AIAA JOURNAL
Vol. 44, No. 4, April 2006

Statistical Improvement Criteria for Use in Multiobjective Design Optimization

A. J. Keane*
University of Southampton, Southampton, England SO17 1BJ, United Kingdom

Design of experiment and response surface modeling methods are applied to the problem of constructing Pareto fronts for computationally expensive multiobjective design optimization problems. The work presented combines design of experiment methods with kriging (Gaussian process) models to enable the parallel evolution of multiobjective Pareto sets. This is achieved via the use of updating schemes based on new extensions of the expected improvement criterion commonly applied in single-objective searches. The approaches described provide a statistically coherent means of solving expensive multiobjective design problems using single-objective search tools. They are compared to the use of nondominated sorting genetic algorithm (NSGA-ii) based multiobjective searches, both with and without response surface support. The new approaches are shown to give more exact, wider ranging, and more evenly populated Pareto fronts than the genetic algorithm based searches at reduced or similar cost.

Nomenclature

A	=	cross-sectional area of beam
b	=	breadth of beam
Cd	=	drag coefficient
D^0	=	set of initial response surface modeling training data
$d(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$	=	distance measure between two design vectors
E	=	Young's modulus
$E[I]$	=	expected improvement
F	=	force on tip of beam
F_{crit}	=	critical force for twist buckling
f	=	safety factor on critical force for twist buckling
$\hat{f}_e(\boldsymbol{x})$	=	approximation of expensive goal or response function
$f_e^{\min}(x)$	=	actual response at current best design vector
$f_{1,2}^*$	=	two-dimensional Pareto set of objective function values
G	=	modulus of rigidity
h	=	height of beam
l	=	length of beam
M	=	number of designs in Pareto set
N	=	number of points used in design of experiment
$P[I]$	=	probability of improvement
p_h	=	krig hyperparameter
$\boldsymbol{R}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$	=	correlation function between two design vectors
$\boldsymbol{r}(\boldsymbol{x})$	=	vector of correlations between new design point and existing data
$s^2(\boldsymbol{x})$	=	mean squared error
x	=	design variable
$\boldsymbol{x}$	=	vector of design variables
$y(\boldsymbol{x})$	=	response variable being modeled
$\hat{y}(\boldsymbol{x})$	=	approximation of response variable at $\boldsymbol{x}$
$\bar{\hat{y}}_{1\text{aug}}(x^{(N_0+1)})$	=	predicted response for design goal 1 that should augment Pareto front
$\bar{\hat{y}}_{1\text{dom}}(x^{(N_0+1)})$	=	predicted response for design goal 1 that should dominate at least one point in Pareto front
δ	=	tip deflection
δ_{ij}	=	Dirac delta function
$\varepsilon(\boldsymbol{x})$	=	Gaussian random function with zero mean and variance σ^2
θ_h	=	krig hyperparameter
Λ	=	krig regularization parameter
μ	=	mean of input responses
σ^2	=	variance
σ_B	=	maximum bending stress
σ_Y	=	yield stress
τ	=	maximum shear stress
υ	=	Poisson's ratio
$\Phi()$	=	normalized Gaussian distribution function
$\phi()$	=	normalized Gaussian density function

Introduction

WHEN dealing with design optimization problems it is common to encounter multiple and conflicting goals. In general, these can be dealt with in one of two ways: Either the goals must be combined to form a composite single objective or, alternatively, a so-called Pareto set of nondominated designs must be sought. There is significant literature on both approaches. See Keane and Nair[1] for an introduction to their respective advantages and disadvantages. When the design process being dealt with involves the use of expensive high-fidelity simulation codes to evaluate the competing objectives, selecting the most appropriate designs to study becomes particularly difficult. In this paper, attention is focused on leveraging ideas developed for solving expensive, single-objective optimization problems in a multiobjective Pareto front setting. Tools from the single-objective, global response surface modeling (RSM) literature, for example, Myers and Montgomery,[2] are adapted to multiobjective problems to minimize the computational costs involved. Specifically, design of experiment (DOE) techniques[3,4] and kriging (Gaussian processes) (see Jones et al.[5] and Jones[6]) are used to develop a coherent framework for exploiting results from maximum likelihood theory in this context. Variants on these methods have been used in aerospace design for some time.[1] However, so far they have mostly been used to accelerate single-objective optimization approaches using expensive codes.[7] Here, for what is believed to be the first time, approaches based on these ideas are applied directly to the construction of multiobjective Pareto sets.

This paper is laid out as follows: First, the basic RSM approach to optimization is briefly summarized, along with the use of DOE techniques in generating data for initial RSM construction. Next, the

Received 29 March 2005; revision received 29 September 2005; accepted for publication 29 September 2005.

*Professor of Computational Engineering and Director, BAE Systems/Rolls–Royce University Technology Partnership for Design Search and Optimisation, School of Engineering Sciences, Highfield.

steps needed to construct kriging models are set out, followed by an introduction to probabilistic improvement metrics for use in RSM updating. Then the underlying search methods needed for this work are noted before moving to a first example that illustrates the use of standard probabilistic improvement schemes for single-objective kriging models. Multiobjective problems and the construction of Pareto fronts are then considered, followed by the introduction of two novel probabilistic Pareto front improvement metrics based on the ideas introduced for single-objective kriging-based RSM development. These new metrics are then applied to two example problems and compared to conventional multiobjective genetic algorithm (GA)-based searches, both with and without RSM support. The paper is closed with some conclusions and ideas for further work based on these new ideas.

RSM and DOE Methods

RSMs are surrogate metamodels produced by curve-fitting techniques to samples of computationally expensive data. They are widely used in the design community when carrying out optimization studies on expensive simulation codes. (Note that throughout this paper it is assumed that all of the goals being studied need to be minimized. Problems involving maximization can be dealt with by negating the functions being dealt with.) The basic RSM process involves selecting a limited number of points at which the expensive code will be run, normally using formal DOE methods. Then, when these designs have been analyzed, usually in parallel, a response surface (curve fit) is constructed through or near the data. Design optimization is then carried out on this surface to locate new and interesting combinations of the design variables, which may then, in turn, be fed back into the full code. These data can then be used to update the metamodel, and the whole process is repeated until the user either runs out of effort, some form of convergence is achieved, or sufficiently improved designs are reached. This process is shown in Fig. 1, based on an expensive computational fluid dynamics (CFD) code.

The update process central to this strategy must attempt to address two conflicting goals when building RSMs. First, there is the need to ensure that the RSM is a reasonably accurate model throughout the design space of interest. (This aspect is commonly termed exploration.) Second, there is the aim of rapidly converging the process to the global optimum (exploitation). It will be obvious that these goals are generally in conflict in that to learn the most about the accuracy of the model we commonly wish to place new calculations in regions that have yet to be sampled, whereas to exploit most rapidly the model one seeks to search in the (apparently) most promising basins of attraction. If one exploits too early or too severely, there is a danger of missing an entire region of high-quality designs in the rush for rapid closure.

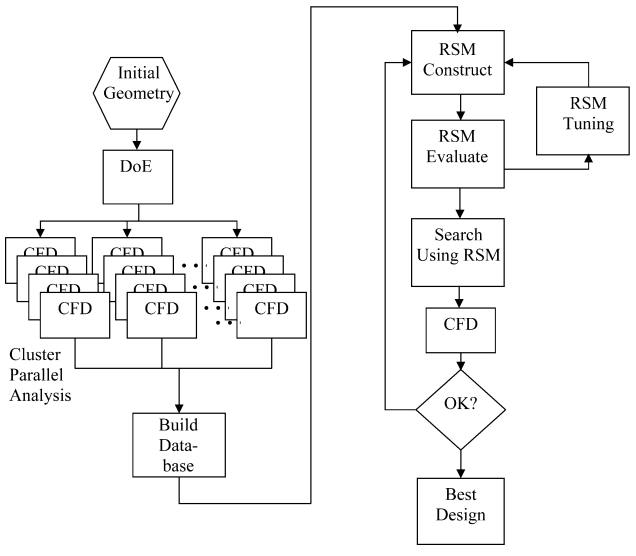


Fig. 1 RSM-based optimization strategy.

Perhaps the most appealing mechanisms for dealing with this dichotomy are based on probability of improvement measures.[5] Formally, the probability of improvement $P[I]$ is the expected (mean) chance that any new design calculation will provide an improvement over the current best design. It is also possible to make use of a quantity derived from such analysis known as the expectation of improvement or expected improvement, denoted here as $E[I]$. $E[I]$ is the mean value of any likely improvement that can be obtained by sampling the RSM in a new location (more formally the first moment of $P[I]$ about the current best point). As will be shown later, it is often possible to set up rather elegant frameworks for evaluating and using these quantities to direct any search process.

Given their popularity and power, it is no surprise that there are a number of variations and refinements that may be applied to the basic RSM approach. The literature offers many possible alternatives, for example, that of Myers and Montgomery.[2] Here, $LP\tau$ DOE sequences[8] and Latin hypercube designs[4] are used to generate the initial set of points and kriging (Gaussian process) models applied to build the RSMs (see Jones et al.[5]).

Most DOE methods seek to sample the entire design space efficiently by building an array of possible designs with relatively even, but not constant, spacing between the points, spanning all dimensions. These properties reduce biases and allow for cases where there is no a priori knowledge of the relative importance of design variables. Notice that this is in contrast to a pure random spacing, which would result in some groups of points occurring in clumps, whereas there were other regions with relatively sparse data. This might be desirable if there were no correlation between the responses at points, however close they were to each other, that is, if the process resembled white noise, but this is highly unlikely in engineering design problems. A particular advantage of the $LP\tau$ approach is that not only does it give good coverage for engineering purposes, but that it also allows additional points to be added to a design without the need to reposition existing points. This can be useful if the designer is unsure how many points will be needed before commencing work. Then, if the initial build of the RSM is found to be inadequate, a new set of points can be inserted without invalidating the statistical character of the experiment. (Similarly, if for some reason the original experiment cannot be completed, the sequence available at any intermediate stage will still have useful coverage.) The Latin hypercube approach offers the advantage that sets of designs with broadly similar properties can be readily set up to study the average performance of any search process taken over different sample sets.

Kriging

In this work, a kriging approach is used to represent the computed data[5] because this provides a theoretically sound basis for judging the degree of curvature needed to model adequately the user's data as well as allowing control of the degree of regression. Additionally, it provides measures of probable errors in the model being built that can be used when assessing where to place any further design points and, in particular, readily supports the calculation of $P[I]$ and $E[I]$. Kriging is not a panacea for all evils, however. It is commonly found that it is difficult to set up such models for more than 15–20 variables and also that the approach is numerically expensive if there are more than a few hundred data points because the setup (hyperparameter tuning) process requires the repetitive lower–upper (LU) decomposition of a correlation matrix that is dense and that also has the same dimensions as the number of points used. Moreover, the number of such LU steps is strongly dependent on the number of variables in the problem, and the likelihood used in tuning the hyperparameters is also commonly highly multimodal. The author has found that it is difficult to deal with krigs involving more than approximately 20 variables and 500 data points.

In kriging, the inputs $\boldsymbol{x}$ are assumed to be related to the outputs (responses) y by an expensive function $f_e(\boldsymbol{x})$, that is, $y = f_e(\boldsymbol{x})$. In the CFD example presented later, this function is the VGK full-potential code drag prediction for an airfoil section. The response of the code is then evaluated for combinations of inputs generated by the DOE and used to construct an approximation

$$\hat{y} = \hat{f}_e(\boldsymbol{x}) \tag{1}$$

The response at any $\boldsymbol{x}$ is then approximated by

$$\hat{y} = \mu + \varepsilon(\boldsymbol{x}) \tag{2}$$

In kriging, ε is assumed to depend on the distance between corresponding points. The distance measure used here is

$$d\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \sum_{h=1}^{k} \theta_h \left(x_h^{(i)} - x_h^{(j)}\right)^{ph} \tag{3}$$

where θ_h and p_h are hyperparameters tuned to the data in hand. The correlation between points $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ is given by

$$\boldsymbol{R}\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \exp\left[-d\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)\right] + \Lambda\delta_{ij} \tag{4}$$

where Λ is a regularization constant that governs the degree of regression in the model. (When set to zero, the krig strictly interpolates the data supplied.) When the response at a new point $\boldsymbol{x}$ is required, a vector of correlations between the point and those used in the DOE is formed, $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x}, \boldsymbol{x}^{(i)})$. The prediction is then given by

$$\hat{y}(\boldsymbol{x}) = \mu + \boldsymbol{r}^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\mu) \tag{5}$$

where the mean μ is found from

$$\mu = \mathbf{1}^T\boldsymbol{R}^{-1}\boldsymbol{y}/\mathbf{1}^T\boldsymbol{R}^{-1}\mathbf{1} \tag{6}$$

The hyperparameters θ_h and p_h and regularization constant Λ are all obtained by maximizing the likelihood, defined as

$$\frac{1}{(2\pi)^{N/2}(\sigma^2)^{N/2}|\boldsymbol{R}|^{\frac{1}{2}}} \exp\left[\frac{-(\boldsymbol{y} - \mathbf{1}\mu)^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\mu)}{2\sigma^2}\right] \tag{7}$$

where the variance σ^2 is given by

$$\sigma^2 = [(\boldsymbol{y} - \mathbf{1}\mu)^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\mu)]/N \tag{8}$$

The mean squared error of the prediction is (ignoring uncertainty introduced from estimating the covariance parameters)

$$s^2(\boldsymbol{x}) = \sigma^2\left[1 + \Lambda + \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{(1 - \mathbf{1}^T\boldsymbol{R}^{-1}\boldsymbol{r})^2}{\mathbf{1}^T\boldsymbol{R}^{-1}\mathbf{1}}\right] \tag{9}$$

which gives a measure of the accuracy of the krig at $\boldsymbol{x}$. This basic approach can be used to model any response quantity, including constraints.

When any form of global RSM is built, it is good practice to update the model at judiciously chosen points once the initial RSM has been built. Such updating is shown in Fig. 1. As has already been mentioned, the choice of updating scheme can be critical in achieving an efficient search: The update process must balance the needs of converging quickly to any optima in the underlying problem while also allowing for wide ranging exploration of the search space. Here this issue is dealt with by using the concepts of probability of, and expectation of, improvement, a methodology that fits very readily into the kriging framework.

Probability of Improvement and Expected Improvement

Suppose that an initial set of training data is made available by running the high-fidelity model at the points generated by a space-filling DOE technique, that is, $D^0 \equiv \{\boldsymbol{x}^{(i)}, y(\boldsymbol{x}^{(i)})\}, i = 1, 2, \ldots, N_0$. This data set can be used to construct a krig model to approximate the input–output relationship where the mean squared error of the prediction gives an estimate of the uncertainty involved in making predictions using a finite set of input–output data. From the viewpoint of DOE, to improve the accuracy of the baseline surrogate, it is sensible to augment the data set D^0 with additional points where this error is high. However, from the perspective of finding iterates that lead to reductions in the objective function, the aim is to minimize $\hat{y}(\boldsymbol{x})$. Statistical improvement criteria attempt to make this balance in a rational way.

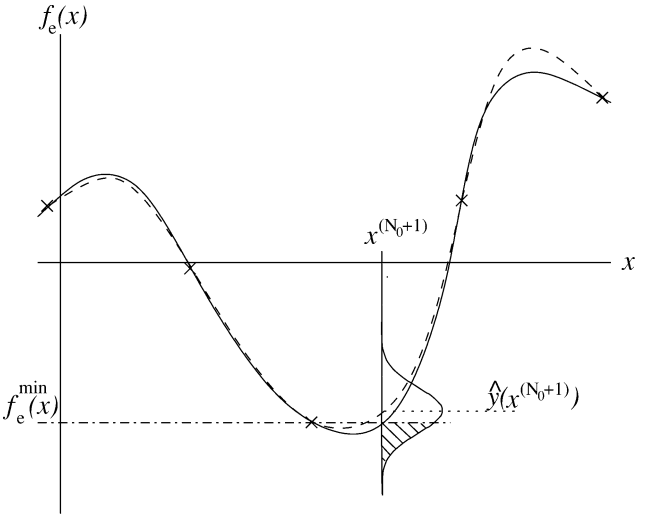


Fig. 2 Probability of improvement and expected improvement for one-dimensional test function: ——, true function $f_e(x)$; ×, sample points; and - - - -, krig.

Because a krig model is a Gaussian process, the probability of any newly calculated design point $y(x^{(N_0+1)})$ representing an improvement over the current best design, $f_e^{\min}(\boldsymbol{x}) = \min[f_e^{(1)}(\boldsymbol{x}^{(1)}), f_e^{(2)}(\boldsymbol{x}^{(2)}), \ldots, f_e^{(N_0)}(\boldsymbol{x}^{(N_0)})]$, is readily calculated from

$$\begin{aligned} P[I] &= P\left[y\left(x^{(N_0+1)}\right) \le f_e^{\min}(x)\right] \\ &= \int_{-\infty}^{f_e^{\min}(x)} \frac{1}{s\left(x^{(N_0+1)}\right)\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{\left[\hat{y} - \mu\left(x^{(N_0+1)}\right)\right]^2}{s^2\left(x^{(N_0+1)}\right)}\right\} \mathrm{d}\hat{y} \\ &= \Phi\left[\frac{f_e^{\min}(x) - \mu\left(x^{(N_0+1)}\right)}{s\left(x^{(N_0+1)}\right)}\right] \end{aligned} \tag{10}$$

where [ ] denotes the expectation operator. This quantity is just the area in the tail of the Gaussian distribution below the current best function value $f_e^{\min}(\boldsymbol{x})$ and indicates the probability that any new design will represent an improvement over those in the existing training data, but does not indicate how much of an improvement will be obtained. This quantity is given by the area of the hatched region shown in Fig. 2, that is, that lying under the Gaussian curve (whose properties are set by the krig at the given design point) and below the current best design.

In contrast, the expected improvement criterion[5] enables the prediction and error of the model to be more strongly biased in terms of exploitation than of exploration. Given the baseline training data set, the magnitude of the improvement that is likely to arise at a new point, $\boldsymbol{x}^{(N_0+1)}$, is given by $I(x^{(N_0+1)}) = \max[f_e^{\min}(x) - y(x^{(N_0+1)}), 0]$. Notice that the max [ ] function here encapsulates that it is quite likely that a new design will, in fact, be worse than those already found, particularly later on in the design process and, of course, such designs will not contribute any improvement. Because $\hat{y}(x^{(N_0+1)})$ is taken to be a random variable in kriging, given $\boldsymbol{x}^{(N_0+1)}$, the expected improvement can be defined as $E[I(x^{(N_0+1)})] = E[\max(f_e^{\min}(x) - \hat{y}(x^{(N_0+1)}), 0)]$. By the use of $\hat{y}(x^{(N_0+1)})$ being Gaussian, this can be written in closed form as

$$\begin{aligned} E\left[I\left(\boldsymbol{x}^{(N_0+1)}\right)\right] &= \left[f_e^{\min}(\boldsymbol{x}) - \hat{y}\left(\boldsymbol{x}^{(N_0+1)}\right)\right] \\ &\quad \times \Phi\left[\frac{f_e^{\min}(\boldsymbol{x}) - \hat{y}\left(\boldsymbol{x}^{(N_0+1)}\right)}{s\left(\boldsymbol{x}^{(N_0+1)}\right)}\right] \\ &\quad + s\left(\boldsymbol{x}^{(N_0+1)}\right)\phi\left[\frac{f_e^{\min}(\boldsymbol{x}) - \hat{y}\left(\boldsymbol{x}^{(N_0+1)}\right)}{s\left(\boldsymbol{x}^{(N_0+1)}\right)}\right] \end{aligned} \tag{11}$$

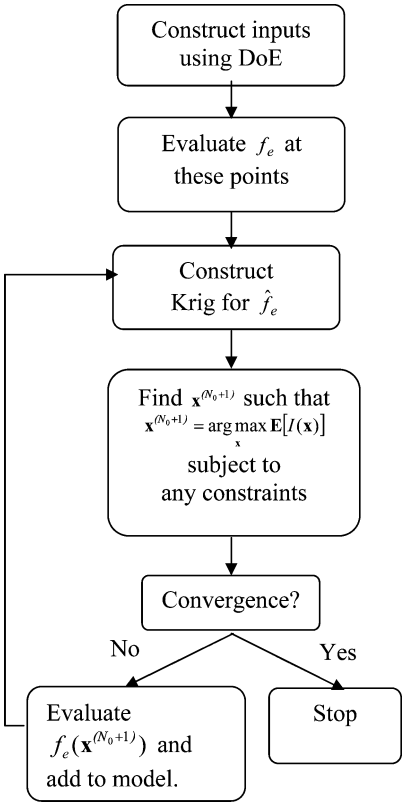


Fig. 3 Kriging procedure for function f_e using $E[I]$.

In Fig. 2, a new point is to be calculated at $x^{(N_0+1)}$ for which the estimated mean value from the krig is $\hat{y}(x^{(N_0+1)})$. The Gaussian process probability density function (PDF) at this location is shown and the area shaded under the Gaussian curve is equal to $P[I]$. $E[I]$ is the first moment of this area about the line through the current best estimate $f_e^{\min}(x)$. Note that the expected improvement gives a measure of how large an improvement will be achieved in the same units as the problem, whereas $P[I]$ is a simple nondimensional probability measure. This is a perhaps subtle distinction that will become important when considering multiobjective problems later in this paper.

In summary, to minimize the original objective function $f_e(\boldsymbol{x})$ the procedure set out in Fig. 3 is followed: First a baseline krig model is constructed using N_0 points generated by applying a space-filling DOE technique. Subsequently, a new iterate is generated by maximizing either the probability of improvement or the expected improvement criteria, that is, $\boldsymbol{x}^{(N_0+1)} = \arg\max_x \boldsymbol{P}[y(\boldsymbol{x}^{(N_0+1)}) \le f_e^{\min}(\boldsymbol{x})]$ or $\boldsymbol{x}^{(N_0+1)} = \arg\max_x E[I(\boldsymbol{x}^{(N_0+1)})]$. The high-fidelity model is then evaluated at $\boldsymbol{x}^{(N_0+1)}$ and the resulting exact function value, $f_e(\boldsymbol{x}^{(N_0+1)})$, is added to the baseline training data set to give D^1. The augmented data set is used to update the krig and its hyperparameters, which are then used to solve the preceding equation for the next iterate. This process is continued until specified convergence criteria are met, for example, when the expected improvement is less than 1% of the best function value obtained so far.

Jones et al.[5] conjecture that, for continuous functions, algorithms based on maximization of these criteria converge to the global optimum. This is based on the argument that if the number of updates tends to infinity, then the design space will be sampled at all locations and, hence, the global optima will eventually be located. A similar argument can be advanced for the metrics to be introduced later in the context of multiobjective problems. However, in practice, due to an inevitably finite computational budget, iterations are usually terminated when the number of points in the training data set reaches a few hundred. Care also needs to be exercised to circumvent numerical problems that may arise due to ill conditioning of the correlation matrix. This may happen when any two points in the update sequence lie too close to each other. One way to avoid this problem is to use a regression model instead of an interpolator. Also note that the landscape of these improvement criteria is invariably highly multimodal. Hence, stochastic search techniques are typically required to locate maxima. Alternatively, branch-and-bound algorithms can be employed. Here, a combined GA and dynamic hill climbing (DHC) search is used.

Search Algorithms

The GA used is fairly conventional[9] except that it incorporates a version of MacQueen's adaptive Kmean algorithm (see Yin and Germay[10]), a clustering algorithm that has been applied with some success to multipeak problems. It works with 12-bit binary encoding, inversion, an elitist survival strategy that ensures that the best of each generation always enters the next generation, has five major control parameters, and has a one-pass external constraint penalty function and optional niche forming.[11]

The main control parameters used in this GA are P[best], the proportion of the population that survives to the next generation (default 0.8); P[cross], the proportion of the surviving population that is allowed to breed (default 0.8); P[invert], the proportion of the surviving population that have their genetic material reordered (default 0.5); P[mutation], the proportion of the new generation's genetic material that is randomly changed (default 0.005); and a proportionality flag that selects whether the new generation is biased in favor of the most successful members of the preceding generation or, alternatively, all P[best], survivors are propagated equally. (The default is to bias in favor of successful members.)

Like all evolutionary methods, the GA is rather slow and inaccurate for problems with few variables but comes into its own as the number of variables grows. It is also not suitable for problems without bounds on all of the variables. Because of their limitations in gaining accurate convergence, the GA is here followed up by a multistart downhill search when searching the krig metamodels or when tuning its hyperparameters, DHC.[12] DHC specifically aims to explore widely spaced basins of attraction and to converge these accurately. It is, thus, well suited to the kind of problems tackled in this work.

First Example and Some Basic Searches

Having set out the basic single-objective RSM approach, its use is illustrated by application to an aerodynamic test case. In this first example problem, the aim is to minimize the drag coefficient on a two-dimensional airfoil section operating at fixed lift and a Mach number of 0.78. The CFD method used to solve this problem is the full potential code VGK that is distributed as part of the Engineering Sciences Data Unit (ESDU) Transonic Aerodynamics pack.[13] It is a two-dimensional, viscous-coupled, finite difference code that solves the full potential equations, modified to approximate the Rankine–Hugoniot relations across shocks. It can solve for either a specified angle of attack or for a target lift. For typical airfoils, a target lift viscous-coupled solution takes a few seconds. If, however, a design shape exhibits strong shocks, VGK typically fails to converge, and, thus, such designs cannot be analyzed in this way. This leads to difficulties in searching for designs that are close to such shock behavior, which is often where the best designs lie when only drag reduction is considered. Structural concerns are dealt with in this case study by requiring the section depth at 25 and 65% of the chord from the leading edge to be at least 9.5 and 7.5% of the section chord, respectively. These restrictions are meant to model the need to accommodate front and rear spars of appropriate depths within the section designs.

The airfoils considered here are based on a family of NASA transonic designs[14] as parameterized by the efficient orthogonal basis function scheme developed by Robinson and Keane,[15] which both minimizes the number of design parameters and provides global control of geometry. This scheme uses six parameters to describe the section. Figure 4 shows typical airfoil sections produced using the scheme, and Fig. 5 shows part of the drag landscape being searched over. Note that the gaps in the data here represent designs where the

CFD solver fails to converge, whereas the ridges arise from changes to the discretization scheme used as the geometry varies. This kind of behavior is typical of CFD prediction processes. The ability of the kriging models to regress such data to model correctly the underlying trends is an important characteristic when dealing with such problems.

Figure 6 shows a typical optimization history arising from the DOE plus kriging approach already outlined, where updates are driven by the expected improvement criterion. Notice that in Fig. 6 evaluations 1–75 represent the initial $LP\tau$ generated DOE, of which 52 designs generate feasible results, whereas points 76–400 are the update sequence. The initial set of DOE generated points is analyzed in parallel to reduce design times, a key feature of RSM-based approaches. The best design achieved using kriging and updates has a drag coefficient of $Cd = 0.00844$, and this drag level is indicated in Fig. 6 by the dotted line. Also shown in Fig. 6 is the optimization history achieved when using the GA to search directly the drag landscape using 1000 evaluations in 20 generations of 50 designs, again analyzed in parallel, giving a best Cd value of 0.00850 and requiring 250% of the effort of the krig approach. Note that the results for the direct search are not joined together by a line in Fig. 6 to aid clarity, rather the best of generation values are joined instead.

During the searches, and particularly during the initial DOE/first random generation of the GA, a number of the designs violate the structural constraints and hence, although having lower nominal drag values, are rejected. Designs that fail to solve at all are given a very high drag value and are eliminated from the process and plots. It is clear that, to begin with, during the update process many infeasible designs are generated along with a good number that fail to solve. This is quite normal during this kind of improvement process

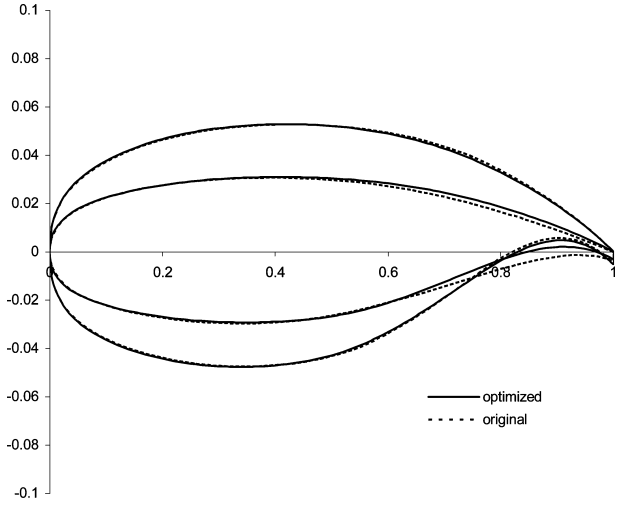


Fig. 4 Typical airfoil shapes produced using parameterization scheme of Robinson and Keane[15]; original shapes from NASA transonic set and optimized ones are variants produced by section optimizations of the sort considered here.

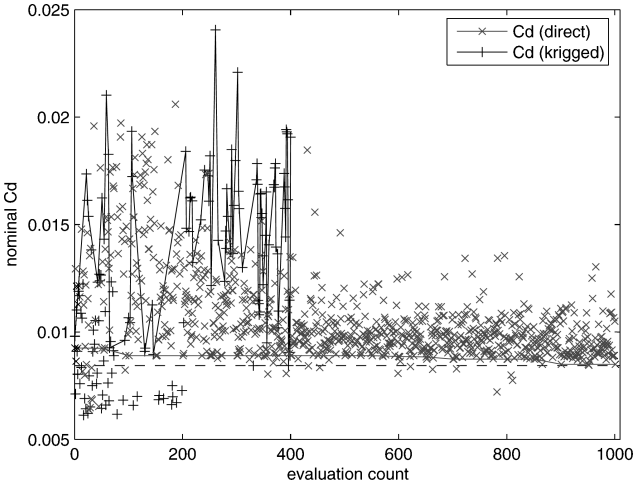


Fig. 6 Optimization histories for drag reduction using DOE and krig approach and for direct GA search (20 generations of 50 members).

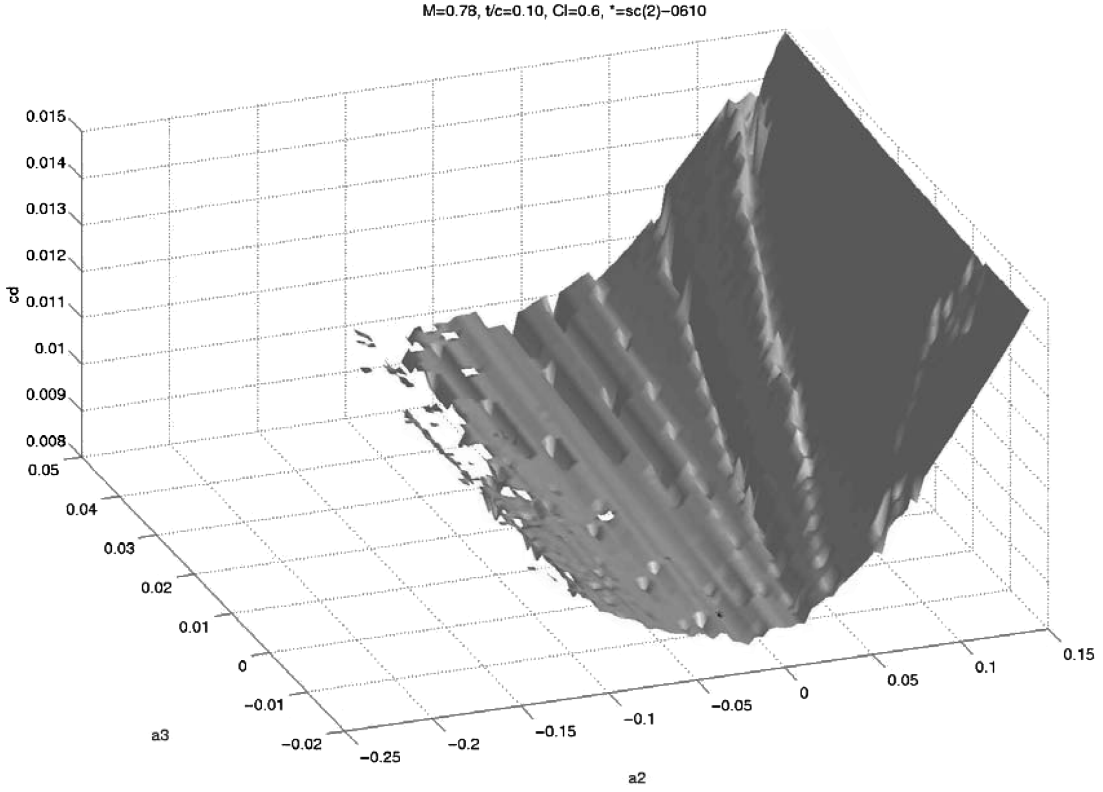


Fig. 5 Part of drag landscape produced using full potential code VGK.

because a considerable degree of exploration and response surface refinement is being undertaken. Then, after approximately evaluation 200, the krig process starts to settle down, and a steady improvement sequence is achieved until at iteration count 396 the best design found is identified. In all probability further searching would be able to improve on this, although no doubt only to a limited extent.

In this problem the number of designs that fail to solve rises as the designs reach lower and lower Cd values because the searches are then exploring regions where strong shocks readily occur. It is for this reason that the design of such sections must usually also consider the robustness issues that are addressed later in this paper, using multiobjective formulations.

Multiobjective Optimization

Most real design problems have more than one goal that the designer is trying to improve. In aerospace design it is common to be aiming for lightweight, low-cost, robust, high-performance systems. These aspirations are clearly in tension with each other and so compromise solutions have to be sought. Such compromises inevitably involve deciding on some form of weighting between the desired goals. However, before this stage is reached, it is possible to study design problems from the perspective of Pareto sets. A Pareto set of designs is one whose members are all optimal in some sense, but where the relative weighting between the competing goals is yet to be finally fixed; see, for example, Fonseca and Fleming.[16] More formally, a Pareto set of designs contains systems that are sufficiently optimized that, to improve the performance of any set member in any one goal function, its performance in at least one of the other functions must be made worse. Moreover, the designs in the set are said to be nondominated in that no other set member exceeds a given design's performance in all goals. It is customary to illustrate a Pareto set by plotting the performance of its members against each goal function (Fig. 7). The series of horizontal and vertical lines joining the set members is referred to as the Pareto front. Any design lying above and to the right of this line is dominated by members of the set. Note the distinction between the hatched region in Fig. 7 and the hatched and shaded regions. If a new design point is placed in the hatched and shaded regions this point will augment the Pareto set, whereas if it is placed below and to the left of these areas it will dominate and, thus, replace at least one member of the set.

There are a number of technical difficulties associated with constructing Pareto sets. First, the set members need to be optimal in some sense. Because it is desirable to have a good range of designs in the set, this means that an order of magnitude more optimization effort is usually required to produce such a set than to find a single design that is optimal against just one goal. Second, it is usually necessary to provide a wide and even coverage in the set in terms of the goal function space. Because the mapping between design parameters and goal functions is usually highly nonlinear, gaining such coverage is far from simple. Finally, and in common with single objective design, many problems of practical interest involve the use of expensive computer simulations to evaluate the performance of each candidate, and this means that only a limited number of such simulations can usually be afforded.

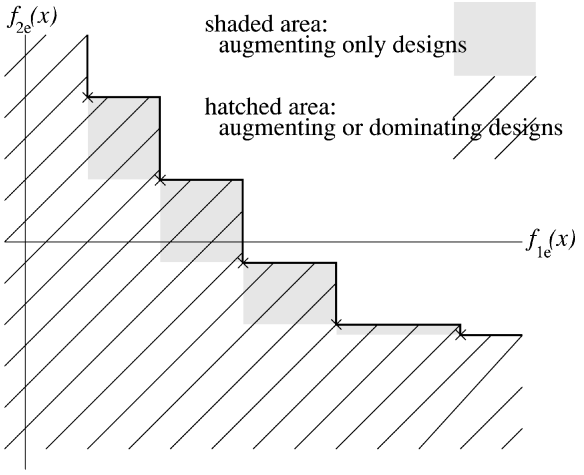


Fig. 7 Pareto set of five nondominated points for problem with two objectives: ——, Pareto front; hatched area, where improvements can be achieved; and shaded area, where improvements lead only to the set being augmented rather than existing members being dominated and replaced.

Currently, there appear to be two popular ways of constructing Pareto sets, for example, see Keane and Nair.[1] First, and most simply, one chooses a (possibly nonlinear) weighting function to combine all of the goals in the problem of interest into a single quantity and carries out a single-objective optimization. The weighting function is then changed and the process repeated. By slowly working through a range of weightings it is possible to build up a Pareto set of designs. This approach allows the full gamut of single-objective search methods to be applied, including the use of DOE and RSM technologies to speed up the search as per the earlier example. It does, however, suffer from a major drawback: It is by no means clear what weighting function to use and how to alter it to be able to reach all parts of the potential design space (and, thus, to have a wide ranging Pareto set). The nonlinear nature of most design problems will make it very difficult to ensure that the designs achieved are reasonably evenly spaced out through the design space. Moreover, if the Pareto front is concave, then highly nonlinear weighting functions must be used if all regions of the front are to be reached.

In an attempt to address this limitation, designers have turned to a second way of constructing Pareto sets via the use of population-based search schemes. In such schemes, a set of designs is worked on concurrently and evolved toward the final Pareto set in one process. In doing this, designs are compared to each other and progressed if they are of high quality and if they are widely spaced apart from other competing designs. Moreover, such schemes usually avoid the need for an explicit weighting function to combine the goals being studied. Perhaps the most well known of these schemes is the nondominated sorting GA (NSGA-ii) method introduced by Deb et al.[17] In this approach a GA is used to carry out the search, but the goal function used to drive the genetic process is based on the relative ranking and spacing of the designs in the set rather than their combined weighted performance. More specifically, at each generation all of the designs are compared and the nondominated designs set to one side. These are assigned rank one. The remaining designs are compared, and those that now dominate are assigned rank two and so on. Thus, the whole population is sorted into rank order based on dominance. This sorting into rank order dominance can be carried out irrespective of the relative importance of the objectives being dealt with or the relative magnitudes and scaling of these quantities.

Once the population of designs is sorted into ranks, they are next rewarded or penalized depending on how close they are to each other in goal space (and sometimes also in design variable space). This provides pressure to cause the search to fan out and explore the whole design space, but does require that the competing objectives be suitably scaled, an important issue that arises in many aspects of dealing with multiobjective approaches to design. When combined with the traditional GA operators of selection, crossover, and mutation, the NSGA-ii scheme is remarkably successful in evolving high-quality Pareto sets. As originally described, however, no means were provided for mitigating run-time issues arising from using expensive computer simulations in assessing competing designs. Moreover, the genetic operators used were somewhat simplistic. More sophisticated GA search engines are now commonly invoked, as is the case here.

To overcome the problem of long run times, a number of workers have advocated the use of RSM approaches, including kriging within Pareto front frameworks.[18,19] It is also possible to combine tools such as NSGA-ii with kriging (see Voutchkov et al.[20]) In such schemes an initial DOE is carried out and RSMs built as per the single-objective case, but now there is one RSM for each goal function. In the NSGA-ii approach the search is simply applied to the resulting RSMs and used to produce a Pareto set of designs. These

designs are then used to form an update set and, after running full computations, the RSMs are refined and the approach continued. Although sometimes quite successful, this approach suffers from an inability to balance explicitly exploration and exploitation in the RSM construction in the same way as when using such models in a single-objective search and "greedily" seeking the best designs, although the crowding or niching measures normally used help mitigate these problems to some extent. The central thrust of this paper is to construct statistically based operators for use in an RSM-based multiobjective search to tackle this problem explicitly.

Statistical Metrics for Multiobjective Search

When it is assumed that we have an expensive multiobjective search problem that is being tackled using a combined DOE and kriging approach, it is possible to revisit the ideas of improvement and devise appropriate metrics. To begin with, consider a problem with two expensive goal functions $f_{e1}(\boldsymbol{x})$ and $f_{e2}(\boldsymbol{x})$ with outputs (responses) y_1 and y_2 that must both be minimized. Moreover, for simplicity, assume that $\boldsymbol{x}$ consists of just one design variable x. When a DOE is constructed as before, it will be possible to construct a set of training data. This will allow us to identify the initial Pareto set of M_0 designs that dominate all of the others in the training set:

$$\boldsymbol{f}^*_{1,2} = \left\{\left[f^{(1)*}_{1e}\left(x^{(1)}\right), f^{(1)*}_{2e}\left(x^{(1)}\right)\right], \left[f^{(2)*}_{1e}\left(x^{(2)}\right), f^{(2)*}_{2e}\left(x^{(2)}\right)\right], \ldots, \left[f^{(M_0)*}_{1e}\left(x^{(M_0)}\right), f^{(M_0)*}_{2e}\left(x^{(M_0)}\right)\right]\right\}$$

In this set, the asterisk indicates that the designs are nondominated. We may plot these results on the Pareto front axes as per Fig. 7, discussed in the preceding section. In Fig. 7 the solid line is the Pareto front, and the hatched area represents locations where new designs would need to lie if they are to become members of the Pareto set. Recall that if new designs lie in the hatched shaded area they augment the set and that if they lie in the unshaded hatched area they will replace at least one member of the set (because they will then dominate some members of the old set). It is possible to set up our new metric such that an improvement is achieved if we can augment the set or, alternatively, only if we can dominate at least one set member. In this work we consider both metrics and also a combined form.

Given the training set, it is also possible to build a pair of krig metamodels. Here it is assumed that these models are independent, though it is also possible to build correlated models by adopting the formalism known as co-kriging.[21] The pair of metamodels $\hat{y}_1(x)$ and $\hat{y}_2(x)$ will be Gaussian processes as before, and each term in the two models will be identified by suffixes 1 and 2, respectively.

Given a proposed new design point x, this pair of models will provide a prediction of the two goal function values and also their standard errors, here taken to be uncorrelated. These values may then be used to construct a two dimensional Gaussian PDF for the predicted responses of the form

$$\phi(\hat{y}_1, \hat{y}_2) = \frac{1}{s_1(x)\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{[\hat{y}_1 - \mu_1(x)]^2}{s_1^2(x)}\right\} \times \frac{1}{s_2(x)\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{[\hat{y}_2 - \mu_2(x)]^2}{s_2^2(x)}\right\} \tag{12}$$

where it is made explicitly clear that s_1, μ_1, s_2, and μ_2 are all functions of the location at which an estimate is being sought. Clearly, this joint PDF accords with the predicted mean and errors coming from the two krig models at x. When seeking to add a new point to the training data, we wish to know the likelihood that any newly calculated point will be good enough to become a member of the current Pareto set and, when comparing competing potential designs, which will improve the Pareto set most.

We begin by first considering the probability that a new design will dominate a single member of the existing Pareto set, for example, $[f^{(i)*}_{1e}(x^{(i)}), f^{(i)*}_{2e}(x^{(i)})]$. For a two-dimensional problem, this may arise in one of three ways: Either the new point improves over the existing set member in goal one, or in goal two, or in both

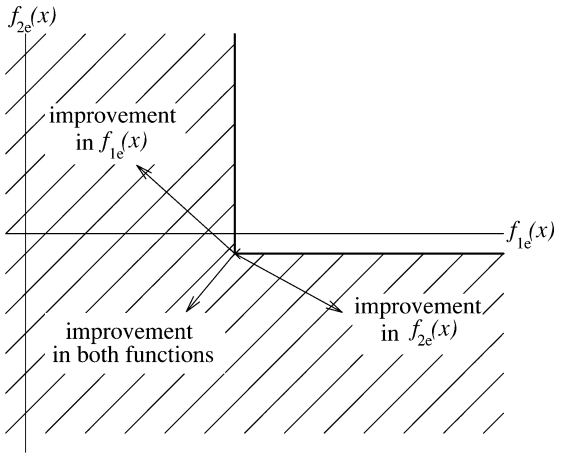


Fig. 8 Improvements possible from single point in Pareto set.

(Fig. 8). It will be obvious that the probability of the new design being an improvement is simply $P[\hat{y}_1(x^{(N_0+1)}) \le f^{(i)*}_{1e}(x^{(i)}) \cup \hat{y}_2(x^{(N_0+1)}) \le f^{(i)*}_{2e}(x^{(i)})]$, which is given by integrating the volume under the joint PDF to get

$$\Phi\left[\frac{f^{(i)*}_{1e}\left(x^{(i)}\right) - \hat{y}_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] + \Phi\left[\frac{f^{(i)*}_{2e}\left(x^{(i)}\right) - \hat{y}_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right] - \Phi\left[\frac{f^{(i)*}_{1e}\left(x^{(i)}\right) - \hat{y}_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\Phi\left[\frac{f^{(i)*}_{2e}\left(x^{(i)}\right) - \hat{y}_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right]$$

that is, by integrating over the hatched area in Fig. 8.

Next, consider the probability that the new point is an improvement given all of the points in the Pareto set. Now we must integrate over the hatched area in Fig. 7. (The distinction made there concerning whether or not the new point augments the existing set or dominates at least one set member will be perhaps clearer. By changing the area over which the integration takes place, we can assess the probability of either a new point dominating a set member or merely augmenting the existing set.) Carrying out the desired integral is best done by considering the various rectangles that comprise the hatched area in Fig. 7, and this gives

$$\begin{aligned} P\left[\hat{y}_1\left(x^{(N_0+1)}\right) \le \boldsymbol{f}^*_{1e}(x) \cup \hat{y}_2\left(x^{(N_0+1)}\right) \le \boldsymbol{f}^*_{2e}(x)\right]_{\text{aug}} &= \int_{-\infty}^{f^{*(1)}_{1e}} \int_{-\infty}^{\infty} \phi(\hat{y}_2, \hat{y}_1)\, \mathrm{d}\hat{y}_2\, \mathrm{d}\hat{y}_1 \\ &+ \sum_{i=1}^{M_0-1} \int_{f^{(i)}_{1e}}^{f^{*(i+1)}_{1e}} \int_{-\infty}^{f^{*(i)}_{2e}} \phi(\hat{y}_2, \hat{y}_1)\, \mathrm{d}\hat{y}_2\, \mathrm{d}\hat{y}_1 \\ &+ \int_{f^{*(M_0)}_{1e}}^{\infty} \int_{-\infty}^{f^{*(M_0)}_{2e}} \phi(\hat{y}_2, \hat{y}_1)\, \mathrm{d}\hat{y}_2\, \mathrm{d}\hat{y}_1 \end{aligned} \tag{13}$$

or

$$\begin{aligned} P[I]_{\text{aug}} = \Phi\left[\frac{f^{*(1)}_{1e}(x) - \mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] &+ \sum_{i=1}^{M_0} \left\{\Phi\left[\frac{f^{*(i+1)}_{1e}(x) - \mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] - \Phi\left[\frac{f^{*(i)}_{1e}(x) - \mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\}\Phi\left[\frac{f^{*(i)}_{2e}(x) - \mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right] \\ &+ \left\{1 - \Phi\left[\frac{f^{*(M_0)}_{1e}(x) - \mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\} \times \Phi\left[\frac{f^{*(M_0)}_{2e}(x) - \mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right] \end{aligned} \tag{14}$$

If instead of augmenting the existing Pareto set it is required that any new point dominates at least one existing set member, we get (noting the changed superscript on $f_{2e}^{*\,(i+1)}$)

$$
\begin{aligned}
P[I]_{\text{dom}} = \Phi&\left[\frac{f_{1e}^{*\,(1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\\
&+\sum_{i=1}^{M_0}\left\{\Phi\left[\frac{f_{1e}^{*\,(i+1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right.\\
&\left.-\Phi\left[\frac{f_{1e}^{*\,(i)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\}\\
&\times\Phi\left[\frac{f_{2e}^{*\,(i+1)}(x)-\mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right]\\
&+\left\{1-\Phi\left[\frac{f_{1e}^{*\,(M_0)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\}\\
&\times\Phi\left[\frac{f_{2e}^{*\,(M_0)}(x)-\mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right] \qquad (15)
\end{aligned}
$$

It is also possible to use the average of these two measures, which in some sense represents an attempt to model a Pareto front that runs between the two measures and, thus, more nearly accords with the likely shape of the real front in the problem being studied. These metrics are, of course, nondimensional and are the multiobjective equivalents of Eq. (10). Moreover, they work irrespective of the relative scaling of the objectives being dealt with. When used as search goals, they do not, however, necessarily encourage very wide ranging exploration because they are not biased by the degree of improvement being achieved. To do this, we must consider the first moment of the integral, as before when dealing with single-objective problems.

The equivalent improvement metric we require for the two-objective case will be the first moment of the joint PDF integral taken over the area where improvements occur, calculated about the current Pareto front. Now, although it is simple to understand the region over which the integral is to be taken [the same as in Eq. (13) and once again set either to augment or dominate existing set members], the moment arm about the current Pareto front is a less obvious concept. To understand what is involved it is useful to consider a geometrical interpretation of $E[I]$. $P[I]$ represents integration over the PDF in the area below and to the left of the Pareto front where improvements can occur. $E[I]$ is the first moment of the integral over this area, about the Pareto front. Now the distance the centroid of the $E[I]$ integral lies from the front is simply $E[I]$ divided by $P[I]$ (Fig. 9). Given this position and $P[I]$ it is simple to calculate $E[I]$ based on any location along the front. Hence, we first calculate $P[I]$ and the location of the centroid of its integral, $(\bar{\hat{y}}_1, \bar{\hat{y}}_2)$ (by integration with respect to the origin and division by $P[I]$). It is then possible to establish the Euclidean distance the centroid lies from each member of the Pareto set D^0. The expected improvement measure used in updates is subsequently calculated using the set member closest to the centroid, $[f_{1e}^*(x^*), f_{2e}^*(x^*)]$, by taking the product of the volume under the PDF with the Euclidean distance between this member and the centroid, shown by the arrow in Fig. 9. This leads to the following definition of $E[I]$:

$$
E[I] = P[I]\sqrt{\left[\bar{\hat{y}}_1\left(x^{(N_0+1)}\right)-f_{1e}^*(x^*)\right]^2+\left[\bar{\hat{y}}_2\left(x^{(N_0+1)}\right)-f_{2e}^*(x^*)\right]^2} \qquad (16)
$$

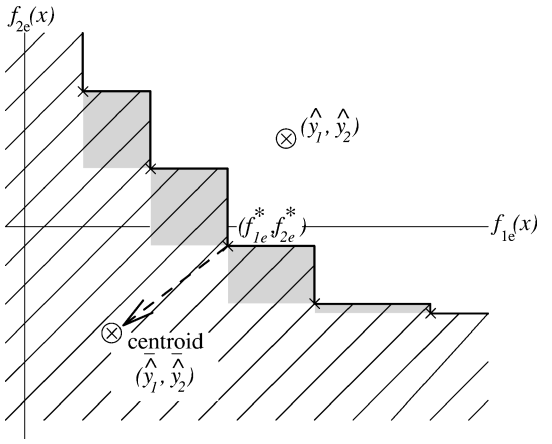


Fig. 9 Centroid of probability integral and moment arm used in calculating $E[I]$, also showing predicted position of currently postulated update $(\hat{y}_1, \hat{y}_2)$.

where when solutions that augment the front are permissible:

$$
\begin{aligned}
\bar{\hat{y}}_{1\text{aug}}\left(x^{(N_0+1)}\right) = &\left\{\int_{-\infty}^{f_{1e}^{*\,(1)}}\int_{-\infty}^{\infty}\hat{y}_1\phi(\hat{y}_1,\hat{y}_2)\,\mathrm{d}\hat{y}_2\,\mathrm{d}\hat{y}_1\right.\\
&+\sum_{i=1}^{M_0-1}\int_{f_{1e}^{(i)}}^{f_{1e}^{*\,(i+1)}}\int_{-\infty}^{f_{2e}^{*\,(i)}}\hat{y}_1\phi(\hat{y}_1,\hat{y}_2)\,\mathrm{d}\hat{y}_2\,\mathrm{d}\hat{y}_1\\
&\left.+\int_{f_{1e}^{*\,(M_0)}}^{\infty}\int_{-\infty}^{f_{2e}^{*\,(M_0)}}\hat{y}_1\phi(\hat{y}_1,\hat{y}_2)\,\mathrm{d}\hat{y}_2\,\mathrm{d}\hat{y}_1\right\}\Big/P[I] \qquad (17)
\end{aligned}
$$

and $\bar{\hat{y}}_{2\text{aug}}(x^{(N_0+1)})$ is defined similarly. The integrals of Eq. (17) are somewhat tedious but may be carried out by parts to yield

$$
\begin{aligned}
\bar{\hat{y}}_{1\text{aug}}\left(x^{(N_0+1)}\right) = &\left\{\mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right.\\
&-s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\\
&+\sum_{i=1}^{M_0-1}\left(\left\{\mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(i+1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right.\right.\\
&\left.-s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(i+1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\}\\
&-\left\{\mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(i)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right.\\
&\left.\left.-s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(i)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\}\right)\\
&\times\Phi\left[\frac{f_{2e}^{*(i)}(x)-\mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right]\\
&+\left\{\mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(M_0)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right.\\
&\left.+s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(M_0)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\}\\
&\times\Phi\left[\frac{f_{2e}^{*(M_0)}(x)-\mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right]\Bigg\}\Big/P[I] \qquad (18)
\end{aligned}
$$

If dominating solutions are required, this result becomes instead (noting again the changed superscript on $f_{2e}^{*(i+1)}$)

$$\bar{\hat{y}}_{1\mathrm{dom}}\left(x^{(N_0+1)}\right) = \left\{ \mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] - s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] + \sum_{i=1}^{M_0-1}\left(\left\{\mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(i+1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] - s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(i+1)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\} - \left\{\mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(i)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] - s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(i)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\}\right) \times \Phi\left[\frac{f_{2e}^{*(i+1)}(x)-\mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right] + \left\{\mu_1\left(x^{(N_0+1)}\right)\Phi\left[\frac{f_{1e}^{*(M_0)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right] + s\left(x^{(N_0+1)}\right)\phi\left[\frac{f_{1e}^{*(M_0)}(x)-\mu_1\left(x^{(N_0+1)}\right)}{s_1\left(x^{(N_0+1)}\right)}\right]\right\} \times \Phi\left[\frac{f_{2e}^{*(M_0)}(x)-\mu_2\left(x^{(N_0+1)}\right)}{s_2\left(x^{(N_0+1)}\right)}\right]\Bigg\Big/ P[I] \tag{19}$$

When defined in these ways, $E[I]$ varies with the location of the predicted position of the currently postulated update $(\hat{y}_1, \hat{y}_2) \equiv (\mu_1, \mu_2)$, also shown in Fig. 9 and also with the estimated errors in this prediction, s_1 and s_2, because it is these quantities that define the PDF being integrated.

The further the predicted update location lies below and to the left of the current Pareto front, the further the centroid will lie from the front. Moreover, the further the prediction lies in this direction, the closer the integral becomes to unity because the greater the probability of the update offering an improvement. Both tendencies will drive updates to be improved with regard to the design objectives. Note that if there is a significant gap in the points forming the existing Pareto front then centroid positions lying in or near such a gap will score proportionately higher values of $E[I]$ because the Euclidean distances to the nearest point will then be greater. This pressure will tend to encourage an even spacing in the front as it is updated. Also, when the data points used to construct the krigs, that is, all points available and not just those in the Pareto set, are widely spaced the error terms will be larger and this tends to further increase exploration. Thus, this multiobjective $E[I]$ definition balances exploration and exploitation in the same way as its one-dimensional equivalent in Eq. (11). The choice between using the augmenting formulation or the dominating one will depend on the designer's aims and also the nature of the problem under study, as will be shown later. Once more, it is also possible to use an average of the two metrics to try to approximate the likely final shape of the Pareto front.

It will also be clear that when calculating the location of the centroid there is still no requirement to scale the objectives being studied, but that when deciding which member of the current Pareto set lies closest to the centroid, relative scaling will be important, that is, when calculating the Euclidean distance. This is an unavoidable and difficult issue that arises whenever explicitly attempting to space out points along the Pareto front, whatever method is used to do this. Another subtlety arises when the centroid lies in the shaded regions in Fig. 9, that is, an augmentation of the existing Pareto set is acceptable. In such circumstances the Euclidean separation from the closest Pareto set member may, in fact, be greater than for a point lying outside the shaded region. This will cause greater pressure to be exerted on filling in the Pareto front than in migrating it to better regions of the design space. This is a direct consequence of accepting augmentation when defining the areas being integrated over. If this is a concern, then it is best to remove the shaded areas from the integrals by using the dominating formulation, which attempts to ensure that new points dominate at least one member of the existing Pareto set.

Before moving on to study examples making use of these metrics, note that there is no fundamental difficulty in extending this form of analysis to problems with more than two goal functions. This does, of course, increase the dimensionality of the Pareto surfaces being dealt with and so inevitably complicates further the expressions needed to calculate the improvement metrics. Nonetheless, they always remain expressible in closed form, it always being possible to define the metrics in terms of summations over known integrable functions.

Now that the necessary theoretical background has been given, the multiobjective $P[I]$ and $E[I]$ measures defined for two-function problems are next used and compared to surrogate assisted and direct NSGA-ii approaches to develop Pareto fronts for two example problems: first a simple structural design problem that may be expressed in closed form and then a modified version of the earlier airfoil design problem.

Multiobjective Example 1

The first multiobjective problem considered here is a variant of the classic Norwacki beam problem.[22] In this problem, the aim is to design a tip-loaded encastre cantilever beam for a minimum cross-sectional area and lowest bending stress subject to a number of constraints (Fig. 10 and Table 1). Specifically, the aim is to design a minimum-cost, low-stress beam carrying a tip load F of 5 kN at a fixed span l of 1.5 m. The beam is taken to be rectangular in section, breadth b, height h, and cross-sectional area A, and subject to the following design criteria: 1) a maximum tip deflection, $\delta = Fl^3/3EI_y$, of 5 mm, where $I_y = bh^3/12$; 2) a maximum allowable direct (bending) stress, $\sigma_B = 6Fl/bh^2$, equal to the yield stress of the material, σ_Y; 3) a maximum allowable shear stress, $\tau = 3F/2bh$, equal to one-half the yield stress of the material; 4) a maximum height to breadth ratio, h/b, for the cross-section of 10; and 5) the failure force for twist buckling, $F_{\mathrm{crit}} = (4/l^2)\sqrt{[GI_TEI_Z/(1-\upsilon^2)]}$, to be greater than the tip force multiplied by a safety factor f of two, where $I_T = b^3h + bh^3/12$ and $I_Z = b^3h/12$.

The material used is mild steel with a yield stress of 240 MPa, Young modulus E of 216.62 GPa, bulk modulus G of 86.65 GPa, and Poisson ratio υ of 0.27. Here, the trial vector describing the design consists of the cross-sectional breadth and height. Notice that it is not clear from the preceding specifications which of the design limits will control the design, although clearly at least one will if the beam is to have a nonzero cross-sectional area.

The design space for this problem can be readily mapped by a systematic variation of both design variables (Fig. 11). In Fig. 11, the stepped nature of the constraint boundaries arise from the finite

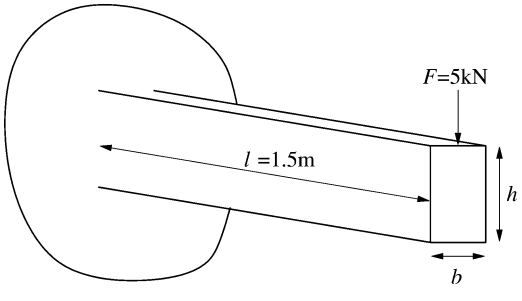
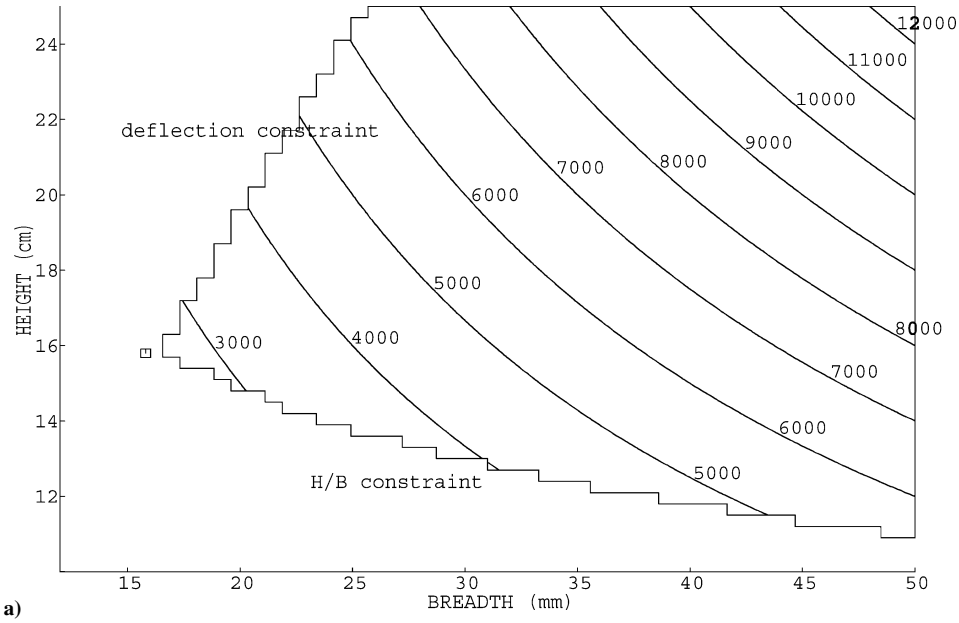


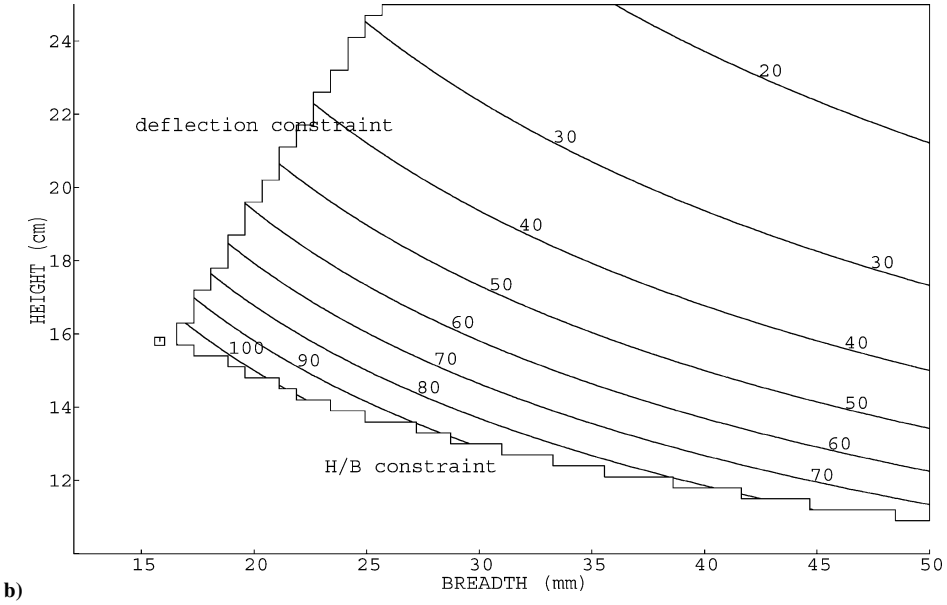
Fig. 10 Norwacki beam problem.

Table 1 Beam design variables

Variable	Value	Units	Type	Meaning
A	result	mm^2	Objective	Cross-sectional area of beam
σ_B	$<\sigma_Y$	MPa	Objective and constraint	Maximum bending stress
b	5.0 < free < 50.0	mm	Variable	Breadth of beam
h	2.0 < free < 25.0	cm	Variable	Height of beam
l	1,500.0	mm	Constant	Length of beam
F	5,000.0	N	Constant	Force on tip of beam
E	216,620.0	MPa	Constant	Young's modulus
G	86,650.0	MPa	Constant	Modulus of rigidity
υ	0.27		Constant	Poisson's ratio
σ_Y	200.0	MPa	Constant	Yield stress
f	2.0		Constant	Safety factor on critical force for twist buckling
τ	$<\sigma_Y/2$	MPa	Constraint	Maximum shear stress
δ	<5.0	mm	Constraint	Tip deflection
h/b	<10.0		Constraint	Height-to-breadth ratio
F_{crit}	>5,000.0	N	Constraint	Critical force for twist buckling



a)



b)

Fig. 11 Maps of Norwacki beam problem[22] objective functions vs breadth and height: a) cross-sectional area in millimeters squared and b) bending stress in megapascals.

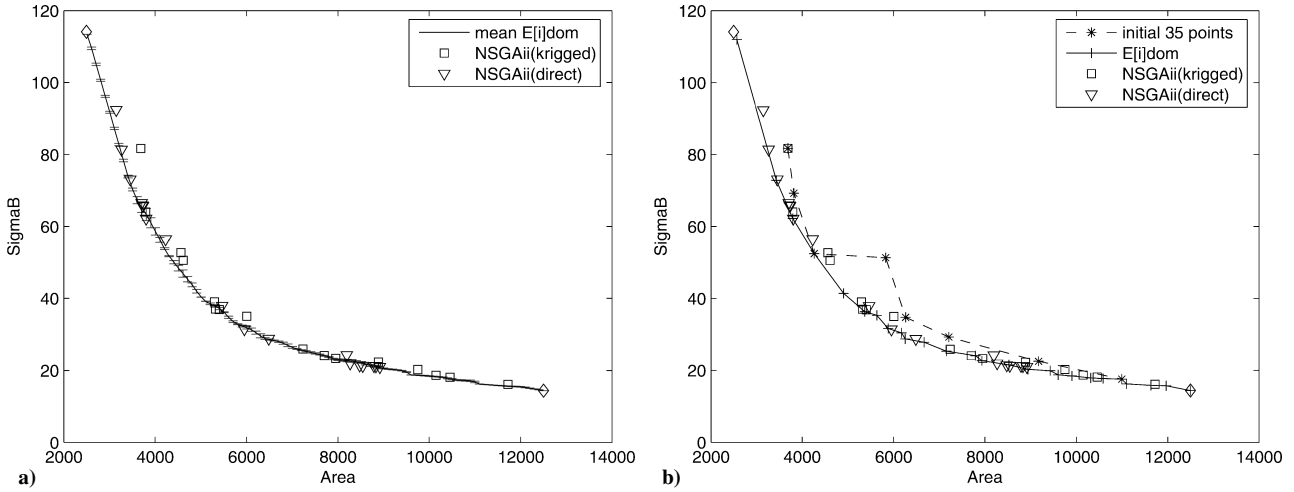


Fig. 12 Pareto front resulting from $E[I]_{\text{dom}}$ search on Norwacki beam problem[22] along with direct and krig-based NSGA-ii searches and front from initial DOE: a) average results with errors bars of ± 1 standard deviation and b) one example.

sample size used to map the functions; also the square marker indicates the minimum cross-sectional area end of the Pareto front. Note that because only feasible points are used to produce the contours, the actual feasible region will always be bigger than that contoured. Here this will lead to lower optimal values for both variables.

It will be clear from this problem description that there will be a tradeoff possible between minimum bending stress and minimum cross-sectional area solutions. The constraints on the tip deflection and height-to-breadth ratio mark one end of the resulting Pareto front, whereas setting both variables to their maximum values defines the other end of the front. Locating the minimum stress solution requires an exploration of the constraint boundaries, a characteristic common in engineering problems, but one that some search methods find difficult to deal with.

Figure 12 shows the results of searching this front using the $E[I]_{\text{dom}}$ metric already defined by Eqs. (16) and (19), in conjunction with the update process of Figs. 1 and 3 and two krigs, one each for the cross-sectional area and the bending stress. The constraints are applied directly to the various equations detailed earlier, although these, too, could, of course, be modeled using krigs and limits applied to these metamodels. In this example the initial DOE contains 35 points taken from a Latin hypercube design, and the krigs are tuned using the combined GA/DHC search with the expected improvement being searched with the same scheme to provide 100 updates. In this case the krigs are tuned after each update and the points are added one at a time because these never fail to return meaningful results.

Here Fig. 12a shows the results of averaging over five independent runs with errors bars of ± 1 standard deviation, whereas Fig. 12b shows a typical example result along with the front from the initial sample of 35 points. Also shown in Figs. 12 are the results of running a direct NSGA-ii search for 1000 evaluations using the same GA engine and also a further NSGA-ii search applied instead to the krig models and updated in the same way as for the $E[I]$ metric, again using 35 initial points and 100 updates (Ref. 20). Note that the endpoints for the Pareto front are known for this problem and occur for designs with breadths and heights of 15.8 mm and 15.8 cm and 50 mm and 25 cm, leading to endpoints in the objective front space of 2496.4 mm^2 and 114.1 MPa and 12,500 mm^2 and 14.4 MPa. These are shown as diamonds in Fig. 12.

It is clear from Fig. 12 that good-quality Pareto fronts can be established using $E[I]_{\text{dom}}$. In fact, all of the improvement metrics introduced here are generally better than those achieved by simply applying NSGA-ii to the krig models directly, using the same number of function evaluations and update strategies. The fronts are also competitive with those coming from direct application of NSGA-ii to the problem code but use eight times fewer function evaluations.

Notice also that both the direct and krig-based variants of NSGA-ii fail to identify the true ends of the front, whereas all runs of the $E[I]$ metric identify these locations to within rather tight bounds. This is because searches based on improvement metrics are able to make use of gradient based optimizers to identify directly these locations because these metrics reduce the multiobjective problem to a single measure of merit. This is particularly important for the (2496.4 mm^2, 114.1 MPa) point because this location is defined by the intersection of two constraint boundaries and these have to be searched accurately to reveal the front end. To achieve similar quality coverage, a direct NSGA-ii search requires on the order of 5000 function evaluations.

Multiobjective Example 2

The second multiobjective problem solved here is a variant on the airfoil section design problem considered earlier. Here, in addition to the requirement for low drag designs, robustness to slight variations in geometry is also considered. Specifically, the standard deviation of drag with respect to $\pm 5\%$ variations in the design parameters taken over a finite sample of 20 random design perturbations is considered as the second goal function. This is the design problem solved by Keane and Nair[1] in their second case study, using an extended direct search on the aerodynamic code with more than 50,000 function evaluations. The results produced here are compared directly with that work.

As already noted, designs that are very strongly optimized for low drag are increasingly sensitive to geometry perturbations, and so these two goals are again in tension and lead to a Pareto front. Furthermore, this design problem is made more difficult by geometry variants causing the VGK CFD solver to fail if they exhibit strong shocks. One of the purposes of studying robustness in this way is to make due allowance for such behavior because CFD analysis is never perfectly precise and also as-manufactured shapes always differ to some extent from the nominal design.

To study this multiobjective problem the same initial $LP\tau$ DOE of 75 points is used as in the single-objective study considered earlier and then updated with 375 further function evaluations in 25 batches of 15 points. Now, however, krigs are built for both the drag coefficient and its standard deviation. The multipoint batch update approach, using the averaged statistical metrics introduced earlier, gives rise to the Pareto fronts shown by the solid lines in Fig. 13. Also shown in Figs. 13 are the results of direct and krig-based NSGA-ii searches and those taken from Keane and Nair,[1] here labeled the extended direct GA search data. Table 2 provides a summary of the results obtained. Note that in the last column of Table 2 the Pareto front produced by a search is compared to the combined set of all results coming from the other four searches put together. It is

Table 2 Summary of results for example 2

Search method	Total function evaluations	Cd for lowest drag design	Standard deviation in Cd for most robust design	Number of points in final Pareto front	Number of dominating points from other searches
Extended direct GA search	50,000	0.00878	0.0000743	6	30
NSGA-ii (direct)	1,000	0.00878	0.000166	2	105
NSGA-ii (krig)	450	0.00857	0.000123	3	26
$P[I]$	450	0.00820	0.0000136	9	8
$E[I]$	450	0.00810	0.0000156	12	28

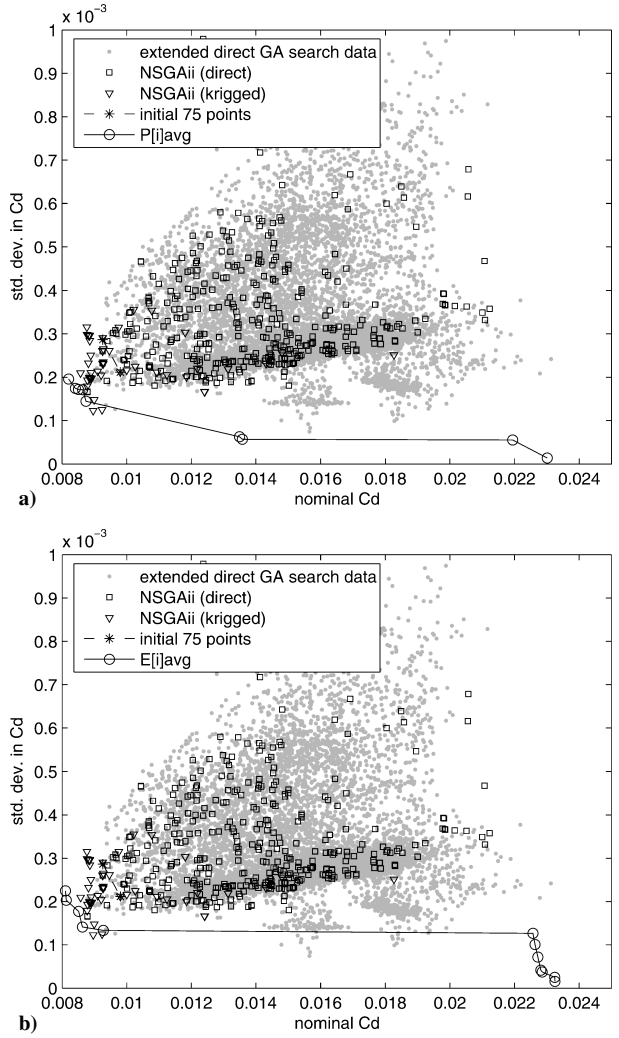


Fig. 13 Pareto fronts resulting from a) $P[I]_{\text{avg}}$ search and b) $E[I]_{\text{avg}}$ search on airfoil problem along with direct and krig-based NSGA-ii searches, front from initial DOE together with results from extended direct GA search, direct NSGA-ii search, and krig-based NSGA-ii searches.

clear that in this rather difficult problem it is not easy to establish a well-populated Pareto front:

1) The extended direct GA search shows the extent of the objective function space quite well and produces a Pareto front spanning a reasonable range of designs. However, it fails to find either very low drag or very robust designs. The results are competitive with the direct NSGA-ii search but at very high cost.

2) The direct application of NSGA-ii using 20 generations each of 50 evaluations also covers the objective function space fairly well but yields only two competitive design points (nearly identical and in the low-drag region. It completely fails to explore the space of robust designs with larger nominal Cd values and yields a very poor front of only two designs, even though it uses more than twice the number of function calls of the krig-based methods.

3) The krig-based NSGA-ii search performs much better and is significantly more efficient because it uses only the initial 75-point DOE followed by 25- and 15-point updates, that is, less than 50% of the effort of the direct search. Nonetheless, it too fails to find any designs at the low standard deviation end of the Pareto front, although it offers one more design in its front than the direct NSGA-ii search. Moreover, the three designs in the front are more broadly spread than those coming from the direct NSGA-ii search.

4) The $P[I]_{\text{avg}}$ based search gives well-optimized designs lying all along the Pareto front, dominating all but two of the designs from the extended 50,000 point direct search reported by Keane and Nair[1] and using only 1% of the computing effort required. The front contains designs with lower drag and lower standard deviation in drag than either of the NSGA-ii searches, giving the most robust design ever found for this problem. These designs are also, on average, more dominant than those coming from any of the other searches. Moreover, as has already been noted, this method does not require any scaling of problem terms in its formulation.

5) The search based on $E[I]_{\text{avg}}$ does slightly less well in this example than $P[I]_{\text{avg}}$ because the more aggressive nature of this metric leads to many more designs where VGK cannot solve for all of the geometry perturbations considered. This means that significantly less information is available to build the krig models used in searching for new updates. It does, however, still outperform the NSGA-ii searches and identifies the lowest drag design ever found for this problem, along with many designs at the robust end of the Pareto front. Note that the ability to deal with calculation failures is a key aspect of coping with wide-ranging CFD-based searches. This ability is often overlooked by those developing new search engines in isolation. In real engineering problems, such difficulties can arise from failures of the CAD engine to regenerate correct geometries, the meshing tool to mesh any geometries that do build, or the solver to converge the flowfield surrounding the design. In the author's experience, it is not uncommon for 30% of all solutions to fail to solve for one reason or another and, moreover, for these failures to occur in regions close to the most desirable solutions. By avoiding gradient or downhill searches applied to the actual functions being optimized, all of the methods used here overcome this difficulty.

Based on these results and those from the first example, it would appear that the precise choice of improvement metric is not crucial but should reflect the stability of the analysis tool, any difficulties in the relative scaling of design goals, and the balance desired between aggressively seeking the ends of the Pareto front as opposed to finding a more evenly spread set of designs that lie along it, $P[I]_{\text{aug}}$ being the least aggressive search and $E[I]_{\text{dom}}$ the most aggressive.

Conclusions

In this paper, three distinct methods for carrying out multiobjective design optimization are described: a direct optimization of the user's analysis codes using NSGA-ii, a search of krig-based response surfaces using NSGA-ii, and a search of a series of new, statistically based improvement metrics based on response surfaces. It is shown that the statistical tools built on top of krig, that is, Gaussian process, models offer enhanced performance in terms of both speed and quality of results. The searches are faster than direct application of NSGA-ii while providing more diversely populated Pareto fronts than krig-based searches with NSGA-ii. In particular, the probability of improvement metric $P[I]_{\text{avg}}$ introduced here seems to allow a powerful multiobjective search process to be formulated

that can make use of existing single-objective search tools, including gradient-based methods, while delivering good-quality, widely spaced Pareto sets, all set in a parallel-cluster-based computing environment. Importantly, this approach works without the need for any scaling between the goals being studied, which is in contrast to NSGA-ii and almost all other approaches for constructing Pareto fronts. For more aggressive searches, the expected improvement metric $E[I]_{\text{avg}}$ can be used, but this does require scaling between the goals being searched in a similar fashion to NSGA-ii, so that Euclidean measures may be taken in objective function space. It also proves to be more sensitive in coping with design codes that sometimes fail to converge, a not uncommon situation.

Further work on this topic will examine cokriging for problems with strongly correlated goals, problems with more than two goals, and also the use of the improvement metrics embedded within NSGA-ii searches, where they may be used to help rank new design variants before expensive function evaluations are committed to.

Acknowledgments

The use in this work of Ivan I. Voutchkov and Wenbin Song's implementation of the nondominated sorting genetic algorithm with kriging support is gratefully acknowledged, as is their help in producing results from their code.

References

[1]Keane, A. J., and Nair, P. B., *Computational Approaches for Aerospace Design*, Wiley, Chichester, England, U.K., 2005.

[2]Myers, R. H., and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Design of Experiments*, Wiley, Chichester, England, U.K., 1995.

[3]Mead, R., *The Design of Experiments*, Cambridge Univ. Press, Cambridge, England, U.K., 1988.

[4]McKay, M. D., Conover, W. J., and Beckman, R. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[5]Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[6]Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, 2001, pp. 345–383.

[7]Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimisation," *Structural Optimization*, Vol. 15, No. 1, 1998, pp. 16–23.

[8]Statnikov, R. B., and Matusov, J. B., *Multicriteria Optimization and Engineering*, Chapman and Hall, New York, 1995.

[9]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Boston, 1989.

[10]Yin, X., and Germay, N., "A Fast Genetic Algorithm with Sharing Scheme Using Cluster Methods in Multimodal Function Optimization," *Proceedings of the International Conference on Artificial Nueral Nets and Genetic Algorithms*, edited by R. F. Albrecht, C. R. Reeves, and N. C. Steele, Springer-Verlag, Heidelberg, Germany, 1993, pp. 450–457.

[11]Keane, A. J., *The OPTIONS Design Exploration System User Guide and Reference Manual*, URL: http://www.soton.ac.uk/∼ajk/options.ps [cited 1 Jan. 2001].

[12]Yuret, D., and de la Maza, M., "Dynamic Hill Climbing: Overcoming the Limitations of Optimization Techniques," *Proceedings of the 2nd Turkish Symposium on Artificial Intelligence and Neural Networks*, Bogazici Univ., Istanbul, Turkey, 1993, pp. 254–260.

[13]"VGK Method for Two-Dimensional Aerofoil Sections," Engineering Sciences Data Unit, TR 96028, IHS ESDU International, London, 1996.

[14]Harris, C. D., "NASA Supercritical Airfoils: A Matrix of Family-Related Airfoils," NASA TP 2969, March 1990.

[15]Robinson, G. M., and Keane, A. J., "Concise Orthogonal Representation of Supercritical Airfoils," *Journal of Aircraft*, Vol. 38, No. 3, 2001, pp. 580–583.

[16]Fonseca, C. M., and Fleming, P. J., "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computing*, Vol. 3, No. 1, 1995, pp. 1–16.

[17]Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-ii," *Lecture Notes in Computer Science*, Vol. 1917, 2000, pp. 848, 849.

[18]Wilson, B., Cappelleri, D., Simpson, W., and Frecker, M., "Efficient Pareto Frontier Exploration Using Surrogate Approximations," *Optimization and Engineering*, Vol. 2, 2001, pp. 31–50.

[19]Knowles, J. D., and Hughes, E. J., "Multiobjective Optimization on a Budget of 250 Evaluations," *Evolutionary Multi-Criterion Optimization, Third International Conference, EMO*, edited by C. A. Coello Coello, A. Hernandez Aguirre, and E. Zitzler, Lecture Notes in Computer Science, Vol. 3410, Springer-Verlag, Heidelberg, Germany, 2005, pp. 176–190.

[20]Voutchkov, I. I., Song, W., and Keane, A. J., "Further Developments of Multi-Objective Searches Using Evolutionary Methods and Response Surfaces: NSGA-iii," *AIAA Journal* (submitted for publication).

[21]Cressie, N. A. C., *Statistics for Spatial Data*, Wiley, Chichester, England, U.K., 1993.

[22]Nowacki, H., "Modelling of Design Decisions for CAD," *CAD Modeling, Systems Engineering, CAD-Systems*, Lecture Notes in Computer Science, No. 89, Springer-Verlag, Berlin, 1980.

A. Berman
Associate Editor