Journal of Global Optimization **26:** 297–319, 2003.

A Knowledge-Based Approach To Response Surface Modelling in Multifidelity Optimization

STEPHEN J. LEARY, ATUL BHASKAR and ANDY J. KEANE
Computational Engineering and Design Centre, School of Engineering Sciences, The University of Southampton, Highfield, Southampton, SO17 1BJ, UK

Abstract. This paper is concerned with approximations for expensive function evaluation – the expensive functions arising in an engineering design context. The problem of reducing the computational cost of generating sufficient learning samples is addressed. Several approaches of using *a priori* knowledge to achieve computational economy are presented. In all these, the results of a cheap model are treated as knowledge to be incorporated in the training process. Several approaches are described here: in particular, we focus on neural based systems. This approach is then developed as a new knowledge-based kriging model which is shown to be as accurate as neural based alternatives while being much easier to train. Examples from the domain of structural optimization are given to demonstrate the approach.

Key words: Multifidelity modelling, Knowledge-based neural networks, Kriging, Expensive function optimization.

1. Introduction

The problem of optimization using high cost models is common to many engineering design problems. The design optimization of an engineering system typically requires many analyses of that system. Due to this high computational cost, approximate models for the system are sought. The approximations may be global or local. Global approximations try to capture the behaviour of the objective function and/or constraints over the entire domain of interest. Local models are defined in a specific region of the design space. The work described here focuses on global approximations; nevertheless the methods used could be implemented locally.

Perhaps the most common way of tackling the problem of expensive function optimization is through the use of approximations to the expensive model. Response surface methods (see for example Myers and Montgomery, 1995) seek polynomial approximations to the function. These models, once constructed, provide a cheap means of approximating the original expensive function/model. An approach based on kriging is described in Jones et al. (1998). Their algorithm builds a global approximation using a kriging model and then performs optimization using this model. Other approximation strategies include the use of neural networks. All these methods are, in effect, forms of curve fitting. They are built using data points from the high cost model and do not attempt to incorporate any further information on the problem in hand.

One concern with these approaches is the level of accuracy of the resulting approximation arising from the inevitably limited quantities of training data used. As a result, there has been a growing interest in the use of simple lower fidelity models to overcome this problem by sampling the parameter space at points that are not sampled for expensive function evaluation. These low fidelity models, while being less accurate than the original model, are generally much cheaper to compute. As an example, in a finite element analysis the cheap model may use a coarser mesh than the original expensive model, while during a CFD analysis a panel code may replace an expensive Euler analysis. Combining the low fidelity models with the more accurate but expensive high fidelity models can provide a good combination of high accuracy and low cost.

The low fidelity model may be included in the approximation in many ways, we present several of these approaches and make comparison.

Perhaps the simplest way of utilizing low fidelity information is to consider the differences between the two models. Watson and Gupta (1996) implement this idea using a neural network to model differences between the two models and apply the approach to microwave circuit design. The technique uses a design of experiments (DOE) methodology to identify configurations of the input variables for which to run the high fidelity model. The low fidelity model is then run at these design points providing information on the differences between the two models. An approximation to the high fidelity model can then be constructed using the low fidelity model and an approximation to the difference.

An alternative to this approach is to model the ratio of the high and low fidelity models. For example, Haftka (1991) and Chang et al. (1993) calculate the ratio and derivatives at one point in order to provide a linear approximation to the ratio at other points in the design space. The approach is applied to a wing-box model of a high speed civil transport aircraft. More recently the approach has been applied using polynomial models to approximate the ratio. The approach, termed a "correction response surface" model, has been applied to aerodynamic drag approximation by Hutchinson et al. (1994) as well as structural problems (for example, see Vitali et al., 1999).

A third approach termed "space-mapping" aims to establish a relationship or mapping between the input space of the low fidelity model and that of the high fidelity model such that the low fidelity model with the mapped parameter accurately reflects the behaviour of the high fidelity model. Both linear and non-linear mappings have been considered in the literature (see, e.g., Bandler et al., 1999; Bakr et al., 2000a, 2000b; Bandler et al., 2001) for details. This technique has been used extensively in microwave circuit design.

Recently Wang and Zhang (1997) developed a knowledge-based neural network model for microwave design. This approach includes problem specific knowledge in the form of generic empirical functions inside the neural network. We extend this approach here by incorporating low fidelity information inside the network. This has particularly important implications when an empirical function representing knowledge (as in Wang and Zhang) is unavailable.

In this paper, a knowledge-based kriging model is developed. This approach enjoys similar levels of performance to the knowledge-based neural network approach, but is much simpler to train. This is an extension of the classical kriging approach; here the approximation includes information from a weighted low fidelity model. The weights in this model are determined along with the kriging hyperparameters by maximizing the likelihood.

The rest of the paper is set out as follows. Section 2 briefly reviews the use of multifidelity correction models using neural networks, Section 3 considers a new application of knowledge-based neural networks in the context of multifidelity modelling. A knowledge-based kriging model is developed in Section 4. Structural examples are given in Section 5 that demonstrate some of the capabilities of the various approaches. Finally, conclusions are drawn in Section 6.

2. Multifidelity Modelling Using Artificial Neural Networks

An artificial neural network, (see, for example, White et al., 1992), consists of a set of simple processing units which communicate by sending signals to each other over a large number of weighted connections. The network is trained using training data obtained from selective calls to the high fidelity model. The trained model can then be used as a surrogate to the original expensive code. However, when training data is limited due to the prohibitive cost of generating sufficient learning samples, then such approximations can be inadequate. The use of multifidelity models is becoming increasingly popular in overcoming such problems.

We briefly discuss one such approach and then consider a new application of knowledge-based neural networks, the empirical functions used in Wang and Zhang (1997) being replaced by a low fidelity model.

Although most applications of neural networks are in function approximation, they have also been successfully applied to multifidelity modelling, (Watson and Gupta, 1996). Here the idea is to approximate a function f_e which is expensive to compute. As a result, very few training data are available. The idea is to improve the approximation given by a cheap function f_a which approximates f_e and is less costly to compute but lacks accuracy. This cheaper function contains useful information about the behaviour of f_e in regions where f_e is not sampled.

The difference between the two models

$$d = f_e - f_a \tag{1}$$

is considered. This is sampled at various locations $\mathbf{x}_i,\ i = 1, 2, ..., N$ and provides training data $(\mathbf{x}_i, d_i),\ i = 1, 2, ..., N$ which are used to train the neural network. After training, the network provides a cheaper approximation $\hat{d}$ to d throughout the

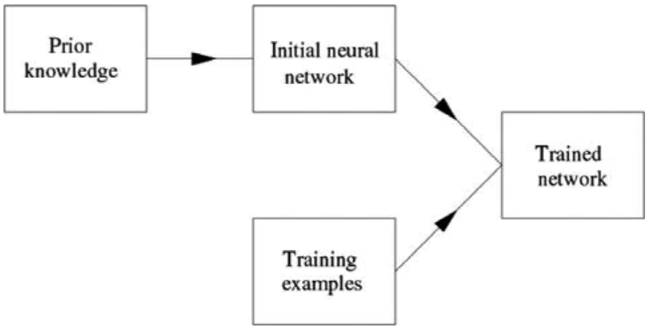

Figure 1. Schematic diagram of the KBNN approach of Towell and Shavlik (Towell and Shavlik, 1994).

whole domain. As a result,

$$f_a + \hat{d} \approx f_e \tag{2}$$

can be used as a surrogate repetitively at little cost. This is obviously useful when we wish to optimize the expensive model: we can optimize $f_a + \hat{d}$ instead of f_e.

Another approach, as highlighted earlier, is to model the ratio $r = f_e/f_a$ and then consider $\hat{r} f_a$ as a surrogate for f_e.

We now turn our attention to another neural network approach which, we observe, has applications to multifidelity modelling, the knowledge-based neural network (KBNN).

3. Multifidelity Modelling using Neural Networks by Incorporating Low Fidelity Data as Knowledge

Typically, neural networks, including the multilayer perceptron (MLP), learn by example. In function approximation, the network learns by training the outputs of the network to agree with a set of training data outputs. Another approach is to train the network using prior knowledge of the function we wish to approximate, provided this information is available.

The motivation behind knowledge-based neural networks (KBNN) is that people never really learn by example or theory alone. Our knowledge comes from both teaching and learning by example. So some hybrid approach seems sensible.

This is the idea in a knowledge-based neural network: prior knowledge is somehow built into the model. There are at least two possible approaches:

(1) As described in Towell and Shavlik (1994), the prior knowledge is used to define a network topology and the initial weights within this network. This is shown in Figure 1. The network is trained using the training examples.
(2) The alternative, Wang and Zhang (1997), is to incorporate this knowledge inside the network in the form of empirical functions (see Figure 2).

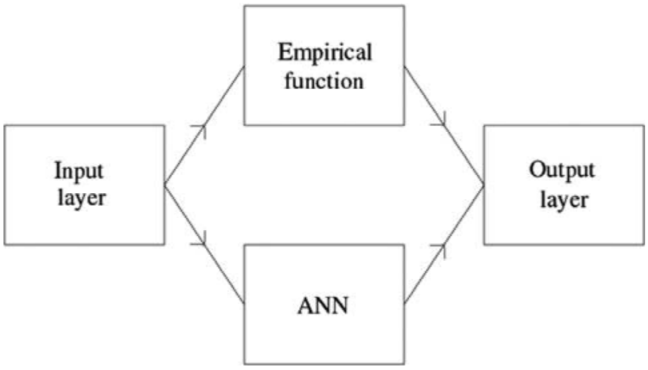

Figure 2. Schematic of the KBNN approach of Wang and Zhang (Wang and Zhang, 1997).

It is argued in Wang and Zhang that for functional problems, the latter approach is more suitable. The former, which typically uses symbolic information, is more suited to the problem of classification. As a result we will only be interested in the approach of Wang and Zhang here.

As an example, consider data coming from an exponentially decaying sinusoid with small random pertubation (Figure 3). Suppose that this is expensive to produce so we only sample at a few locations (six here, Figure 4) and use this to train the network.

A multilayer perceptron approximation is first calculated, this passes through the training data, but lacks accuracy away from it. Suppose we now introduce some empirical function knowledge, namely that the function behaves similarly to an exponentially decaying sinusoid. This knowledge need not be precise. For instance, here we assume nothing about the frequency or amplitude of the empirical function. These parameters are determined by the network during training. This empirical knowledge can be included in the network.

To compare the results, the test data are shown along with the multilayer perceptron approximation and the knowledge-based neural network approximation in Figure 5. As can be seen, incorporating knowledge in this way has considerably increased the accuracy of the approximation in the regions away from the points where the expensive function has been sampled.

The knowledge-based approach is particularly appealing here since it is readily applicable to multifidelity modelling. In this paper, we propose to use the information generated by the low fidelity model in place of the empirical knowledge: thus adapting the framework of Wang and Zhang (1997) in the context of multifidelity modelling (compare Figures 2 and 6). This low fidelity model *shares physics* with the high fidelity expensive model, but differs in details. Here the ‘empirical function’ which defines our prior knowledge comes from the cheap model f_a: this gives us some information as to the behaviour of the expensive model f_e away from expensive sampled points. If there is reasonable correlation between the models

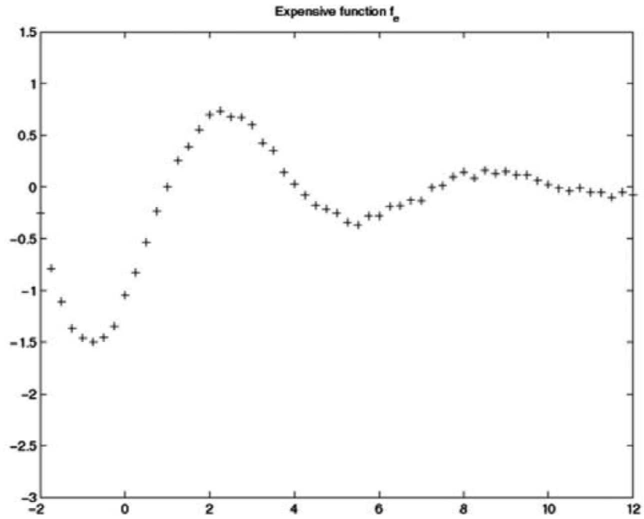

Figure 3. Test data. Noise is added symmetrically to a decaying sinusoid sampled at 57 points.

then this approach is likely to increase the accuracy of the prediction, particularly at extrapolated points.

3.1. MODEL STRUCTURE

With the modification to the approach of Wang and Zhang (1997) as suggested earlier in this paper, we consider a network with input layer **X**, knowledge layer **Z**, boundary layer **B**, region layer **R**, normalized region layer **R**′ and output layer **Y** (see Figure 7). The low fidelity model now appears in the knowledge layer. The outputs of the knowledge layer **Z** and neural layers **R**′ are weighted and merged by multiplication. In our experience this approach is seen to perform better than using a multilayer perceptron with a single hidden layer in a KBNN approach.

The input layer accepts inputs **x** from outside the model, details of the knowledge layer, boundary layer, region layer, normalized region layer and output layer follow in Equations (3)–(8).

In the following we consider the problem with input vector **x** $(N_x \times 1)$, output y (approximating the high fidelity model $f_e(\mathbf{x})$) and knowledge z (see Equation (3)). Both y and z could be vectors, but we will consider the case of a single output only.

As stated earlier, the ‘empirical knowledge’ is given by the cheap model. The input **x** is weighted so that the knowledge vector is calculated from the low fidelity model evaluation as

$$z = f_a(\mathbf{W}_1\mathbf{x} + \mathbf{w}_2), \tag{3}$$

where here $\mathbf{W}_1 = \mathrm{diag}\{w_1^1, w_1^2, ..., w_1^{N_x}\}$ is a diagonal matrix of weights for scaling and $\mathbf{w}_2$ is a vector of weights for a shift of the input arguments. This procedure can

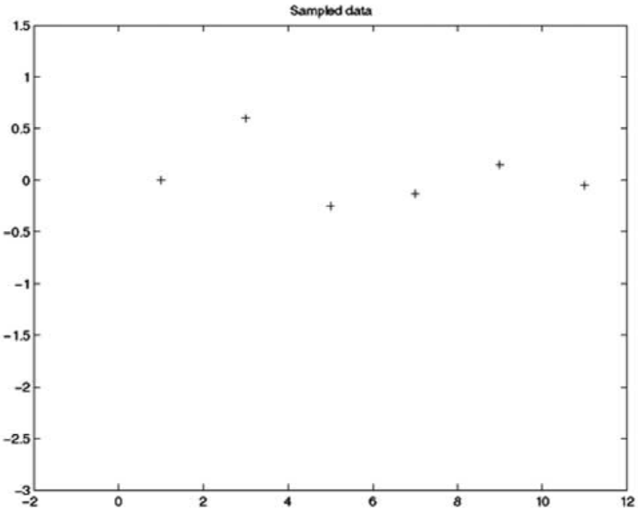

Figure 4. Training data. Six data points out of the data in Figure 3 are used for training the network.

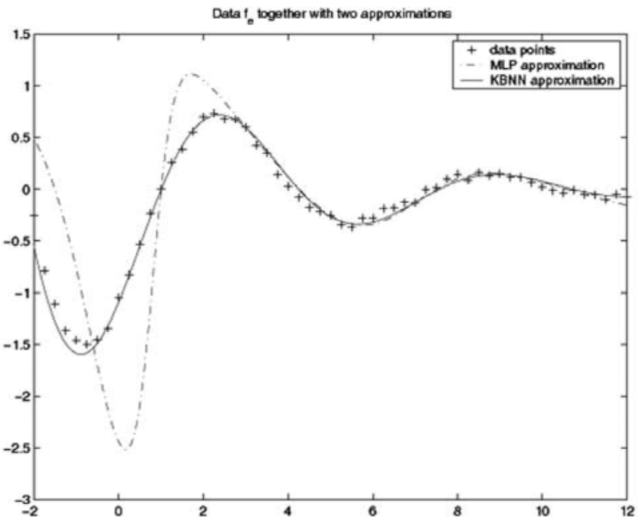

Figure 5. Comparison of approaches.

easily cope with situations where the cheap and expensive models differ only by a *scaling or a shift* in the inputs. The weights in (3) are parameters to be determined when training the network. Since the low fidelity model should be a reasonable approximation to the high fidelity model the matrix $\mathbf{W}_1$ should be close to the identity matrix and $\mathbf{w}_2$ should be close to the zero vector.

We note here that weighting the low fidelity models in this way has parallels with the space-mapping technique: we are essentially trying to find weights on the

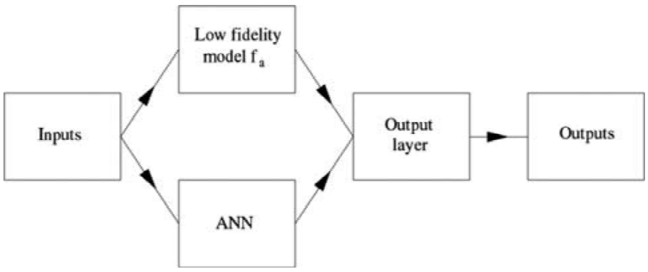

Figure 6. Multifidelity modelling using the KBNN framework. Note that the low fidelity model is treated as prior knowledge in a neural network setting.

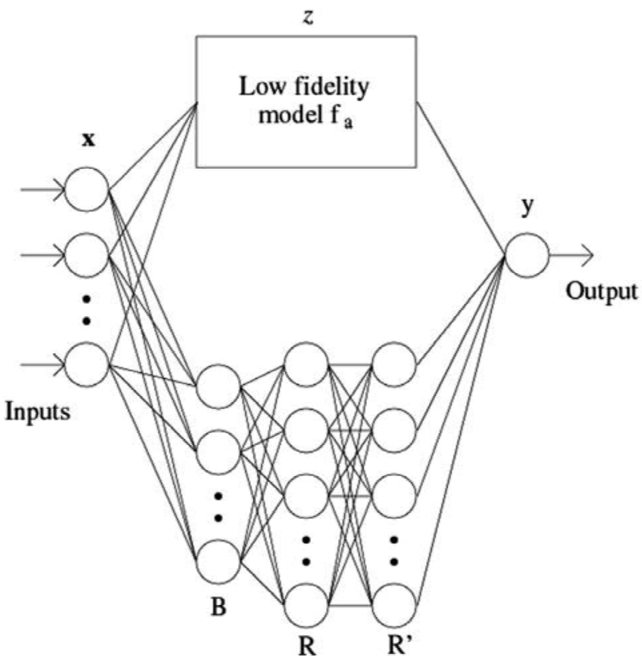

Figure 7. Application of Wang and Zhang's approach to multifidelity modelling.

inputs such that the low fidelity model is better correlated with the high fidelity model.

In the boundary layer, the neuron i is calculated as

$$b_i = \mathbf{B}(\mathbf{x}, \mathbf{v}_i), \qquad i = 1, ..., N_b. \tag{4}$$

This layer could also incorporate function knowledge as in Wang and Zhang (1997). However, we take it simply as the inner product of **x** and $\mathbf{v}_i$

$$b_i = \mathbf{x}^T \mathbf{v}_i, \qquad i = 1, ..., N_b, \tag{5}$$

where $\mathbf{v}_i$ are a set of free parameters that will be determined during the training process.

Using a sigmoid function $\mathcal{F}$, the region layer neurons are constructed from boundary neurons as

$$r_i = \prod_{j=1}^{N_b} \mathcal{F}(\alpha_{ij} b_j + \theta_{ij}), \qquad i = 1, 2, ..., N_r. \tag{6}$$

Here α_{ij} and θ_{ij} are scaling and bias parameters respectively.

The normalizing layer normalizes the outputs of the region layer, that is,

$$r'_i = \frac{r_i}{\sum_{j=1}^{N_r} r_j} \qquad i = 1, ..., N_{r'} = N_r. \tag{7}$$

Finally, the output is given by

$$y = \beta_1 z \left(\sum_{k=1}^{N_{r'}} \rho_k r'_k \right) + \beta_0, \tag{8}$$

where ρ_k are further parameters satisfying

$$\sum_{k=1}^{N_{r'}} \rho_k^2 = 1. \tag{9}$$

Note that the merging of the knowledge layer and the neural layer has been performed using multiplication. This is consistent with the approach of Wang and Zhang (1997). Clearly other choices of combining this information (e.g., addition) exist and could be considered. In this way simple relationships between the high and low fidelity models can be exploited. Extension to problems with multiple outputs on the lines of Wang and Zhang (1997) is possible but will not be considered here.

3.2. TRAINING THE NEURAL NETWORK THAT INCORPORATES LOW FIDELITY KNOWLEDGE

Let y represent the neural model output and f_e represent the expensive model output. The neural network learns from the training data $(\mathbf{x}_i, f_e(\mathbf{x}_i)),\ i = 1, 2, ..., N_{\text{data}}$. The trainable parameters are the knowledge weights $\mathbf{W}_1$ and $\mathbf{w}_2$, the boundary layer weights $\mathbf{v}_i,\ i = 1, 2, ..., N_b$, the scaling parameters α_{ij} and $\theta_{ij},\ i = 1, 2, ..., N_r,\ j = 1, 2, ..., N_b, \beta_1, \beta_0$ and $\rho_k,\ k = 1, 2, ..., N_{r'}$. For the 2D example described in Section 5, this requires a total of 33 parameters to be determined during training, the majority of these being due to the neural network structure.

The undetermined parameters are chosen to minimize the difference between neural network outputs y and the actual training outputs f_e in the least square sense. Thus we minimize

$$E = \frac{1}{2} \sum_{i=1}^{N_{data}} (y_i - (f_e)_i)^2 \tag{10}$$

with respect to these parameters.

The derivatives of E with respect to the unknown parameters are given in Wang and Zhang (1997) and can be used in gradient descent minimization. Updating the weights in this case requires modifying the traditional backpropagation algorithm, Rumelhart et al. (1996), slightly to cope with the different network topology. (Backpropagation was proposed for the MLP which has a rigorously ordered structure). Of course, other optimization strategies such as conjugate gradient minimization (Press et al., 1992) could be used to determine the weights.

4. Inclusion of Low Fidelity Knowledge in Kriging Models

Having presented a method to include cheap but low fidelity information along with expensive but high quality information in a neural network framework, we now turn to the problem of achieving the same in the context of kriging.

In a typical approximation method, the non-linear relationship between observations (responses) and independent variables is expressed as

$$y = f(\mathbf{x}) \tag{11}$$

where y is the observed response, $\mathbf{x}$ is a vector of k independent variables

$$\mathbf{x} = [x_1, x_2, ..., x_k] \tag{12}$$

and $f(\mathbf{x})$ is some unknown function. We define

$$\hat{y} = \hat{f}(\mathbf{x}), \tag{13}$$

an approximation for y based on kriging. A brief description of its implementation now follows. We then modify this classical approach to incorporate knowledge that comes from a weighted low fidelity model.

Given a set of N training data $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}]$ the kriging model can be used to make a prediction $\hat{y} = \hat{f}(\mathbf{x})$ at untested points $\mathbf{x}$ in the design space.

A correlation matrix of the training data

$$\mathbf{R}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp[-d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})] \tag{14}$$

is first sought where d is some distance measure. For example

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{h=1}^{k} \theta_h |x_h^{(i)} - x_h^{(j)}|^{p_h} \quad (\theta_h \geqslant 0,\ 1 \leqslant p_h \leqslant 2) \tag{15}$$

where θ_h and p_h are some as yet undetermined parameters.

When we wish to sample at a new point **x**, we form a vector of correlations between the new points and the training data

$$\mathbf{r}(\mathbf{x}) = \mathbf{R}(\mathbf{x}, \mathbf{x}^{(i)}) = [\mathbf{R}(\mathbf{x}, \mathbf{x}^{(1)}), ..., \mathbf{R}(\mathbf{x}, \mathbf{x}^{(N)})]. \tag{16}$$

The prediction is then given by

$$\hat{y}(\mathbf{x}) = \mu + \mathbf{r}^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu). \tag{17}$$

The parameters θ_h and p_h are determined by maximizing the likelihood of the sample

$$\frac{1}{(2\pi)^{\frac{N}{2}}(\sigma^2)^{\frac{N}{2}}|\mathbf{R}|^{\frac{1}{2}}} \exp\left[\frac{-(\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}\right] \tag{18}$$

where the parameters μ and σ^2 are given by

$$\mu = \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \tag{19}$$

and

$$\sigma^2 = \frac{(\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{N} \tag{20}$$

respectively. Note that this model strictly interpolates the training data.

4.1. TWO METHODS OF INCORPORATING MULTIFIDELITY DATA IN KRIGING

The simplest multifidelity modelling strategies using kriging would again model the difference or the ratio of two models at a given set of sampled points. That is, we may approximate

$$d = f_e - f_a \tag{21}$$

and add it to f_a to approximate f_e. Alternatively we may model

$$r = f_e/f_a, \tag{22}$$

and then take $\hat{r} f_a$ as a surrogate for f_e.

As a second approach, we propose to use low fidelity cheap information along with the high quality expensive information *within* the approximating model itself. This is in contrast with the first approach where some relationship between the data of two different fidelities is modelled explicitly. The cheap model, now taken as prior knowledge, can be suitably weighted to ensure best agreement between the two models of differing fidelity. The general structure of information flow is shown

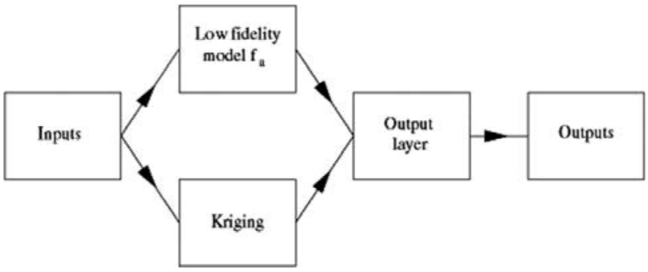

Figure 8. Schematic of the knowledge-based kriging approach with the low fidelity model treated as prior knowledge.

in Figure 8. Note the similarity in the approach of the strategy of Figure 6 and that of Figure 8. The only departure is in the way parameters are extracted in the two cases – while the algorithm underlying the diagram in Figure 6 uses an artificial neural network technique, that of Figure 8 uses kriging. In this way, in both cases, the low fidelity model is kept as an integral part of the approximation throughout. This is opposed to standard correction techniques where the low fidelity data do not inherently control the model training process. In mathematical terms, it amounts to a lack of implicit influence on the likelihood function that needs to be maximized (a discussion on this will follow shortly).

In this paper, we consider modelling a response with a single output. The inputs **x** are fed into the knowledge layer (the weighted low fidelity method) and into the kriging model. As shown in Figure 8, the knowledge layer outputs the value z, using the weighted low fidelity method according to Equation (3). The kriging model inputs **x** and outputs some prediction κ say. The output of the model can be defined, as before, in several ways, e.g., based on either addition $z + \kappa$ or multiplication $z \times \kappa$. These could also be weighted as in Equation (8). It may also be possible to let the model itself decide on the best functional form between the outputs of the knowledge layer z and the kriging prediction κ by using further parameters.

While considering the strategy of Figure 6 using neural networks, the undetermined weights are extracted by minimizing the sum of squares of differences (see Equation (10)). It is not any more possible to train the model in this way. This is because kriging models interpolate the data exactly, thus the difference between the data and the model is zero for all the sampled points, whatever our choice of weights. Therefore, the free parameters of the model (including the weights in the low fidelity model) need to be determined by maximizing the likelihood function of the sample as given by Equation (18). This ensures that the best model out of all possible interpolating models is chosen. We have set $p_h = 2$ and optimized with respect to $\theta_h,\ h = 1, ..., k$ and the weights in the knowledge layer. This typically results in a smaller optimization problem than in training the neural network that uses weighted low fidelity data as prior knowledge (as in the developments of Section 3). For the 2D example given in Section 5, this requires just a six-dimensional optimization problem (c.f., 33 for the approach of Section 3) Once again, this optimization problem can be tackled using standard techniques, for example, conjugate gradients.

Our kriging approach has some parallels with existing methodology in the literature (Sacks et al., 1989). The kriging prediction (17) uses a constant term μ added to some stochastic process term $Z(\mathbf{x}) = \mathbf{r}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)$. In the literature polynomial terms

$$\sum_{j=0}^{l} \beta_l f_l(\mathbf{x}) \tag{23}$$

have also been added to the stochastic process term. The β_l are regression parameters and the $f_l(\mathbf{x})$ are regression functions, the prediction is given by

$$\sum_{j=0}^{l} \beta_l f_l(\mathbf{x}) + Z(\mathbf{x}). \tag{24}$$

All unknown parameters are typically chosen by maximizing the likelihood.

We consider a model of the form $f_a(\mathbf{x}) + Z(\mathbf{x})$ in our difference approach or with the inputs to the low fidelity model weighted in our KBK approach, where the weightings again come from maximizing the likelihood. Clearly there are parallels with these approaches. Further, using

$$\sum_{j=0}^{l} \beta_l f_l(\mathbf{x}) \times Z(\mathbf{x}) \tag{25}$$

would have parallels with our ratio model.

Past experience suggests that using polynomial terms for the $f_l(\mathbf{x})$ rarely adds much accuracy to the kriging approximation. Whereas with our knowledge-based approach, using problem specific knowledge in the form of low fidelity models we are able to substantially increase the accuracy of the approximation.

As with the possible extension of the method presented in Section 3, we could consider multiple outputs: This can be achieved here using a cokriging approach ((see Cressie, 1993), for a description of cokriging). In this paper, however, we confine ourselves to problems with a single output.

We now turn our attention to demonstrating standard surrogate modelling approaches, correction approaches and the methods developed in this paper on two structural examples. In what follows, the methods that incorporate low fidelity models as knowledge inside the approximation are referred to as a knowledge-based neural network (KBNN) and a knowledge-based kriging model (KBK) respectively.

5. Examples

5.1. EXAMPLE 1

An example from the domain of structural design is presented now. Consider an elastic structure as shown in Figure 9. In this example we consider the length L to be 1 m. The horizontal beam is subjected to a uniformly distributed load $p_0 = 50$ N/m. We wish to minimize the weight of the structure by varying the cross section in various ways. We consider two two-dimensional problems as in Figure 10 and one four-dimensional problem as in Figure 11. In all cases the minimization is carried out subject to the constraints

$$\sigma_{max} < 100\,000 \text{ N/m}^2, \tag{26}$$

where σ_{max} is the maximum stress in the structure and

$$0.05 \text{ m} \leqslant t_i \leqslant 0.1 \text{ m} \tag{27}$$

where i varies from 1 to 2 or from 1 to 4 for the two- and four-dimensional problems, respectively.

The problem was analysed using a simple finite element beam model. Two levels of complexity were considered: a coarse model consisting of just four elements and a fine model consisting of 100 elements. In these two models the objective V (volume is proportional to weight) remains the same, whereas the stress, which forms the constraint, varies. It is this variation in stress between f_e and f_a that we attempt to model.

5.1.1. *2D beam problem*

For the purposes of approximation, the nine expensive data points ((0.05, 0.05), (0.075, 0.05), (0.1, 0.05), (0.05, 0.075), (0.075, 0.075), (0.1, 0.075), (0.05, 0.1), (0.075, 0.1), (0.1, 0.1)) are considered.

Results are presented for the following eight approaches:

- Low fidelity model optimization (Cheap)
- Kriging the expensive data at the sampled points and optimizing (Kriging)
- Kriging the difference $f_e - f_a$ at the sampled points adding this to f_a and optimizing (Addition).
- Kriging the ratio f_e/f_a at the sampled points multiplying this by f_a and optimizing (Ratio).
- The knowledge-based neural network approach (KBNN) presented in Section 3.
- The knowledge-based kriging approach presented in Section 4 using addition (KBK1).
- The knowledge-based kriging approach presented in Section 4 using multiplication (KBK2).

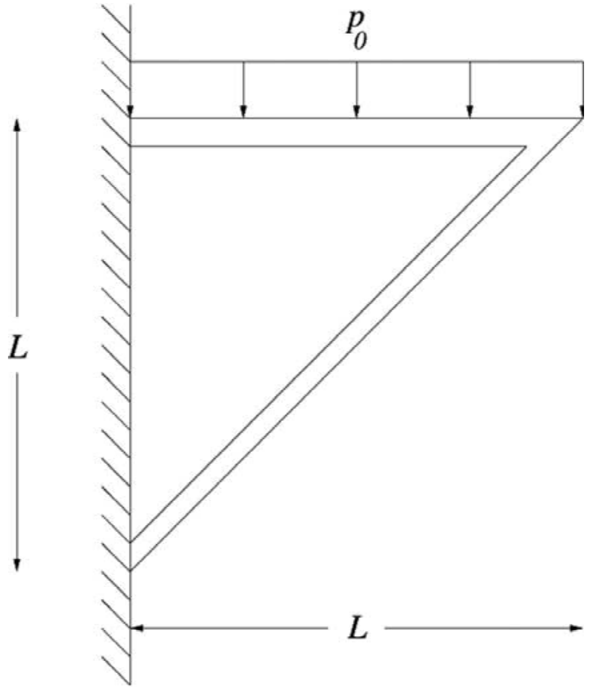

Figure 9. The problem.

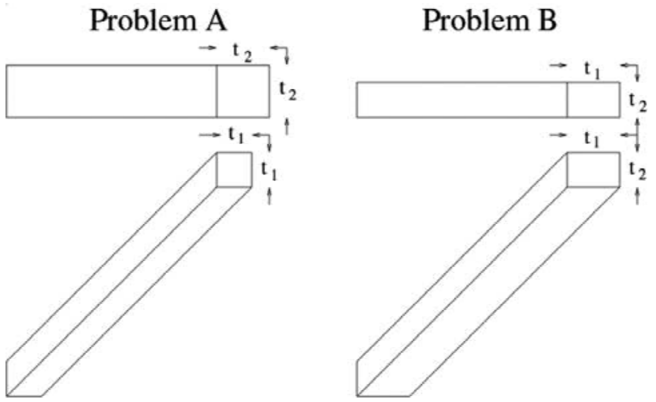

Figure 10. The two dimensional optimization problems - A) independent square sections, B) similar rectangular sections.

– Direct optimization of the high fidelity model (Expensive).

In both the KBNN and the KBK models we consider the elements of $\mathbf{W}_1$ in the range [0.75, 1.10] and those of $\mathbf{w}_2$ in the range $[-0.025, 0.025]$. In the knowledge-based neural network we take $N_b = N_r = N_{r'} = 3$. The results for problem A are shown in Table 1. The expensive model result is shown for comparison purposes only: in general this much information about the high fidelity model is unavailable.

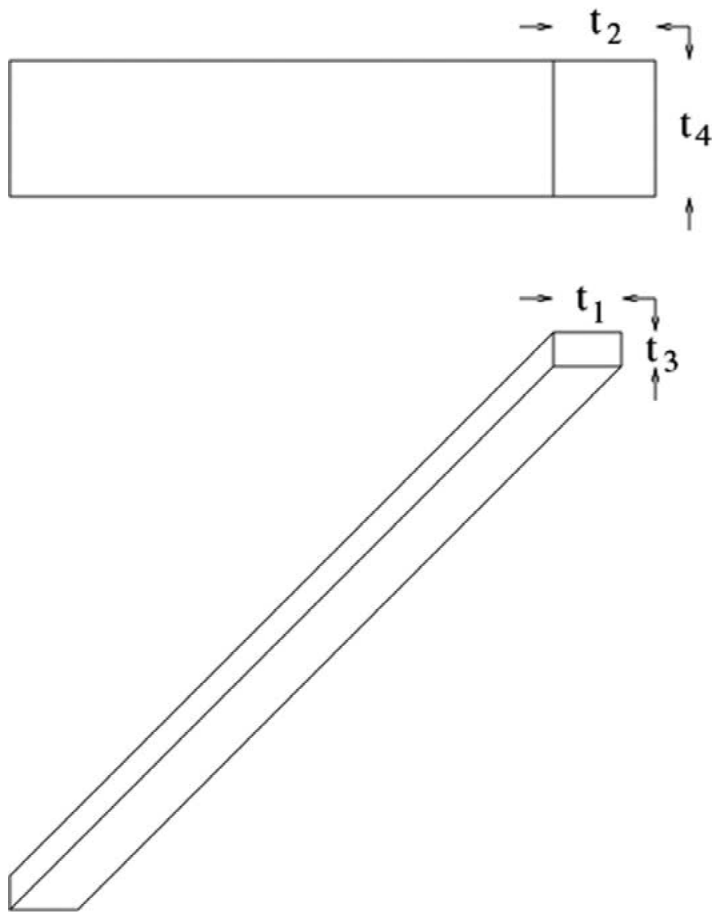

Figure 11. The four dimensional optimization problem.

Table 1 also lists the relative error (stress) in each model. This is an average error taken over 441 new test points spread throughout the design space. The error has been computed by taking the high fidelity model as exact. Table 2 lists the same results as Table 1, but for problem B.

It is clear from these tables that modelling using the nine high fidelity model response points alone leads to relatively large errors. Introducing knowledge in the form of a cheap approximation is beneficial as there is some degree of correlation between the models. We should expect this since the two models represent the same physical system. In the example, of the techniques discussed, the addition model performs worst. The ratio model performs better in this case but how these models perform relative to each other is problem dependent. The knowledge-based approaches perform better: the methods being more flexible than modelling the difference and ratio alone they are expected to outperform the addition and ratio models. It is not clear as to which of the knowledge-based approaches is likely to

Table 1. Results for 2D beam problem A

Model	t_1	t_2	V	Relative error
Cheap	5×10^{-2}	6.7846×10^{-2}	8.139×10^{-3}	1.837×10^{-1}
Kriging	5×10^{-2}	7.3944×10^{-2}	9.003×10^{-3}	9.267×10^{-2}
Addition	5×10^{-2}	7.2780×10^{-2}	8.832×10^{-3}	2.418×10^{-2}
Ratio	5×10^{-2}	7.2645×10^{-2}	8.813×10^{-3}	2.262×10^{-3}
KBNN	5×10^{-2}	7.2576×10^{-2}	8.803×10^{-3}	2.8160×10^{-4}
KBK1	5×10^{-2}	7.2693×10^{-2}	8.820×10^{-3}	1.272×10^{-2}
KBK2	5×10^{-2}	7.2576×10^{-2}	8.803×10^{-3}	1.723×10^{-3}
Expensive	5×10^{-2}	7.2571×10^{-2}	8.802×10^{-3}	N/A

Table 2. Results for 2D beam problem B

Model	t_1	t_2	V	Relative error
Cheap	5×10^{-2}	7.5101×10^{-2}	9.066×10^{-3}	1.811×10^{-1}
Kriging	5×10^{-2}	8.4340×10^{-2}	1.0181×10^{-2}	6.736×10^{-2}
Addition	5×10^{-2}	8.3597×10^{-2}	1.0091×10^{-2}	1.281×10^{-2}
Ratio	5×10^{-2}	8.3376×10^{-2}	1.0064×10^{-2}	6.663×10^{-5}
KBNN	5×10^{-2}	8.3380×10^{-2}	1.0065×10^{-2}	8.7170×10^{-6}
KBK1	5×10^{-2}	8.3379×10^{-2}	1.0065×10^{-2}	1.5870×10^{-4}
KBK2	5×10^{-2}	8.3379×10^{-2}	1.0065×10^{-2}	1.5740×10^{-5}
Expensive	5×10^{-2}	8.3379×10^{-2}	1.0065×10^{-2}	N/A

perform best in general, although the kriging based approach is much quicker to set up.

Figure 12 shows the objectives and constraint boundaries for the cheap, expensive and best approximation models for problem A. Similarly, Figure 13 shows the objectives and constraint boundaries for the cheap, expensive and best approximation models for problem B.

5.1.2. *4D beam problem*

We further consider the knowledge-based models on the 4D problem. The four parameters of the design space are the cross sectional properties of each beam. In this problem we use 21 space filling points to train the model. The high fidelity model is evaluated at these points only. The results for the low fidelity, KBNN, KBK1, KBK2 and expensive model are shown in Table 3. Once again, the expensive model optimization has been carried for comparison purposes only (it requires 185 expensive function evaluations using the L-BFGS-B optimizer [Zhu

Table 3. Results for 4D beam problem

Model	t_1	t_2	t_3	t_4	V
Cheap	5×10^{-2}	5×10^{-2}	5×10^{-2}	7.9943×10^{-2}	7.5327×10^{-3}
KBNN	5×10^{-2}	5×10^{-2}	5×10^{-2}	8.8470×10^{-2}	7.9590×10^{-3}
KBK1	5×10^{-2}	5×10^{-2}	5×10^{-2}	8.8562×10^{-2}	7.9637×10^{-3}
KBK2	5×10^{-2}	5×10^{-2}	5×10^{-2}	8.8576×10^{-2}	7.9643×10^{-3}
Expensive	5×10^{-2}	5×10^{-2}	5×10^{-2}	8.8427×10^{-2}	7.9569×10^{-3}

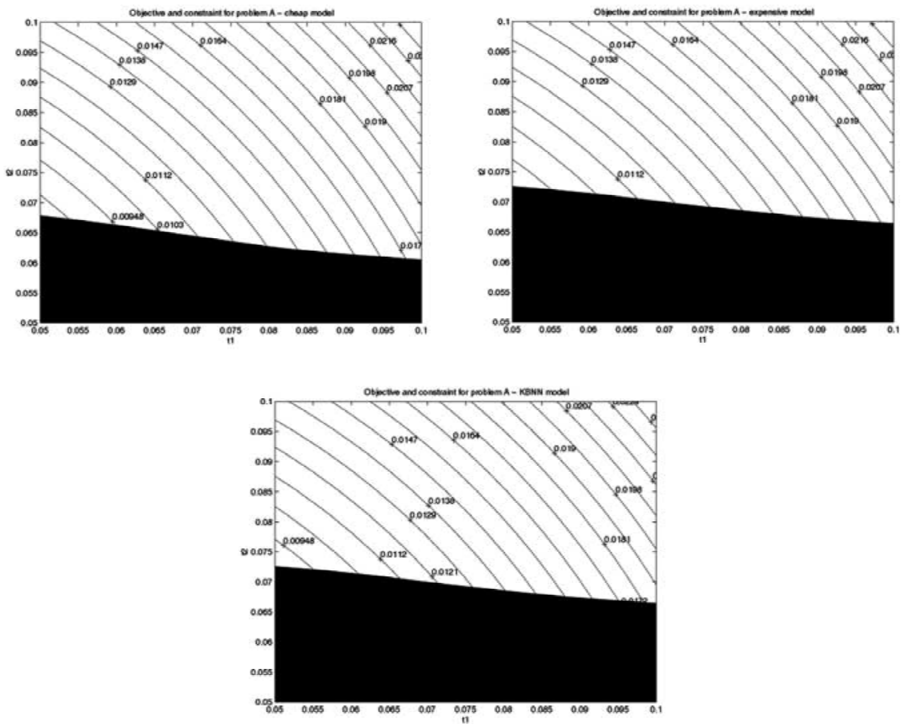

Figure 12. Objective and constraints for 2D beam problem A

et al., 1994] to optimize the problem in this way compared to just 21 using the knowledge-based approaches).

The knowledge-based approaches are again seen to perform well: including information from the low fidelity model leads to a prediction of the optimum which is very close to the true optimum in all cases.

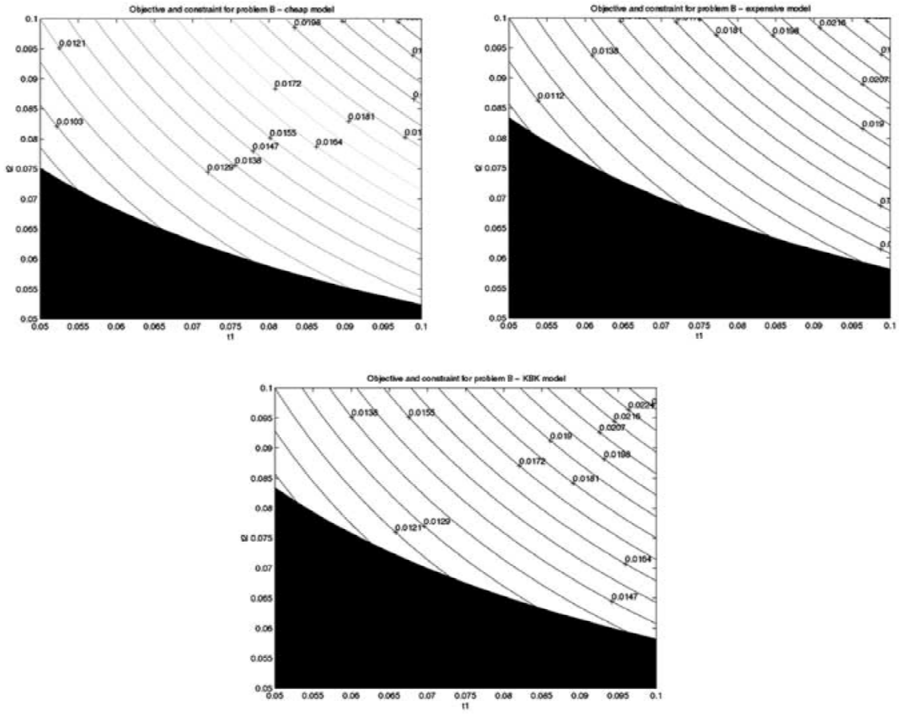

Figure 13. Objective and constraints for 2D beam problem B

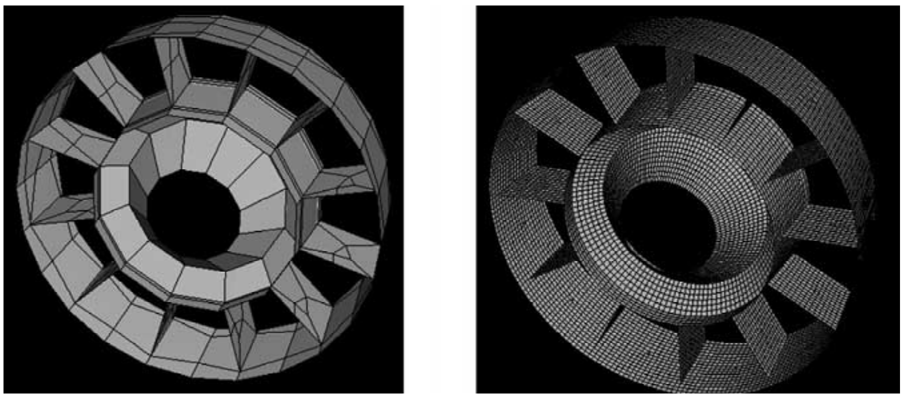

Figure 14. Example 2 - Low fidelity model (left) and high fidelity model (right).

5.2. EXAMPLE 2

As a final example we consider the design of an aero-engine component. Again the objective is to minimize the weight of the structure whilst keeping the stress at a key point below a prescribed value. The low fidelity model is shown in Figure 14 (left). This model consists of 246 finite elements and requires solution of a system of 1470 equations. A much more sophisticated high fidelity model is shown in

Figure 14 (right). This model consists of 11640 elements and requires solution of a system of 71064 equations. Clearly the former should be much quicker to solve than the latter and can be used as a guide to the behaviour of the high fidelity model. One might expect a well tuned solver dealing with banded matrices to scale with perhaps $O(N^2)$ – here this would give a ratio of run times of over 2000. In fact, because these models are, in absolute terms, both quite small, the savings are less because of the overheads associated with commercial finite element codes – here we saw a ratio closer to 20. In the following work an extremely accurate surrogate of the low fidelity model (built using 500 calls to f_a) is used in the knowledge layer. Such accuracy can be achieved, as the model is relatively simple to compute.

Here four design variables define the structural geometry, these are constrained within the following realistic bounds:

$$\begin{aligned} 1\text{ mm} &\leqslant x_1 \leqslant 4\text{ mm} \\ 2\text{ mm} &\leqslant x_2 \leqslant 5\text{ mm} \\ 2\text{ mm} &\leqslant x_3 \leqslant 5\text{ mm} \\ 2\text{ mm} &\leqslant x_4 \leqslant 5\text{ mm}. \end{aligned} \tag{28}$$

These variables relate to the thickness of the inner ring faces, inner ring thickness, outer ring thickness and spoke thickness, respectively. The loading on the structure is also given a realistic value as suggested by colleagues in industry.

Initially 16 runs of the high fidelity model are made at a space filling set of the input parameters and following this various approximations are sought. For the purposes of assessing our model's accuracy 484 further high fidelity model evaluations at alternative combinations of the inputs are made (but not used in model training). Our approximations are then compared with these results. There is reasonably good correlation between the resulting stresses in the high and low fidelity models. However, the error in the low fidelity model is large. The kriging model applied to the high fidelity model alone leads to a much reduced average error. Here we only report results from the most accurate multifidelity modelling strategy, the ratio model. This reduced the average error still further whilst the knowledge-based kriging model (using the ratio) led to the lowest error of all. Training the KBNN proved to be difficult in this example: we tried training a KBNN with three neurons per layer (43 optimization variables) as well as a KBNN with five neurons per layer (85 optimization variables). In both cases we were unable to fully train the model hence the results are poor. This highlights the potential difficulties with the KBNN approach. It might be that more neurons are required before we can obtain an acceptable approximation, this however would involve solving an even larger optimization problem during training. Full details of these results can be found in Table 4. In this case we chose the elements of $\mathbf{W}_1$ in the range [0.75, 1.10] and those of $\mathbf{w}_2$ in the range [−0.25, 0.25]. Finally the optimum design produced by the most accurate model (KBK) weighed 73.31 kg. The optimum design variables were

Table 4. Average errors in example 2

Model	Average % error
Low fid.	46.4919
Kriging	2.4074
Ratio	1.7040
KBNN(3)	3.2215
KBNN(5)	3.8932
KBK	1.4251

(2.0233, 2.0, 2.0, 2.0 mm), here the stress takes the value 2.013 N/mm^2, which is very close to our predefined maximum value of 2.0 N/mm^2.

In this case a direct optimization was performed and the resulting optimum design had a weight of 73.58 kg. The optimum design variables were (2.0547,2.0, 2.0,2.0 mm) and the stress value here was 1.972 N/mm^2. This required a total of 158 calls to the high fidelity model. Again we see the knowledge-based approach leads to a significant reduction in computational cost.

6. Conclusions

The idea of multifidelity modelling applied to expensive function optimization has been explored. Approaches that take account of lower fidelity models are more effective than standard response surface approaches built on expensive models alone because:

– they allow comparable approximations with fewer training data,
– they are more accurate at extrapolation.

In particular, the application of knowledge-based artificial neural networks (KBNN) to the problem of multifidelity modelling has been described. The prior knowledge used need not be very accurate or complete: one source of this knowledge is a low fidelity model. This leads us to a simple strategy. A new knowledge-based kriging (KBK) model that draws on these ideas is developed.

There is little to choose between the KBNN and KBK approaches based on results alone when they work: both are very accurate for Example 1. The advantage of the kriging model lies in the training. Gibbs and Mackay (1997) note that determining the most probable values of the hyperparameters in a Gaussian process framework is a straightforward process. Kriging falls within this framework. Optimizing hyperparameters for neural networks is usually a more complicated problem and is sometimes very difficult to achieve. In the first two-dimensional example given, training the KBNN required solving a 33-variable optimization problem, whereas training the KBK required solving only a six variable optimization problem which is, of course, considerably easier. In the second example an adequate neural network was not achieved at all. See Rasmussen (1996) for an extensive comparison of Gaussian process models and neural networks.

The knowledge-based approaches provide improved accuracy on a global scale compared to the other methods described. Clearly the first example given is somewhat simple, however it does provide a benchmark result so we can compare the approaches easily. It also allows for visualization of the principles involved. The second example demonstrates the approach on a more realistic problem. Further work will consider the application of these methods on even more challenging problems in structural optimization.

Acknowledgements

We would like to thank Rolls-Royce Plc for the financial support that the first author receives as a part of the University Technology Partnership (UTP). We also thank Dr. Alison McMillan and Dr. Steve Wiseall of Rolls-Royce Plc for several useful discussions and for their industrial feedback. We gratefully acknowledge Willard McClune and Cecil Armstrong of Queens University Belfast as part of the MMFSC project for kindly providing the model used in Example 2 and Dr. Ivan Voutchkov for assisting in generating the results. Finally, we are very grateful to two anonymous referees whose comments helped to considerably improve the paper.

References

1. Bakr, M. H., Bandler, J. W., Madsen, K., Rayas-Sanchez, J. E. and Sondergaard, J. (2000a), Space-mapping optimization of microwave circuits exploiting surrogate models, *IEEE Transactions on Microwave Theory and Techniques* 48: 2297–2306.
2. Bakr, M. H., Bandler, J. W., Ismail, M. A., Rayas-Sanchez, J. E. and Zhang., Q. (2000b). Neiral space-mapping for EM-based design, *IEEE Transactions on Microwave Theory and Techniques* 48: 2307–2315.
3. Bandler, J. W., Ismail, M. A., Rayas-Sanchez, J. E. and Zhang., Q. (1999). Neuromodelling of microwave circuits exploiting space-mapping technology, *IEEE Transactions on Microwave Theory and Techniques* 47: 2417–2427.
4. Bandler, J. W., Georgieva, N., Ismail, M. A., Rayas-Sanchez, J. E. and Zhang., Q. (2001), A generalized space-mapping tableau approach to device modelling, *IEEE Transactions on Microwave Theory and Techniques* 49: 67–79.
5. Chang, K. J., Haftka, R. T., Giles, G. L. and Kao, P. -J. (1993). Sensitivity-based scaling for approximating structural response, *Journal of Aircraft* 30: 283–287.
6. Cressie, N. A. C. (1993). *Statistics for Spatial Data*, John Wiley and Sons Inc, New York.
7. Gibbs, M. N. and Mackay, D. J. C. (1997). Efficient implementation of Gaussian processes, http://wol.ra.phy.cam.ac.uk/mackay/abstracts/gpros.html
8. Haftka, R. T. (1991). Combining global and local approximations, *AIAA Journal* 29: 1523–1525.

9. Hutchinson, M. G., Unger, E. R., Mason, W. H., Grossman, B. and Haftka, R. T. (1994). Variable-complexity aerodynamic optimization of a high speed civil transport wing, *Journal of Aircraft* 31: 110–116.
10. Jones, D. R. (1998). Schonlau, M. and Welch, W. J., Efficient global optimization of expensive black-box functions, *Journal of Global Optimization* 13: 455–492.
11. Myers, R. H. and Montgomery, D. C. (1995). *Response surface methodology: Process and product optimization using designed experiments*, John Wiley and Sons inc, New York.
12. Press, W. H., Teukolsky, S. A., Vellering, W. T. and Flannery, B. P. (1992). *Numerical recipes in fortran, 2nd Edition*, Cambridge University Press, Cambridge.
13. Rasmussen, C. E. (1996). *Evaluation of Gaussian processes and other methods for nonlinear regression*, PhD Thesis, University of Toronto.
14. Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by backpropagating errors, *Nature* 323: 533–536.
15. Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989). Design and Analysis of Computer Experiments, *Statistical Science* 4: 409–435.
16. Towell, G. G. and Shavlik, J. W. (1994). Knowledge-based artificial neural networks, *Artificial Intelligence* 70: 119–165.
17. Vitali, R., Haftka, R. T. and Sankar, B. V. (1999). Multifidelity design of a stiffened composite panel with a crack, *4th World Congress of Structural and Multidisciplinary Optimization*, Buffalo, NY.
18. Wang, F. and Zhang, Q. (1997). Knowledge-based neural models for microwave design, *IEEE Transactions on Microwave Theory and Techniques* 45: 2333–2343.
19. Watson, P. M. and Gupta, K. C. (1996). EM-ANN models for microstrip vias and interconnects in dataset circuits, *IEEE Transactions on Microwave Theory and Techniques* 44: 2495–2503.
20. White, H., Gallant, A. R., Kornik, K., Stinchcombe, M. and Wooldridge, J. (1992). *Artificial Neural Networks: Approximation and Learning Theory*, Blackwell publishers, Oxford.
21. Zhu, C., Byrd R. H., Lu P. and Nocedal J. (1994). *L-BFGS-B: a limited memory FORTRAN code for solving bound constrained optimization problems*, Tech. Report, NAM-11, EECS Department, Northwestern University.