaluated the current approximation on a 29,800-point subset of the current mesh and returned the three points with the lowest values. The true objective function was then evaluated sequentially at each of these points until one was found to be better than the current iterate.

- **PDS**: This is Torczon's implementation [77] of the parallel direct search method of [34], with modifications by Serafini to support constraints and the standard Message Passing Interface (MPI) parallel communications library [73]. The initial iterate was the baseline solution provided by Boeing. PDS was executed using 96 evaluations of the objective per iteration, more than the minimal number (62) required to ensure convergence. This algorithm readily takes advantage of a parallel environment.

- **GA**: This is a genetic algorithm from PGAPack [62]. On the advice of its author, David Levine, we used a steady-state reproductive strategy with a population size of 200 and a replacement rate of 10% of the population per iteration.

4.3.1 Comparative results

For each of the above optimization methods, the best objective function value obtained after selected numbers of function evaluations was plotted against the number of function evaluations. The resulting graph, adapted from [14, 71], is displayed in Figure 7. We report the total number of *attempts* to evaluate the objective function, whether or not the attempt was successful. However,

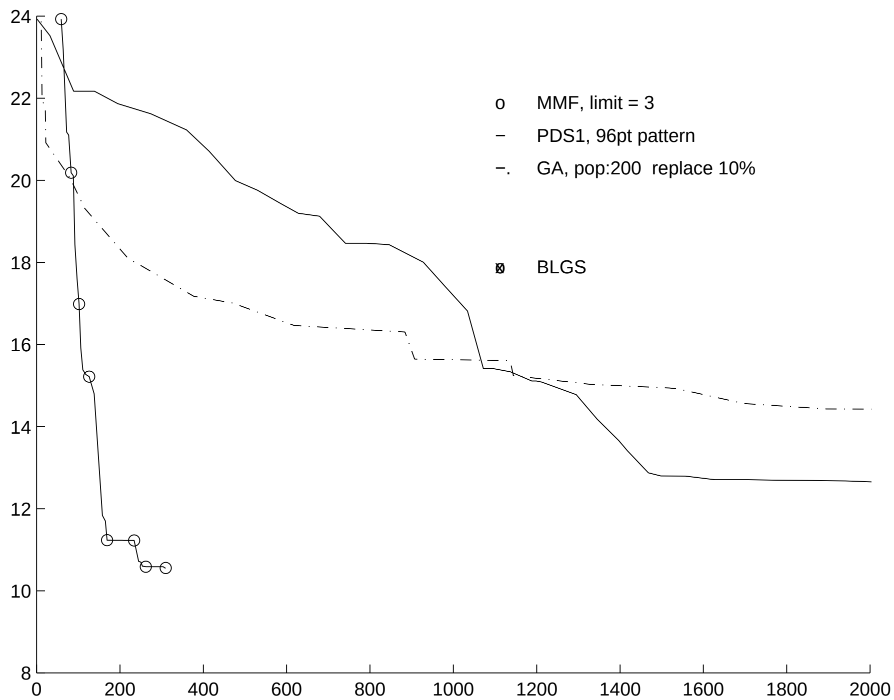


Figure 7: Results for the 31-variable helicopter rotor blade design example

we did not count unsuccessful attempts encountered during the construction of initial approximations, prior to commencement of the optimization algorithm. Thus, for MMF, our count includes the 59 successful function evaluations obtained by DACEPAC, but not the additional 97 evaluation attempts that failed.

The results summarized in Figure 7 are quite encouraging—so good, in fact, that it may be that the 31-variable helicopter rotor blade design problem is substantially easier to solve than we anticipated. Both GA and PDS performed as advertised. GA produced substantial decrease with a small number of function evaluations, but then had difficulty descending below a fairly high value of the objective function. PDS descended somewhat more steadily to an appreciably lower value of the objective function. MMF found even lower objective function values in a number of function evaluations that would be considered extremely small for finite-difference quasi-Newton methods.

The variables in the 31-variable helicopter rotor blade design problem differ by ten orders of magnitude, and so all of the algorithms scale the decision variables to be of comparable magnitudes.

Finally, we observe that most of the above algorithms can exploit parallelism to reduce the “wall clock” time required to get a solution, principally by concurrent evaluations of the objective function. The implementations used here differ with respect to how many concurrent evaluations can be used effectively. In particular, two of the codes, PDS and GA, were designed explicitly to be executed in parallel and so they have the advantage that they can use any number of processors

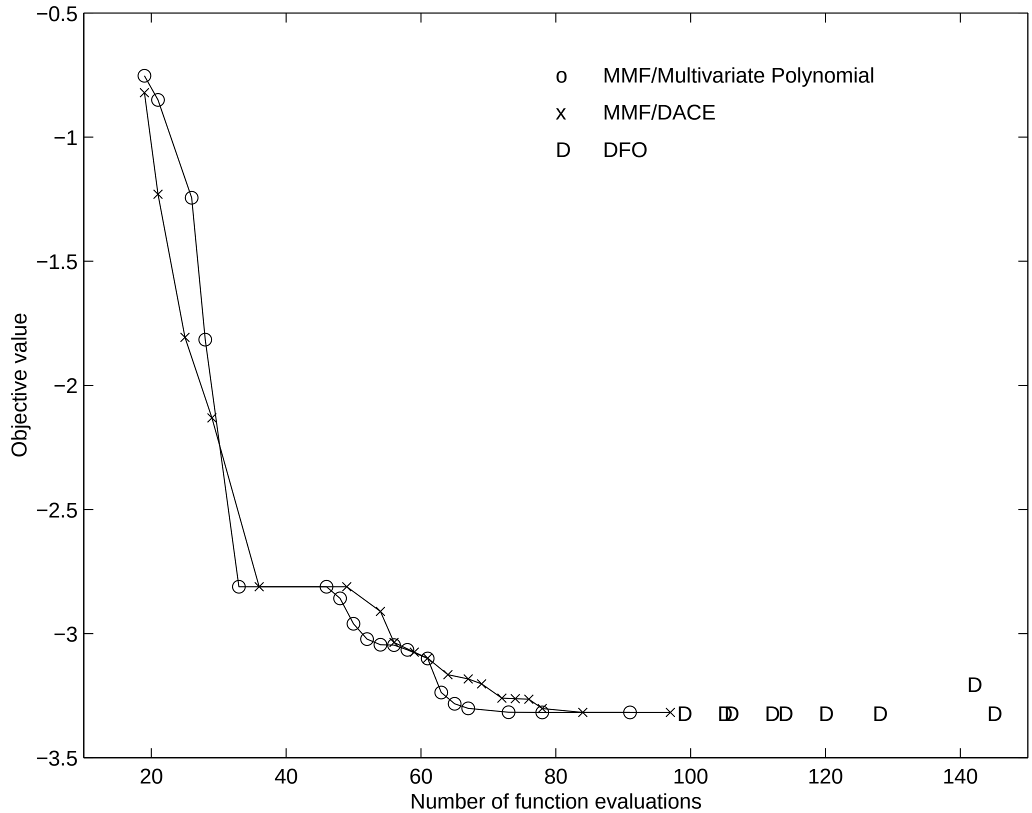


Figure 8: Results for DACE and polynomial surrogates

without any recoding. Given this implementation philosophy, the total number of function evaluations these methods take to reach a solution does not compare favorably with the sequential implementations of the other algorithms. However, when executed in parallel, the "wall clock" time for PDS and GA is more competitive. Nonetheless, for the tests reported here, MMF found feasible solutions with appreciably lower values of the objective and required far fewer total evaluations of the objective function in the process. Work on a parallel version of the surrogate management framework is underway at more than one location.

4.4 Results for polynomial and DACE surrogates

Dave Serafini in his thesis [71] ran various tests comparing DACE and least degree polynomials[30]. The results in Figure8 are typical. The two surrogates were computed from the same initial orthogonal array. The function in Hartmann6, a standard global optimization test problem in 6 variables. Of course, it is very cheap to evaluate, so if one actually wanted to solve this problem, it would not pay to use surrogates.

4.5 Filter results on a Boeing planform design application

The GPS filter algorithm has been applied often to Boeing wing planform design problems. The wing planform is the two-dimensional, downward, vertical projection of the wing. The design variables are the line segment end point for the wing leading edges, trailing edges, and spars. Also there are variables related to wing thickness and aerodynamic loading. A typical design problem is to minimize direct operating cost subject to several constraints. The constraints include required range, maximum approach velocity, maximum required runway length, and several others. The analysis code is a sophisticated combination of preliminary design tools from many disciplines. The disciplines include structures, aerodynamics weights, costing, and configuration management.

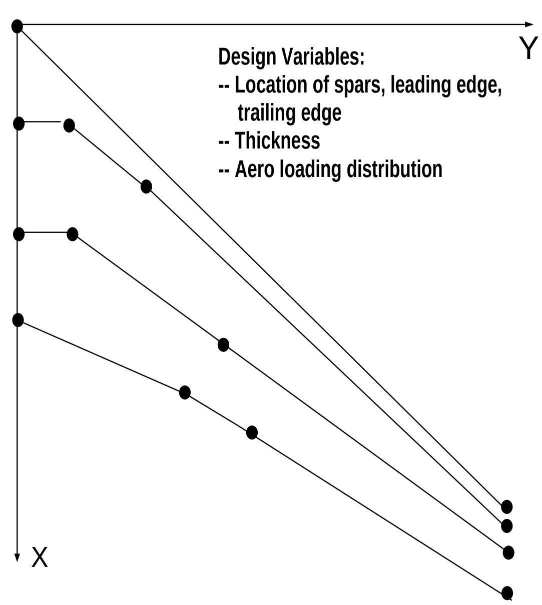
This problem has 17 variables, 13 nonlinear constraints, and no linear constraints. The best point in the initial surrogate (a kriging model that interpolates data from 200 points obtained from an orthogonal-array-based Latin hypercube) is the least infeasible point, which has a constraint violation of 0.426 and an objective of 9.845.

Figure 9 illustrates the progression of the filter for this application. In all plots, the symbol $\times$ represents the initial point, except in the bottom right plot due to the scale change. The top two plots correspond to the first 15 function evaluations, the middle ones the first 50 and the bottom ones are the whole filter after completing the 117 evaluations. The initial point gets filtered at the 3rd function evaluation. The first feasible point is found at the 58th evaluation. The best feasible point is denoted on the two bottom plots by a star at (0, 9.75). The bottom left plot contains several trial points with an objective function near 9.6. This suggests that the SEARCH strategy tried but was unsuccessful in finding a feasible design with such a low f value.

Here are some earlier less detailed results using the filter pattern search method on a pair of planforms:

Infeasible baseline design.

	n	# of ctrs	# of fevals
A	15	11	304
B	15	11	292



The computational results presented here were enriched significantly by the cooperation and collaboration of our Boeing colleagues. Our results showing that SMF can work with polynomial as well as DACE approximations would have been much more difficult without the cooperation of Tom Grandine, who provided his implementation of the least-degree polynomial interpolant [30]. We also thank David Levine for suggesting parameter values for use in PGAPack.

This field is developing rapidly. Probably the most important obstacle ahead is to extend surrogate methods to handle an order of magnitude more variables, say 200. The main difficulty in handling larger problems is the cost of finding the parameters for a response surface type model. Our hunch is that this will drive us to more complex linking between the optimization algorithm and the surrogates.

References

[1] ALEXANDROV N., DENNIS J.E.JR, LEWIS R., TORCZON V. (1998), "A trust region framework for managing the use of approximation models in optimization," *Structural Optimization* 15, 16-23.

[2] ALEXANDROV N., AND KODIYALAM S(1998), "Initial results of an MDO method evaluation study," AIAA Paper 98-4884 *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* ICASE

[3] ALEXANDROV N., LEWIS R.(2000), "Analytical and computational approaches to MDO," AIAA paper 2000-4718 *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* ICASE

[4] AUDET C. (1998), "Convergence results for patterns search algorithms are tight," *TR98-24* Department of Computational & Applied Mathematics, Rice University, Houston Texas.

[5] AUDET C., DENNIS J.E.JR (2000), "Pattern search algorithms for mixed variable programming," *SIAM Journal on Optimization* Vol.11 No.3, 573-594.

[6] AUDET C., DENNIS J.E.JR.(2000), "Analysis of generalized pattern searches," *TR00-07* Department of Computational & Applied Mathematics, Rice University, Houston TX.

[7] AUDET C., DENNIS J.E.JR (2000), "A Pattern Search Filter Method for Nonlinear Programming without Derivatives," *TR00-09*, Department of Computational and Applied Mathematics, Rice University, Houston Texas.

[8] AUDET, C., BOOKER, A.J., DENNIS, J.E.JR., FRANK P.D., and MOORE, D.(2000), "A Surrogate-Model-Based Method For Constrained Optimization", AIAA no.4891, *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*.

[9] Barthelemy, J-F. M. and Haftka, R T. 1993: Approximation concepts for optimum structural design – a review. *Structural Optimization*. **5**, 129–144.

[10] Booker, A. J. 1994: DOE for computer output. Technical Report BCSTECH-94-052, Boeing Computer Services, Research and Technology, M/S 7L–68, Seattle, Washington 98124.

[11] Booker, A. J. 1996: Case studies in design and analysis of computer experiments. In *Proceedings of the Section on Physical and Engineering Sciences*, American Statistical Association.

[12] BOOKER A.J., DENNIS J.E.JR, FRANK P.D., MOORE D.W. and SERAFINI D.B.(1999), "Managing Surrogate Objectives to Optimize a Helicopter Rotor Design - Further Experiments," AIAA Paper 98-4717, St. Louis, September 1998.

[13] Booker, A. J.; Conn, A. R.; Dennis, J. E. Jr; Frank, P. D.; Trosset, M. W. and Torczon, V. 1995: Global modeling for optimization: Boeing/IBM/Rice collaborative project 1995 final report. Technical Report ISSTECH–95–032, Boeing Information & Support Services, Research and Technology, M/S 7L–68, Seattle, Washington 98124.

[14] Booker, A. J.; Dennis, J. E. Jr; Frank, P. D.; Serafini, D. B.; and Torczon, V. 1997: Optimization using surrogate objectives on a helicopter test example. Technical Report SSGTECH-97-027, Boeing Shared Services Group, Applied Research & Technology, M/S 7L–68, Seattle, Washington 98124. Also available as Technical Report 97-31, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005-1892. To appear in Borggaard, J.; Burns, J.; Cliff, E. and Schreck, S., editors, *Optimal Design and Control*. Birkhauser, Cambridge, Massachussets.

[15] BOOKER A.J., DENNIS J.E.JR, FRANK P.D., SERAFINI D.B., TORCZON V. and TROSSET M.W.(1999), "A rigorous framework for optimization of expensive functions by surrogates," *Structural Optimization* Vol.17 No.1, 1-13.

[16] Burgee, S. L.; Giunta, A. A.; Balabanov, V.; Grossman, B.; Mason, W. H.; Narducci, R.; Haftka, R. T. and Watson, L. T. 1996: A coarse-grained parallel variable-complexity multidisciplinary optimization paradigm. *Intl. J. Supercomputing Applications and High Performance Computing*. **10**(4), 269–299.

[17] CHOI T.D., ESLINGER O.J., KELLEY C.T., DAVID J.W., ETHERIDGE M.(1998), "Optimization of automotive valve train components with implicit filtering," to appear in *Optimization and Engineering*.

[18] CHOI T.D., KELLEY C.T.(1999), "Superlinear convergence and implicit filtering," *SIAM Journal on Optimization* Vol.10 No.4, 1149–1162.

[19] CLARKE, F.H.(1990) "Optimization and Nonsmooth Analysis," SIAM Classics in Applied Mathematics Vol 5, Philadelphia.

[20] CONN A.R., GOULD N.I.M., TOINT PH.L.(1991) "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis* Vol 28, 545–572.

[21] Conn, A. R.; Scheinberg, K. and Toint, Ph. L. 1997: On the convergence of derivative-free methods for unconstrained optimization. In Iserles, A. and Buhmann, M., editors, *Approximation Theory and Optimization: Tributes to M. J. D. Powell*, 83–108. Cambridge University Press, Cambridge, United Kingdom.

[22] CONN, A. R., SCHEINBERG, K., AND TOINT, PH.L. (1997). "Recent progress in unconstrained nonlinear optimization without derivatives," *Mathematical Programming* Vol 79, 397-414.

[23] Conn, A. R. and Toint, Ph. L. 1996: An algorithm using quadratic interpolation for unconstrained derivative free optimization. In Di Pillo, G. and Giannessi, F., editors, *Nonlinear Optimization and Applications*, 27–47. Plenum Publishing, New York.

[24] COOPE I.D., PRICE C.J. (2001), "On the convergence of grid-based methods for unconstrained optimization," *SIAM Journal on Optimization* Vol 11, 859-869.

[25] COOPE I.D., PRICE C.J. (2000), "Frame-based methods for unconstrained optimization," *Journal of Optimization Theory and Applications* Vol 107, 261-274.

[26] COOPE I.D., PRICE C.J. (2002), "Grids and Frames in unconstrained optimization," in preparation.

[27] CRAMER EVIN, DENNIS J.E., FRANK P. D., LEWIS R. M., SHUBIN G. R. (1994) "Problem Formulation for Multidisciplinary Optimizaton," *SIAM Journal on Optimization*, Vol 4, 754–776.

[28] Currin, C.; Mitchell, T.; Morris, M. and Ylvisaker, D. 1988: A Bayesian approach to the design and analysis of computer experiments. Technical Report ORNL–6498, Oak Ridge Laboratory.

[29] Davis, C. 1954: Theory of positive linear dependence. *American Journal of Mathematics*. **76**, 733–746.

[30] DE BOOR C., RON, A. (1992), "Computational aspects of polynomial interpolation in several variables," *Mathematics of Computation*, Vol 58, 705–727.

[31] DENNIS J.E.JR(2000), "Danish Mtg on surrogate optimization", *TR00-30*, Department of Computational and Applied Mathematics, Rice University, Houston Texas.

[32] DENNIS J.E.JR, EL-ALEM M. and WILLIAMSON K.(1999), "A trust-region approach to nonlinear systems of qualities and inequalities," *SIAM Journal on Optimization* Vol.9 No.2, 291–315.

[33] , DENNIS, J.E. JR and R.B. SCHNABEL, "Numerical Methods for Unconstrained Optimization and Nonlinear Equations," Prentice-Hall 1983, presently SIAM Classics in Applied Mathematics vol 16, 1996, Philadelphia.

[34] Dennis, J. E. Jr and Torczon, V. 1991: Direct search methods on parallel machines. *SIAM J. Optimization*. **1**(4), 448–474.

[35] Dennis, J. E. Jr and Torczon, V. 1997: Managing approximation models in optimization. In Alexandrov, N. and Hussaini, M. Y., editors, *Multidisciplinary Design Optimization: State-of-the-Art*, 330–347. SIAM, Philadelphia. Also available as Technical Report 95-19, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005–1892.

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

[37] Dixon, L. C. W. and Szegö G. P., editors. 1978: *Towards Global Optimization 2*. North-Holland Pub. Co., Amsterdam.

[38] DOLAN E., LEWIS M., TORCZON V.(2000), "On the local convergence of pattern search", ICASE Technical Report 2000-36.

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

[40] FERRIS M.C., MANGASARIAN O.L. (1994), "Parallel Variable Distribution,", *SIAM Journal on Optimization* Vol 4, 815-832.

[41] FLETCHER R., LEYFFER S.(1997), "Nonlinear Programming without a penalty function," Dundee University, Dept. of Mathematics, Report NA/171.

[42] FLETCHER R, LEYFFER S., TOINT PH.L.(1998), "On the global convergence of an SLP-Filter algorithm," Dundee University, Dept. of Mathematics, Report NA/183.

[43] FLETCHER R, GOULD N.I.M., LEYFFER S., TOINT PH.L.(1999), "On the global convergence of trust-region SQP-Filter algorithms for general nonlinear programming," Department of Mathematics, FUNDP, Namur (B), Report 99/03.

[44] Frank, P. D. 1995: Global modeling for optimization. *SIAG/OPT Views-and-News*. **7**, 9–12.

[45] Geisser, S. and Eddy, W. F. (1979) A predictive approach to model selection. *Journal of the American Statistical Association*. **74**, 153–160.

[46] GILMORE P, KELLEY, C.T. (1995), "An implicit filtering algorithm for optimization of functions with many local minima," *SIAM Journal on Optimimization*, Vol. 5, 269285.

[47] Giunta, A. A. 1997: *Aircraft Multidisciplinary Optimization using Design of Experiments Theory and Response Surface Modeling Methods*. PhD thesis, Virginia Tech. Available as MAD 97-05-01, May 1997, Department of Aerospace and Ocean Engineering, Virginia Tech, 215 Randolph Hall, Blacksburg, Virginia 24061.

[48] JOHN F.(1948), "Extremum problems with inequalities as subsidiary conditions," in *Studies and essays: Courant Anniversary Volume*, (eds FRIEDRICHS K.O., NEUGEBAUER O.E. and STOKER J.J.), Wiley-Interscience, New-York, 187–204.

[49] HILAL M.A. and BOOM R.W.(1977), "Optimization of Mechanical Supports for Large Superconductive Magnets," *Advances in Cryogenic Engineering* 22, 224–232.

[50] HIRIART-URRUTY J.-B., LEMARÉCHAL C.(1993) "Convex analysis and minimization algorithms," Springer-Verlag Berlin, New-York.

[51] HOUGH P.D., KOLDA T.G., TORCZON V.(2000), "Asynchronous parallel pattern search for nonlinear optimization," Sandia National Laboratories, Report SAND2000-8213.

[52] KOKKOLARAS M., AUDET C. and DENNIS J.E.JR(2000), "Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system," *Optimization and Engineering*, Vol.2 No.1 to appear.

[53] LEWIS R.M., TORCZON V.(1999), "Pattern search algorithms for bound constrained minimization," *SIAM Journal on Optimization*, Vol.9 No.4, 1082-1099.

[54] LEWIS R.M., TORCZON V.(1996), "Rank ordering and positive basis in pattern search algorithms," *ICASE NASA Langley Research Center* TR 96-71.

[55] LEWIS R.M., TORCZON V.(2000), "Pattern search methods for linearly constrained minimization," *SIAM Journal on Optimization*, Vol.10 No.3, 917–941.

[56] LEWIS R.M., TORCZON V.(1998), "A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds," *ICASE NASA Langley Research Center* TR 98-31.

[57] POWELL, M.J.D. (2000) "UOBYQA: unconstrained optimization by quadratic approximation," Numerical Analysis Report DAMTP 2000/NA14, University of Cambridge.

[58] VANDERPLAATS G.N.(1984), *Numerical optimization techniques for engineering desing: with applications*, McGraw-Hill, New-York, London.

[59] TORCZON V.(1997), "On the Convergence of Pattern Search Algorithms," *SIAM Journal on Optimization* Vol.7 No.1, 1–25.

[60] Jones, D. R.; Schonlau, M. and Welch, W. J. 1997: A data analytic approach to Bayesian global optimization. In *Proceedings of the ASA*.

[61] Koehler, J. R. and Owen, A. B. 1996: Computer experiments. In Ghosh, S. and Rao, C. R., editors, *Handbook of Statistics, Volume 13*, 261–308. Elsevier Science, New York.

[62] Levine, D. 1996: Users guide to the PGAPack parallel genetic algorithm library. Technical Report ANL-95/18, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439. Available from URL ftp://info.mcs.anl.gov/pub/pgapack/pgapack.tar.Z.

[63] Lewis, R. M. and Torczon, V. 1996: Pattern search algorithms for bound constrained minimization. Technical Report 96–20, ICASE, NASA Langley Research Center, Hampton, Virginia 23681–2199. To appear in *SIAM J. Optimization*.

[64] Lewis, R. M. and Torczon, V. 1996: Rank ordering and positive bases in pattern search algorithms. Technical Report 96–71, ICASE, NASA Langley Research Center, Hampton, Virginia 23681–2199. In revision for *Mathematical Programming*.

[65] Lewis, R. M. and Torczon, V. 1997: Pattern search methods for linearly constrained minimization. Technical Report 98–03, ICASE, NASA Langley Research Center, Hampton, Virginia 23681–2199. Submitted to *SIAM J. Optimization*.

[66] McKay, M. D.; Conover, W. J. and Beckman, R. J. 1979 A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*. **21**(2), 239–245.

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

[68] Rinnooy Kan, A. H. G. and Timmer, G. T. 1984: A stochastic approach to global optimization. In Boggs, P. T.; Byrd, R. H. and Schnabel, R. B., editors, *Numerical Optimization 1984: Proceedings of the SIAM Conference on Numerical Optimization*, 245–262. SIAM, Philadelphia.

[69] Ripley, B. D. 1988: *Statistical Inference for Spatial Processes*. Cambridge University Press, Cambridge, United Kingdom.

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

[71] Serafini, D. B. 1998: A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions. Ph.D. Thesis, Rice University.

[72] Shultz, L. A.; Panda, B.; Tarzanin, F. J.; Derham, R. C.; Oh, B. K. and Dadone, L. 1994: Interdisciplinary analysis for advanced rotors - approach, capabilities and status. AHS Aeromechanics Specialists Conference, January 19–21, 1994, PS 4-1–4-15.

[73] Snir, M.; Otto, S. W.; Huss-Lederman, S.; Walker, D. W. and Dongarra, J. 1996: *MPI: The Complete Reference*. The MIT Press, Cambridge, Massachusetts.

[74] Stein, M. 1987: Large sample properties of simulations using latin hypercube sampling. *Technometrics*. **29**(2), 143–151.

[75] Stephens, C. P. and Baritompa, W. 1998: Global optimization requires global information. *J.O.T.A.* **96**, 575-588.

[76] Tang, B. 1993: Orthogonal array-based latin hypercubes. *Journal American Statistical Association*. **88**(424), 1392–1397.

[77] Torczon, V. 1992: PDS: direct search methods for unconstrained optimization on either sequential or parallel machines. Technical Report 92–9, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005–1892. In revision for *ACM Transactions on Mathematical Software*.

[78] Torczon, V. 1995: Pattern search methods for nonlinear optimization. *SIAG/OPT Views and News*. **6**, 7–11.

[79] Torczon, V. 1997: On the convergence of pattern search algorithms. *SIAM J. Optimization*. **7**(1), 1–25.

[80] Torczon, V. and Trosset, M. W. 1998: From evolutionary operation to parallel direct search: pattern search algorithms for numerical optimization. *Computing Science and Statistics*. **29**, 396–401.

[81] Trosset, M. W. and Torczon, V. 1997: Numerical optimization using computer experiments. Technical Report 97–38, ICASE, NASA Langley Research Center, Hampton, Virginia 23681–2199.

[82] Trosset, M. W. 1997: I know it when I see it: toward a definition of direct search methods. *SIAG/OPT Views and News*. **9**, 7–10.

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

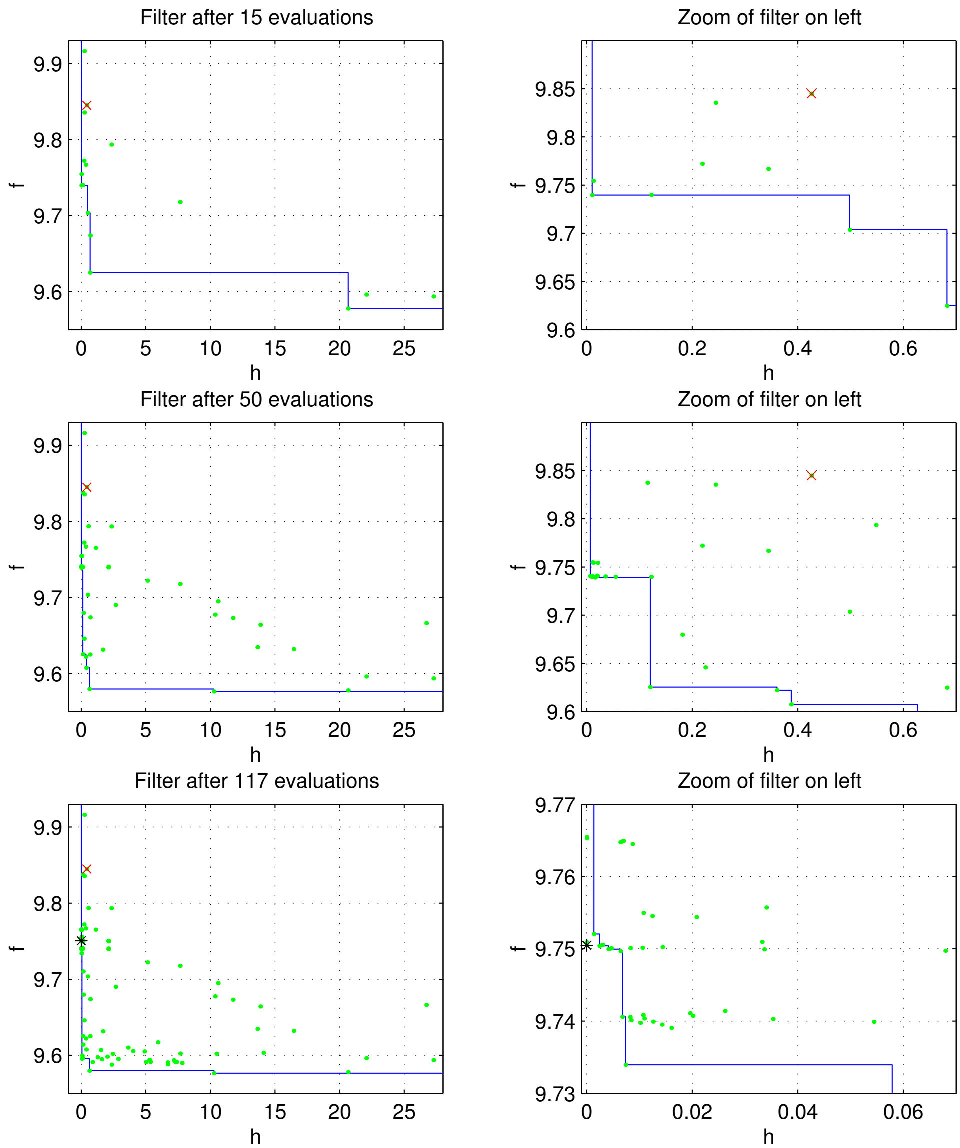


Figure 9: Filter progression on a Boeing planform design application