

Optimization with missing data

BY ALEXANDER I. J. FORRESTER*, ANDRÁS SÓBSTER
AND ANDY J. KEANE

*Computational Engineering and Design Group, School of Engineering Sciences,
University of Southampton, Southampton SO17 1BJ, UK*

Engineering optimization relies routinely on deterministic computer based design evaluations, typically comprising geometry creation, mesh generation and numerical simulation. Simple optimization routines tend to stall and require user intervention if a failure occurs at any of these stages. This motivated us to develop an optimization strategy based on surrogate modelling, which penalizes the likely failure regions of the design space without prior knowledge of their locations. A Gaussian process based design improvement expectation measure guides the search towards the feasible global optimum.

Keywords: global optimization; imputation; Kriging

1. Introduction

New areas of research or the early stages of a design process require a geometry model, with coupled analysis codes, which encompasses a wide range of possible designs. In such situations the analysis code is likely to fail to return results for some configurations due to a variety of reasons. Problems may occur at any stage of the design evaluation process: (i) the geometry described by a parameter combination may be infeasible, (ii) an automated mesh generator may not be able to cope with a particularly complex or corrupt geometry or (iii) the simulation may not converge, either due to mesh problems or the automated solution setup being unable to cope with certain unusual, yet geometrically feasible, configurations.

In an ideal world, a seamless parameterization would ensure that the optimizer could visit wide ranging geometries and move between them without interruption. In reality, however, the uniform, flawless coverage of the design space by a generic geometry model and simulation procedure is fraught with difficulties. In all but the most trivial or over-constrained cases the design evaluation can fail in certain areas of the search space. Of course, if these infeasible areas are rectangular, they can simply be avoided by adjusting the bounds of the relevant variables, but this is rarely the case. Most of the time such regions will have complex, irregular, and even disconnected shapes, and are much harder to identify and avoid. In the presence of such problems it is difficult to accomplish entirely automated optimization without narrowing the bounds of the problem to the extent that promising designs are likely to be excluded. This negates the possible benefits of

* Author for correspondence (alexander.forrester@soton.ac.uk).

global optimization routines, and the designer may well choose to revert to a more conservative local optimization around a known good design.

Although industry and academia direct considerable effort at robust geometry creation and mesh generation, it is unlikely that truly fault free automated design evaluation will be achieved. Rather than tackling problems with the individual design evaluation components, we address, or rather circumvent, the problem of failed design evaluations via the medium of surrogate based optimization methodology. This approach allows us to cope with regions of infeasible designs without knowing where they will occur. Although gradient-free search methods, for example a genetic algorithm (GA), *can* complete a global search of a domain with infeasible regions, the large number of direct calls to the computer code will render such methods impractical when using time consuming simulations, as shown when we compare the performance of our surrogate based methods with a GA in §4.

We begin in the next section by presenting the surrogate modelling based optimization approach we use in this paper. The following section discusses the problems associated with failed design evaluations and introduces the proposed solution methodology. We go on in §4 to demonstrate our method on an aerofoil design case study, before drawing conclusions in the final section.

2. Surrogate based optimization

This paper is centred around surrogate based optimization, and as such we begin by presenting the method of *Kriging*¹—an approximation method made popular due to its ability to model complex functions and provide error estimates. [Sacks et al. \(1989\)](#) introduced the use of Kriging (outside of its birthplace in geostatistics) as a means to approximate the output of computer experiments. Here, equations for the Kriging predictor and error are presented without derivation. The reader is directed to the above reference or [Jones et al. \(1998\)](#) for more information.

As with all surrogate based methods, we start with a set of sample data—usually computed at a set of points in the domain of interest determined by a design of experiment (DoE) procedure. The Kriging approximation is built from a mean base term, $\hat{\mu}$ (the circumflex denotes a maximum likelihood estimate (MLE)), plus n basis functions centred around the n sample points, $\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x} \in \mathbb{R}^k$:

$$\hat{y}(\mathbf{x}_{n+1}) = \hat{\mu} + \sum_{i=1}^n b_i \psi_i(\|\mathbf{x}_{n+1} - \mathbf{x}_i\|), \tag{2.1}$$

where the basis functions ψ_i are given by the column vector

$$\boldsymbol{\psi} = \begin{pmatrix} \text{Corr}[Y(\mathbf{x}_{n+1}), Y(\mathbf{x}_1)] \\ \vdots \\ \text{Corr}[Y(\mathbf{x}_{n+1}), Y(\mathbf{x}_n)] \end{pmatrix} = \begin{pmatrix} \exp \left[-\sum_{j=1}^k \hat{\theta}_j (\|x_{n+1,j} - x_{1,j}\|)^{\hat{p}_j} \right] \\ \vdots \\ \exp \left[-\sum_{j=1}^k \hat{\theta}_j (\|x_{n+1,j} - x_{n,j}\|)^{\hat{p}_j} \right] \end{pmatrix}, \tag{2.2}$$

¹Named after D. G. Krige—a South African mining engineer who developed the method in the 1950s for determining ore grade distributions based on core samples.

(the correlation between a random variable $Y(\mathbf{x})$ at the point to be predicted (\mathbf{x}_{n+1}) and at the sample data points ($\mathbf{x}_1, \dots, \mathbf{x}_n$)). The *hyper-parameter* p_j can be thought of as determining the smoothness of the function approximation. In geostatistical models, where Kriging originated, erratic local behaviour (the *nugget effect*) may benefit from the use of $\hat{p}_j \in [0,1]$ to allow for erratic responses, but here the modelling of engineering functions implies that there will not be any singularities and the use of $p_j=2$ means that the basis function is infinitely differentiable through a sample point (when $\|\mathbf{x}_{n+1} - \mathbf{x}_i\| = 0$). With $p_j=2$, the basis function has a Gaussian distribution with variance $1/\hat{\theta}_j$. $\hat{\theta}$, therefore, can be thought of as determining how quickly the function changes as \mathbf{x}_{n+1} moves away from \mathbf{x}_i , with high and low $\hat{\theta}_j$ s indicating an active or inactive function, respectively. It is usual practice to use a constant $\hat{\theta}_j$ for all dimensions in \mathbf{x} , but the use of variable $\hat{\theta}_j$ s gives a non-axisymmetric basis, allowing for varying impacts of each variable of the design space. In essence, the variance $1/\hat{\theta}_j$ is used to normalize the distance $\|x_{n+1,j} - x_{i,j}\|$ to give equal activity across each dimension Jones *et al.* (1998).

The constants b_i are given by the column vector $\mathbf{b} = \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})$, where \mathbf{R} is an $n \times n$ symmetric matrix of correlations between the sample data, \mathbf{y} is a column vector of the sample data, $(y(\mathbf{x}_1), \dots, y(\mathbf{x}_n))^T$, $\mathbf{1}$ is an $n \times 1$ column vector of ones, and the MLE prediction of $\hat{\mu} = \mathbf{1}^T \mathbf{R}^{-1} \mathbf{y} / \mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}$.

In addition to computing an initial set of experiments and fitting an approximation to the data, the surface is usually refined with additional data (*update* or *infill* points) to improve accuracy in the area of the optimum and confirm the objective function values predicted by the approximation. A natural way of refining the surface is to compute a new simulation at the predicted optimum. The approximation is then rebuilt and new optimum points are added until the predicted optimum agrees with the update simulation to a specified tolerance. The optimization may, however, become ‘trapped’ at local optima when searching multi-modal functions (see Jones (2001) for an excellent review of the pitfalls of various update criteria). An update strategy must allow for the prediction being just that—a prediction. The Kriging predictor, $\hat{y}(\mathbf{x}_{n+1})$, is chosen to maximize the likelihood of the combined test data and prediction (the augmented likelihood). It follows that predicted values are likely to be more accurate if the likelihood drops off sharply as the value of $\hat{y}(\mathbf{x}_{n+1})$ changes, i.e. alternative values are inconsistent with the initial dataset. Conversely, a gentle drop-off in the likelihood indicates that alternative values of $\hat{y}(\mathbf{x}_{n+1})$ are more likely to be possible. This notion leads to an intuitive method of deriving the estimated mean squared error in the predictor. The curvature of the augmented likelihood is found by taking the second derivative with respect to $\hat{y}(\mathbf{x}_{n+1})$, and it follows that the error in the predictor is related to the inverse of this curvature:

$$s^2(\mathbf{x}_{n+1}) = \hat{\sigma}^2 [1 - \boldsymbol{\psi}^T \mathbf{R}^{-1} \boldsymbol{\psi}], \tag{2.3}$$

where $\hat{\sigma}^2 = (\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}) / n$. The full, though rather less revealing derivation of s^2 can be found in Sacks *et al.* (1989).² Equation (2.3) also has

² An extra term in the error prediction of Sacks *et al.* (1989),

$$\hat{\sigma}^2 \frac{(1 - \mathbf{1}^T \mathbf{R}^{-1} \boldsymbol{\psi})^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \ll 1,$$

is attributed to the error in the estimate of $\hat{\mu}$, and is neglected here.

the intuitive property that the error is zero at a sample point, since if ψ is the i th column of \mathbf{R} , $\psi^T \mathbf{R}^{-1} \psi = \psi(\mathbf{x}_i - \mathbf{x}_i) = 1$.

Positioning updates based on the error alone (i.e. maximizing s^2) will, of course, lead to a completely global search, although the eventual location of a global optimum is guaranteed, since the sampling will be dense (Torn & Zilinskas 1987). Here, we employ an infill criterion which balances local exploitation of \hat{y} and global exploration using s^2 by maximizing the expectation of improving upon the current best solution.

The Kriging predictor is the realization of the random variable $Y(\mathbf{x})$, with a probability density function

$$\frac{1}{\sqrt{2\pi}s(\mathbf{x})} \exp -\frac{1}{2} \left(\frac{Y(\mathbf{x}) - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \right)^2,$$

with mean given by the predictor $\hat{y}(\mathbf{x})$ (equation (2.1)) and variance s^2 (equation (2.3)). The most plausible value at \mathbf{x} is $\hat{y}(\mathbf{x})$, with the probability decreasing as $Y(\mathbf{x})$ moves away from $\hat{y}(\mathbf{x})$. Since there is uncertainty in the value of $\hat{y}(\mathbf{x})$, we can calculate the expectation of its improvement, $I = f_{\min} - Y(\mathbf{x})$, on the best value calculated so far,

$$\begin{aligned} E[I(\mathbf{x})] &= \int_{-\infty}^{\infty} \max(f_{\min} - Y(\mathbf{x}), 0) \phi(Y(\mathbf{x})) dY, \\ &= \left\{ \begin{array}{ll} (f_{\min} - \hat{y}(\mathbf{x})) \Phi \left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \right) + s\phi \left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \right), & \text{if } s > 0, \\ 0, & \text{if } s = 0, \end{array} \right\} \end{aligned} \tag{2.4}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function and probability density function, respectively. $f_{\min} - \hat{y}(\mathbf{x})$ should be replaced by $\hat{y}(\mathbf{x}) - f_{\max}$ for a maximization problem, but in practice it is easier to take the negative of the data so that all problems can be treated as minimizations (we consider minimization for the remainder of this paper). Note that $E[I(\mathbf{x})] = 0$ when $s = 0$ so that there is no expectation of improvement at a point which has already been sampled and, therefore, no possibility of re-sampling, which is a necessary characteristic of an updating criterion when using deterministic computer experiments, and guarantees global convergence, since the sampling will be dense.

3. Imputing data for infeasible designs

Parallels can be drawn between the situation of encountering *infeasible* designs in optimization and *missing* data as it appears in statistical literature (see, e.g. Little & Rubin 1987). When performing statistical analysis with missing data, it must be ascertained whether the data is missing at random (MAR), and therefore ignorable, or whether there is some relationship between the data and its ‘missingness’. A surrogate model based on a DoE is, in essence, a missing data problem, where data in the gaps between the DoE points is MAR, due to the space filling nature of the DoE, and so is ignored when making predictions.

When, however, some of the DoE or infill points fail, it is likely that this missing data is not MAR and the missingness is in fact a function of \mathbf{x} .

Surrogate model infill processes may be performed after ignoring missing data in the DoE, whether it is MAR or otherwise. However, when an infill design evaluation fails, the process will fail: if no new data is added to the model, the infill criterion, be it based on \hat{y} , s^2 , $E[I(x)]$, or some other surrogate based criterion, remains unchanged and the process will stall. The process may be jump started by perturbing the prediction with the addition of a random infill point (a succession of random points may be required before a feasible design is found). However, ignoring this missing data may lead to distortions in the surrogate model, causing continued reselection of infeasible designs and the majority of the sampling may end up being based on random points. As such, because failed design evaluations are not MAR we should *impute* values to the missing data before training the model.

While the statistical literature deals with feasible missing data (data missing due to stochastic sampling) and so the value of the imputation is important, here we are faced with infeasible missing data—there can be no successful outcome to the deterministic sampling.³ Thus, the imputed data should serve only to divert the optimization towards the feasible region. The process of imputation alone, regardless of the values, to some extent serves this purpose: the presence of an imputed sample point reduces the estimated error (equation (2.3)), and hence $E[I(\mathbf{x})]$ at this point to zero, thus diverting further updates from this location. A value better than the optimum may still, however, draw the optimization towards the region of infeasibility. We now go on to consider which is the most suitable model by which to select imputation values.

Moving away from the area of feasible designs, $\|\mathbf{x}_{n+1} - \mathbf{x}_i\| \rightarrow \infty$, $\psi_i \rightarrow 0$, and so $\hat{y}(\mathbf{x}_{n+1}) \rightarrow \hat{\mu}$ (from equation (2.1)). Thus, predictions in infeasible areas will tend to be higher than the optimum region found so far. However, the rate at which the prediction returns to $\hat{\mu}$ is strongly linked to $\hat{\theta}$. Therefore, for functions of low modality, i.e. low $\hat{\theta}$, where there is a trend towards better designs on the edge of the feasible region, imputations based on \hat{y} may draw the update process towards the surrounding region of infeasibility. It seems, therefore, logical to take into account the expected error in the predictor to penalize the imputed points by using a statistical upper bound, $\hat{y} + s^2$. Now as $\|\mathbf{x}_{n+1} - \mathbf{x}_i\| \rightarrow \infty$, $\hat{y}(\mathbf{x}_{n+1}) + s^2(\mathbf{x}_{n+1}) \rightarrow \hat{\mu} + \hat{\sigma}^2$ (from equations (2.1) and (2.3)), while we still retain the necessary characteristic for guaranteed global convergence: as $\|\mathbf{x}_{n+1} - \mathbf{x}_i\| \rightarrow 0$, $\hat{y}(\mathbf{x}_{n+1}) + s^2(\mathbf{x}_{n+1}) \rightarrow y(\mathbf{x}_i)$, i.e. our imputation model interpolates the feasible data and so does not affect the asymptotic convergence of the maximum $E[I(\mathbf{x})]$ criterion in the feasible region.

4. Aerofoil design case study

We now demonstrate the penalized imputation method via its application to the optimization of the shape of an aerofoil. The aerofoil is defined by five orthogonal

³There may of course be a physical value to our design criterion at the location of the missing data (if the design is physically realizable), which might be obtained through a different solution setup, but here we are interested in automated processes with a generic design evaluation suited to the majority of viable configurations.

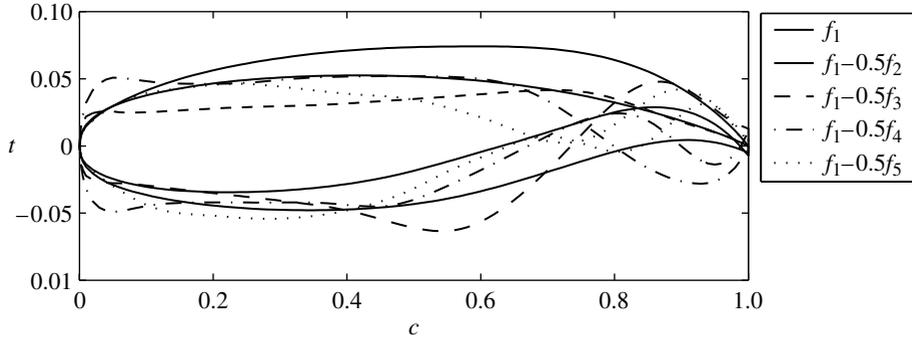


Figure 1. Orthogonal basis function aerofoil parameterization (axes not to scale).

shape functions (Robinson & Keane 2001) and a thickness to chord ratio t/c . The first function, which represents the shape of a NASA supercritical SC(2)-0610 aerofoil (Harris 1990), and t/c are kept constant (t/c is fixed at 10%, as defined by the last two digits of the four digit NACA code). The first function, f_1 is in fact a smooth least-squares fitting to the coordinates of the SC(2) series of aerofoils. Each of the four subsequent functions, f_2, \dots, f_5 is a least-squares fitting to the residuals of the preceding fit, and can be added to f_1 with a greater or lesser weighting, w_i , than is required to give an accurate approximation of the SC(2)-0610 aerofoil. Figure 1 shows the effect of adding each function with a weighting of $w_i = -0.5$ to f_1 . Nonsensical aerofoils are produced by adding individual functions with such large weightings, but a high degree of geometry control is achieved by combining the functions and optimizing their weightings.

(a) Two variables

To ease visualization of the results, we limit ourselves initially to a two variable optimization of weightings, w_2 and $w_3 \in [-0.5, 0.5]$, with $w_4, w_5 = 0$. The drag coefficient (C_D) of the resulting aerofoil is to be minimized at $C_L = 0.6$, $M_\infty = 0.73$ and $Re = 3 \times 10^6$. Flow simulations are performed using VGK (short for ‘viscous Garabedian and Korn’), a full potential code with an empirical drag correction (Freestone 1996). With each VGK calculation taking only a few seconds, it is possible to build a detailed map of the objective function, which is to be optimized. This ‘true’ response surface, built from 41×41 simulations, is shown in figure 2 as a translucent colour surface. 35% of the 1681 simulations were successful and only these C_D values are depicted in figure 2. The region of feasibility is generally well defined, with large regions of failed simulations⁴ for lower values of w_2 and w_3 . However, the pattern of failure is less obvious for high values of w_2 and w_3 , and the optimum is close to the edge of feasibility, making this a difficult problem to solve (with no prior knowledge of the bounds of feasibility).

Figure 2 also shows a typical outcome of applying the penalized imputation scheme to the aerofoil optimization problem. We start with a 20 point optimal Latin hypercube DoE (Morris & Mitchell 1995), of which seven are feasible and are shown as black dots in figure 2. A Kriging model is fitted to the feasible data (shown as a coarse black mesh). We then add s^2 to the Kriging prediction to

⁴ Failures are associated with strong shock waves and regions of boundary layer separation.

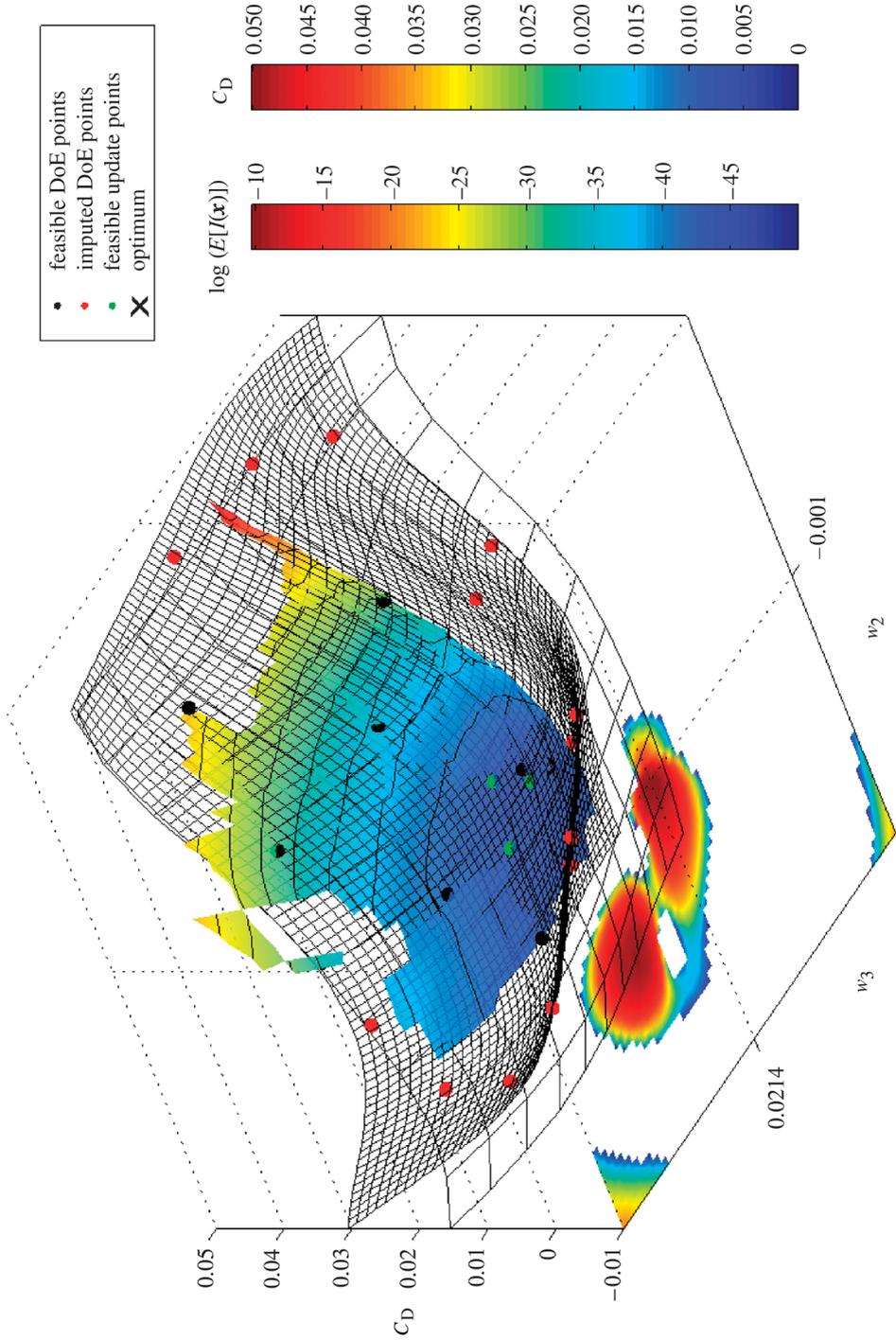


Figure 2. Two variable true function (translucent colour surface with contours), prediction based on feasible points (coarse black mesh), prediction through feasible and penalized imputed data (fine black mesh), and $\log E[I(\mathbf{x})] > -50$ (2D colourmap).

Table 1. Performance comparison averaged over ten runs of the two variable aerofoil optimization.

	failures in initial DoE/population (%)	subsequent failures (%)	total simulations (including failures)
random update	64.0	65.0	106.4
predictor imputation	64.0	22.0	49.9
penalized imputation	64.0	2.0	23.6
GA (population=10)	60.4	12.1	76.8
random search	65.3 (of all designs)		102.6

calculate the penalized imputations at the failed DoE locations (red dots). A Kriging prediction through the feasible and imputed points (fine black mesh) is used to calculate $E[I(\mathbf{x})]$. This expected improvement (shown as $\log E[I(\mathbf{x})]$ by the colourmap at the bottom of the figure, with values below -50 omitted to avoid cluttering the figure) is maximized to find the position of the next update point (updates are shown as green dots). The approximation is augmented with this point and the process continues. The situation in [figure 2](#) is after three updates (that required to get within one drag count of the optimum C_D of 73 counts—we know this value from the 41×41 evaluations used to produce the true function plot), with $E[I(\mathbf{x})]$ as it was after the second update, i.e. that which was maximized to find the third update location, the position of which is depicted on the w_2 and w_3 axes.

It is seen from the figure that the method has successfully identified the region of optimal feasible designs, with no updates having been positioned in an infeasible region. The prediction upon which $E[I(\mathbf{x})]$ is based (the fine black mesh) accurately predicts, based on feasible data, the true function for the majority of the feasible design space and deviates upwards in a smooth and continuous manner in infeasible regions, based on the penalized imputations. Without penalization the prediction (coarse black mesh) remains low in the area of infeasibility close to the optimum and would thus draw wasted updates towards this region.

The penalized imputation method is compared with the two previously mentioned and discarded surrogate based methods: (i) ignoring the missing data and updating at random points until a successful simulation is performed, before continuing the maximum $E[I(\mathbf{x})]$ infill strategy and (ii) imputing values based on \hat{y} . We also optimize the aerofoil using a GA⁵ and a straightforward random search. Each optimization is performed ten times from different initial samples, and [table 1](#) shows the average performance of each method when run until a design is found with C_D within one drag count of the optimum.

As expected, the rate of failure for the random and space filling samples in the first column of [table 1](#) are similar to that of the 41×41 true dataset. The second

⁵The GA performance figures included in this comparative study have been measured over multiple runs of a canonical implementation of the algorithm, where we have used a tournament-type selection mechanism to regulate the fitness-based survival of binary coded individuals. Random designs are generated until there are sufficient successful evaluations to create the initial population. The standard crossover and mutation operators have been modified to cope with offspring fitness evaluation failures by simply allowing the parent(s) to survive in such cases.

column of update failures rates are more interesting. When missing data is ignored a large number of updates are based on random sampling and thus the failure rate is similar to the values in column one. When imputation is used the failures drop dramatically, particularly when penalization forces the updates away from the infeasible region—just two per cent of designs failed here, corresponding to one design out of all ten optimization runs. With the initial population of the GA concentrated within the feasible region, subsequent failures are low here as well. As far as overall performance is concerned, ignoring the missing data proves to be no better than a random search. Although the GA deals well with the missing data, such a search is more suited to finding optimal *regions* in higher dimensional or multi-modal problems and its final convergence to an accurate optimum is slow. The imputation based methods outperform the other optimizers, although the prediction imputation still wastes resources sampling in infeasible regions, with penalized imputation proving to be over 50% faster.

(b) *Four variables*

The problem is now extended to four variables, with $w_2, \dots, w_5 \in [-0.25, 0.25]$. The region of feasibility can be visualized via the hierarchical axis plot in figure 3. The plot is built from 11^4 runs of the VGK code.⁶ Each tile shows C_D for varying w_2 and w_3 for a fixed w_4 and w_5 . w_4 and w_5 vary from tile to tile with the value at the lower left corner of each tile representing the value for the entire tile.

18% of the design space proved to be feasible, for which colour maps of the C_D are shown, with the colour scale the same as in figure 2. A large proportion of the design space is infeasible due to the simple reason of the upper and lower surfaces of the aerofoil intersecting at mid chord. Regions in which this occurs are shown in dark blue and the simplicity of the region means that it could be eliminated by adjusting the bounds of the variables. The blank areas in figure 3, however, indicate regions for which VGK has failed for designs where we cannot easily identify the geometry as being corrupt, e.g. extensive rippling of the aerofoil surface produces problems with the convergence of flow simulations. Also, beyond the problem of identifying corrupt geometries, a sound geometry may fail due to its particular flow characteristics, e.g. the onset of stall before the C_L criterion is met. These factors combine to produce a complex domain of feasibility, which is awkward to identify. While efforts can be made to reduce the bounds of the problem to this feasible domain of interest, we cannot eliminate design evaluation failures completely without removing feasible designs and so compromising the global nature of the search.

The outcome of a run of the penalized imputation method using an initial optimal Latin hypercube DoE of 200 points followed by 23 updates (again to attain a C_D within one drag count of the optimum—now 70 counts) is shown in figure 3, with points depicted using the same legend as figure 2 (we display the best run in order to produce a clearer figure with few points). Only discrete values of w_4 and w_5 can be displayed and as such the design variables are rounded to the nearest 0.05 for the purpose of visualization. It should be borne in mind that there is, in fact, a distribution of designs between tiles. The reader should also appreciate that, although the major axis is much larger than the axes of the

⁶Generating such plots would, of course, not be possible in most cases due to the high computational cost. Here, we have computed this large quantity of data for illustrative purposes.

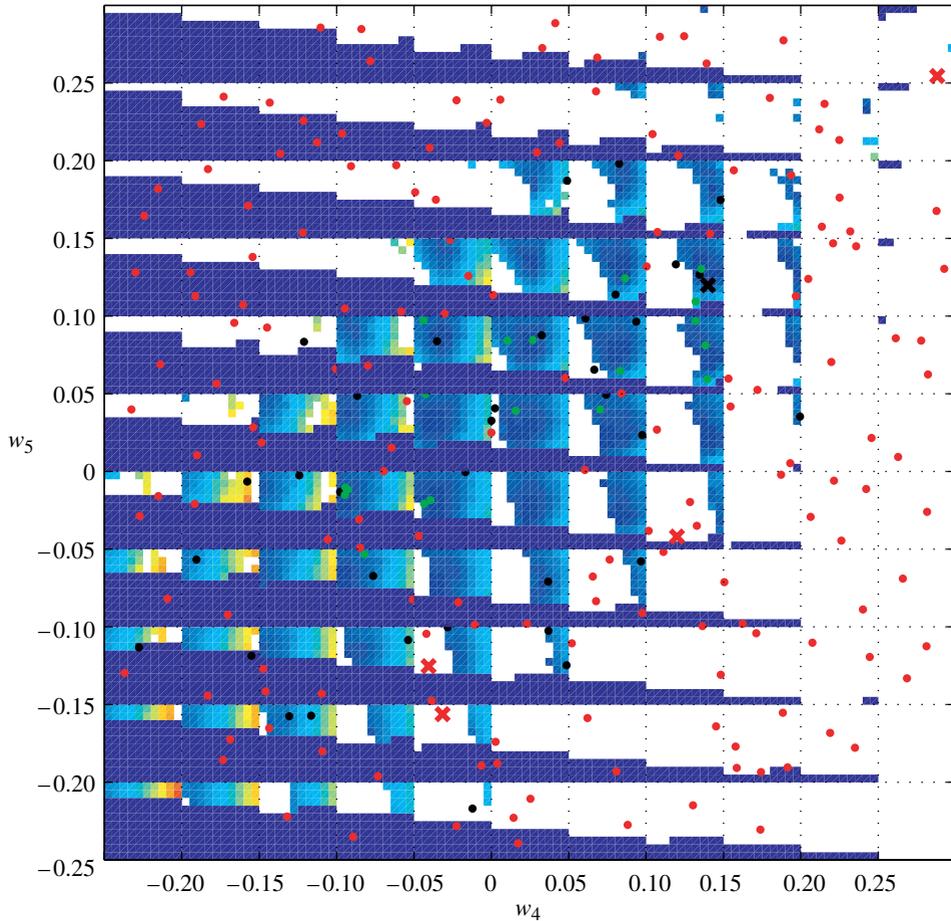


Figure 3. Four variable true function and penalized imputation method data points (see figure 2 for legend). Red crosses indicate imputed update points. Regions of infeasible geometries are shown as dark blue.

individual tiles, they both represent the same variation across a variable. Thus, although there seems to be a wide distribution of update points across w_4 and w_5 , these points represent a tight cluster in the region of optimal feasible designs. The initial approximation after the DoE was based on just 35 successful points, with the remaining 165 being imputed. Despite this small successful sample, it is seen that the feasible updates are clustered in optimal regions with only four of the updates having been positioned outside of the feasible region. Again we make comparisons, averaged over five runs, with four other optimization methods in table 2.

Table 2 follows a similar pattern to table 1. We note here that the number of function evaluations is relatively high for a four variable problem, due to the extremely large area of infeasibility coupled with the complexity of the noisy C_D landscape. Ignoring missing data has little impact on the failure rate, while it is significantly reduced by using imputations. The GA performs best in these terms, but, as before, requires a large number of evaluations to reach the optimum. As well

Table 2. Performance comparison averaged over ten runs of the four variable aerofoil optimization.

	failures in initial DoE/population (%)	subsequent failures (%)	total simulations (including failures)
random update	81.7	77.5	774.6
predictor imputation	81.7	40.8	best = 421
penalized imputation	81.7	36.2	308.8
GA	80.6	21.8	874.4
random search	80.0 (of all designs)		2316.8

as our $C_D \leq 0.007$ stopping criterion, we limit the number of points used to build a surrogate model to 500.⁷ Although when ignoring the missing data the size of the Kriging model remains small, the total number of simulations is high. However, imputing values to these missing data increased the size of the model, leading to the termination of the prediction-based imputations for three out of the five runs prior to attaining the optimum C_D and thus an averaged value is not available. The benefits of penalized imputations outweigh the increase in model size and, as in the two variable example, this method proves to be the most successful.

5. Conclusions

The use of imputation in regions of failure enables surrogate model based global optimization of unfamiliar design spaces. Such methods can provide significant time savings over direct global search methods, such as GAs. The studies performed show promising results for the use of imputations penalized with a stochastic process based statistical upper bound.

References

- Freestone, M. M. 1996 VGK method for two-dimensional aerofoil sections. Technical Report 96028, Engineering Sciences Data Unit, November 1996.
- Harris, C. D. 1990 NASA supercritical airfoils—a matrix of family-related airfoils. NASA Technical Paper (2969), March 1990.
- Jones, D. R. 2001 A taxonomy of global optimization methods based on response surfaces. *J. Global Optim.* **21**, 345–383. (doi:10.1023/A:1012771025575)
- Jones, D. R., Schlonlau, M. & Welch, W. J. 1998 Efficient global optimisation of expensive black-box functions. *J. Global Optim.* **13**, 455–492. (doi:10.1023/A:1008306431147)
- Little, R. J. A. & Rubin, D. B. 1987 *Statistical analysis with missing data*. New York: Wiley.
- Morris, M. D. & Mitchell, T. J. 1995 Exploratory designs for computer experiments. *J. Stat. Plan. Inference* **43**, 381–402. (doi:10.1016/0378-3758(94)00035-T)
- Robinson, G. M. & Keane, A. J. 2001 Concise orthogonal representation of supercritical aerofoils. *J. Aircraft* **38**, 580–583.
- Sacks, J., Welch, W. J., Mitchell, T. J. & Wynn, H. 1989 Design and analysis of computer experiments. *Stat. Sci.* **4**, 409–423.
- Torn, A. & Zilinskas, A. 1987 *Global optimization*. Berlin: Springer.

⁷The size of the training sample limits the applicability of the technique. In our experience training the Kriging model is prohibitively expensive beyond 500 points and/or 20 dimensions.