Struct Multidisc Optim (2009) 38:103–115
DOI 10.1007/s00158-008-0286-8

RESEARCH PAPER

Metamodel-based collaborative optimization framework

Parviz M. Zadeh · Vassili V. Toropov · Alastair S. Wood

Received: 26 July 2007 / Revised: 20 May 2008 / Accepted: 8 June 2008 / Published online: 22 August 2008

Abstract This paper focuses on the metamodel-based collaborative optimization (CO). The objective is to improve the computational efficiency of CO in order to handle multidisciplinary design optimization problems utilising high fidelity models. To address these issues, two levels of metamodel building techniques are proposed: metamodels in the disciplinary optimization are based on multi-fidelity modelling (the interaction of low and high fidelity models) and for the system level optimization a combination of a global metamodel based on the moving least squares method and trust region strategy is introduced. The proposed method is demonstrated on a continuous fiber-reinforced composite beam test problem. Results show that methods introduced in this paper provide an effective way of improving computational efficiency of CO based on high fidelity simulation models.

Keywords Multidisciplinary design optimization · Collaborative optimization · Metamodel · Moving least squares method · Multi-fidelity modelling

1 Introduction

Most real world design problems are complex and multidisciplinary in nature. For example, both aircraft and automotive design are characterized by multidisciplinary interactions in which participating disciplines are intrinsically coupled to each other. Over the past several years there has been significant progress in the application of multidisciplinary design optimization (MDO) to solving such complex design problems. One of typical important problems associated with MDO is a high computational cost of analysis in individual disciplines. The interdisciplinary couplings inherent in MDO cause increased computational effort beyond that encountered in a single discipline optimization. The analysis codes for each discipline must interact with each other for the purpose of system optimization. These issues result in a very high overall computational cost that can become prohibitive. Several MDO approaches have been proposed that include multiple-discipline feasible (MDF; Kodiyalam and Sobieszczanski-Sobieski 2001), all-at-once (AAO) and individual discipline feasible (IDF; Cramer et al. 1992), collaborative optimization (CO; Kroo et al. 1994), bi-level integrated synthesis (BLISS; Sobieszczanski-Sobieski et al. 1998), concurrent subspace optimization

P. M. Zadeh · A. S. Wood
School Engineering, Design and Technology,
University of Bradford,
Bradford, BD7 1DP, UK

P. M. Zadeh
e-mail: eorg@ut.ac.ir

A. S. Wood
e-mail: a.s.wood@bradford.ac.uk

Present Address:
P. M. Zadeh
Engineering Optimization Research Group,
University of Tehran, Tehran, Iran

V. V. Toropov (✉)
Schools of Civil and Mechanical Engineering,
University of Leeds, Leeds,
West Yorkshire, LS2 9JT, UK
e-mail: v.v.toropov@leeds.ac.uk

(CSSO; Sobieszczanski-Sobieski 1988) and analytical target cascading (ATC; Kim et al. 2003).

The key concept in the CO approach is the decomposition of the design problem into two levels, namely disciplinary and system levels. The system level optimizer is used to minimize the system level objective while satisfying consistency requirements among the disciplines by enforcing equality constraints at the system level that coordinate the interdisciplinary couplings (Kroo et al. 1994). CO is designed in such a way that it supports disciplinary autonomy while maintaining interdisciplinary compatibility thus providing added design flexibility. These features make CO well suited for use in a practical multidisciplinary design environment (Kroo 1995; Braun et al. 1996; Sobieski and Kroo 1996).

Despite these advantages, the CO methodology has not become a mainstream design optimization tool in industry due to high computational costs involved that is a feature common to all MDO approaches. In addition, an important difficulty specifically associated with CO is its slow system level optimization convergence rate. This relates to the fact that CO ensures interdisciplinary compatibility by means of system level equality constraints and attempts to minimize the disagreement between disciplines by sending targets to individual disciplines that their optimization runs are required to meet. The discipline level optima can be non-smooth and noisy functions of the system level variables. This, combined with the use of equality constraints at the system level to represent disciplinary feasible regions introduces computational difficulties (Alexandrov and Lewis 2000; Alexandrov et al. 1999). The implications of these issues are that derivative-based optimization techniques cannot be used for the system level optimization whereas more robust optimization techniques such as genetic algorithms are prohibitively expensive for the use in CO. These issues pose significant barriers to real-life applications of CO based on high fidelity simulation models.

It has become increasingly important to utilize the most accurate simulation models in the design process to ensure a realistic design, hence high fidelity models (e.g. finite element, computational fluid dynamics, etc.) are now used extensively. However, the considerable computational expense associated with such models makes it difficult to use them in optimization, where achieving a solution may require hundreds or even thousands of simulation calls. Because of this, in the field of design optimization in general, and MDO in particular, the use of approximations became an effective tool for reducing the computational effort. The basic approach is to replace the computationally expensive simulation model by an approximate one, which is then used in optimization runs. Such an approximate model is often referred to as a metamodel ("model of a model"). A survey on the use of metamodels in structural optimization has been carried out by Barthelemy and Haftka (1993) and in MDO by Sobieszczanski-Sobieski and Haftka (1997) and Simpson et al. (2004). The main benefits of using metamodels in MDO include (Giunta et al. 1995): (1) reduction of the number of expensive high fidelity simulations during optimization; (2) smoothing numerical noise; (3) enabling separation of the simulation code from the optimization routines and easing the integration of codes from various disciplines; (4) making possible the utilization of parallel computer architecture. While the use of metamodels has now become an integral part of an optimization process, allowing a significant reduction of computational costs, it does have some limitations. The cost of providing high fidelity data for building the global metamodel increases rapidly with the number of design variables. Toropov and Markine (1996) suggested the use of low fidelity models (e.g. simplified numerical models) as the basis for the high quality metamodel building.

There have been several metamodel building approaches based on low fidelity models. Mason et al. (1994) used a coarse 2D finite element model as a low fidelity model to predict failure stresses, and corrections are calculated using a full 3D finite element model. Venkataraman et al. (1988) demonstrated the effectiveness of correcting inexpensive analysis based on low fidelity models by results from more expensive and accurate models in the design of shell structures for buckling. Vitali et al. (1998) used a coarse low fidelity finite element model to predict the stress intensity factor, and corrected it with high fidelity model results based on a detailed finite element model for optimizing a blade-stiffened composite panel. These approaches have been developed for single discipline problems. In the field of MDO, there have been several approaches in the use of metamodels to large multidisciplinary optimization problems. The variable complexity modelling proposed by Giunta (1997) simultaneously utilizes low and high fidelity models. The low fidelity predictions were corrected using a scaling factor obtained from the high fidelity model. Toropov and Markine (1996) and Rodriguez et al. (1997) introduced a trust region approach for low fidelity approximate model management for constrained approximate optimization problems to manage convergence. Alexandrov et al. (1998) proposed the trust region management framework to manage the solution of MDO by alternating the use of high and low fidelity models during the optimization process.

In this study two levels of metamodel building techniques are introduced in order to significantly reduce the computational effort while handling high fidelity discipline level simulation models in CO. Metamodels in the disciplinary optimization are based on multi-fidelity modelling and for the system level optimization global metamodels are introduced using the moving least squares method (MLSM) combined with a trust region strategy. The multi-fidelity modelling consists of computationally efficient simplified numerical models (low fidelity) and expensive detailed (high fidelity) models. The main advantage of using simplified numerical models is that they reflect the most prominent features of the original model whilst remaining computationally inexpensive (can be used repeatedly in the optimization process).

2 Collaborative optimization

CO is a bi-level optimization framework, with discipline-specific optimization runs that are free to obtain local designs, and a system level optimizer to coordinate this process while minimizing the overall design objective (Kroo 1995; Braun et al. 1996). Using the terminology defined by Alexandrov and Lewis (2000), and, for brevity, assuming that there are only two individual disciplines in an MDO problem, each discipline is based on a disciplinary analysis, which takes as its input a set of design variables $\boldsymbol{l}_i$, a set of system level design variables $\boldsymbol{s}$ (also referred to as shared variables) and some other parameters. The total outputs from a discipline i including all data passed to the other discipline as parameters and quantities passed to design constraints and objectives are denoted by $\boldsymbol{q}_i$. For example, $\boldsymbol{q}_1 = \left\{\boldsymbol{q}_1^p,\ \boldsymbol{q}_1^g\right\}$ where $\boldsymbol{g}_1^p$ are parameters passed to the discipline 2 (e.g. aerodynamics loads), and $\boldsymbol{q}_1^g$ are the responses essential to the discipline 1 (e.g. drag or lift) that serve in the definition of the objective and constraints of discipline 1.

The system level optimization problem for a two-discipline case can be stated as (Alexandrov and Lewis 2000):

$$\text{Minimize :} \quad f\,(\boldsymbol{s},\ \boldsymbol{t}_1,\ \boldsymbol{t}_2) \tag{1}$$

$$\text{subject to :} \quad \boldsymbol{C}\,(\boldsymbol{s},\ \boldsymbol{t}_1,\ \boldsymbol{t}_2) = 0 \tag{2}$$

where $\boldsymbol{C} = \{C_1,...,C_N\}$ are the N interdisciplinary consistency (or compatibility) constraints, and $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are the interdisciplinary coupling variables for discipline 1 and 2, respectively, serving as system level targets for the disciplinary outputs $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. Note that the output $\boldsymbol{q}_i$ of the discipline i can be an input for the discipline j.

The system level optimization provides target values $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ and required values of shared (system level) variables to disciplines 1 and 2. The discipline level optimizer seeks to match these values ($\boldsymbol{s}$, $\boldsymbol{t}_1$, $\boldsymbol{t}_2$) passed down by the system level. The variables $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ are introduced to relax the coupling between the disciplines introduced by the shared design variables $\boldsymbol{s}$. Thus the variables $\boldsymbol{\psi}_i$ are used as local copies (for discipline i) of the shared variables $\boldsymbol{s}$.

For discipline 1, $\overline{\boldsymbol{\psi}}_1\,(\boldsymbol{s}, \boldsymbol{t}_1, \boldsymbol{t}_2)$ and $\bar{\boldsymbol{l}}_1\,(\boldsymbol{s}, \boldsymbol{t}_1, \boldsymbol{t}_2)$ are components of the solution of the following minimization problem in $\boldsymbol{\psi}_1$ and $\boldsymbol{l}_1$:

$$\text{Minimize :} \quad \left\|\boldsymbol{\psi}_1 - \boldsymbol{s}\right\|^2 + \left\|\boldsymbol{q}_1^g\,(\boldsymbol{\psi}_1, \boldsymbol{l}_1, \boldsymbol{q}_2^{ps}) - \boldsymbol{t}_1\right\|^2 \tag{3}$$

$$\text{subject to :} \quad \boldsymbol{g}_1\Big(\boldsymbol{\psi}_1, \boldsymbol{l}_1, \boldsymbol{q}_1^g\,(\boldsymbol{\psi}_1, \boldsymbol{l}_1, \boldsymbol{q}_2^{ps})\Big) \le \boldsymbol{g}_1^* \tag{4}$$

where ($\boldsymbol{s}$, $\boldsymbol{t}_1$, $\boldsymbol{t}_2$) are target values passed to the disciplinary optimizer by the system level as parameters (fixed values), $\boldsymbol{q}_i^{ps}$ are parameters that are frozen in the discipline i, $\boldsymbol{g}_1$ are disciplinary constraint functions and $\boldsymbol{g}_1^*$ are limits on them.

Similarly, discipline 2 optimization problem can be expressed as:

$$\text{Minimize :} \quad \left\|\boldsymbol{\psi}_2 - \boldsymbol{s}\right\|^2 + \left\|\boldsymbol{q}_2^g\,(\boldsymbol{\psi}_1, \boldsymbol{l}_2, \boldsymbol{q}_1^{ps} - \boldsymbol{t}_2)\right\|^2 \tag{5}$$

$$\text{subject to :} \quad \boldsymbol{g}_2\Big(\boldsymbol{\psi}_2, \boldsymbol{l}_2, \boldsymbol{q}_2^g\,(\boldsymbol{\psi}_2, \boldsymbol{l}_2, \boldsymbol{q}_1^{ps})\Big) \le \boldsymbol{g}_2^* \tag{6}$$

The minimum values of the compatibility functions (3) and (5) will then serve as constraints (2) in the system level optimization problem.

3 Metamodel-based CO framework

The organization of the optimization process and the main components of the metamodel-based CO framework are shown in Fig. 1 and are described in the sections below.

3.1 Disciplinary optimization

Constraints g_i at the discipline level in CO correspond to functions describing the behaviour of a typical engineering system as related to a particular discipline i. The construction of metamodels at the discipline level is based on well-established global metamodelling concepts. In this paper, the multi-fidelity modelling concept (Toropov and Markine 1996; Zadeh and Toropov 2002) is used. Here, the multi-fidelity modelling concept is implemented to simultaneously utilize computational models of different levels of fidelity in a CO process

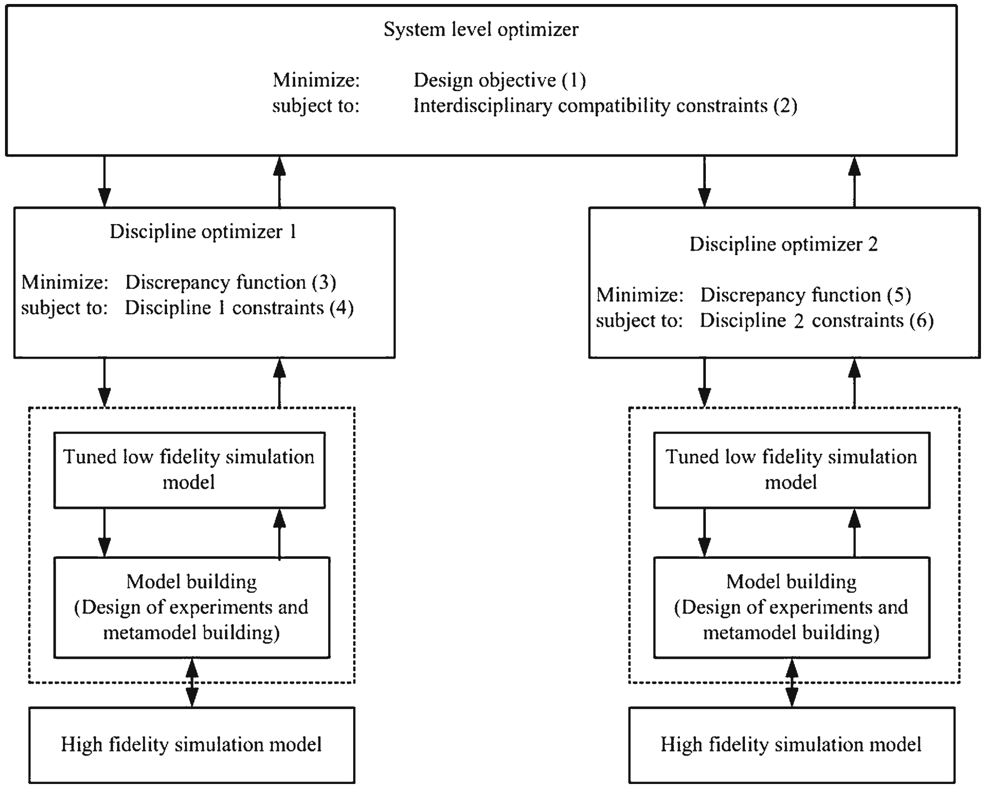

Fig. 1 Metamodel-based collaborative optimization framework

to facilitate the solution of an MDO problem with high fidelity models, see schematic illustration in Fig. 1.

It consists of expensive detailed (high fidelity) models and computationally efficient simplified numerical (low fidelity) models. The low fidelity models are tuned using a small number of high fidelity model runs and then used in place of expensive high fidelity models in the optimization process. The tuned low fidelity model approaches the same level of accuracy as a high fidelity model but at the same time remains computationally inexpensive to be used repeatedly in the optimization process.

3.1.1 Metamodels based on low fidelity numerical models

The quality of a metamodel has a profound effect on the computational cost and convergence characteristics of metamodel-based optimization. It was shown by Box and Draper (1987) that a mechanistic model (one that is built upon some prior knowledge of the system) can often provide a higher quality metamodel than a purely empirical (e.g. polynomial) model. An example of an application to material parameter identification can be seen in Toropov and van der Giessen (1993). A more general way of constructing high quality metamodels adopted here is based on the use of a lower fidelity numerical model (Toropov and Markine 1996). This can be achieved either by simplifying the analysis model (e.g. coarser finite element mesh) or simplifying the modelling concept (e.g. simpler geometry and boundary conditions, use of 2D instead of 3D model, etc.). Such a model should reflect the most prominent physical features of the system under consideration and at the same time remain computationally inexpensive. The low fidelity numerical model can then be used to build the metamodel as follows:

$$\tilde{F}(\boldsymbol{x}, \boldsymbol{a}) \equiv \tilde{F}(f(\boldsymbol{x}), \boldsymbol{a}) \tag{7}$$

where $\boldsymbol{x}$ is the vector of all design variables in a discipline being considered including the local copies $\boldsymbol{\psi}$ of the system level variables and the local variables $\boldsymbol{l}$, see (3), (4) and (5), (6); $f(\boldsymbol{x})$ is the function presenting a response obtained by the low fidelity model. The tuning parameters $\boldsymbol{x}$ are used for minimizing the discrepancy between the high fidelity and the low fidelity models. In building metamodels, $\tilde{F}(\boldsymbol{x})$, it is necessary to select an appropriate structure of the metamodel, i.e. to define them as a function of design variables $\boldsymbol{x}$ and tuning parameters $\boldsymbol{a}$. The efficiency of the optimization process depends strongly on the accuracy of such a metamodel. Zadeh and Toropov (2002) examined several types of metamodel functions on a test problem. The selection of the best metamodel function was based on the error of the corrected low fidelity model as compared to the high fidelity model at points of a design of experiments (DoE) referred to as a verification DoE. The following types of the metamodel functions have been suggested by Toropov and Markine (1996).

Type 1 Linear and Multiplicative Metamodel with Two Tuning Parameters

The simplest form of the metamodel function is a linear or intrinsically linear function with two tuning parameters, which can be represented as follows:

$$\text{Type 1 linear}: \quad \tilde{F}(\boldsymbol{x}, \boldsymbol{a}) = a_0 + a_1 f(\boldsymbol{x}) \tag{8}$$

$$\text{Type 1 multiplicative}: \quad \tilde{F}(\boldsymbol{x}, \boldsymbol{a}) = a_0 f(\boldsymbol{x})^{a_1} \tag{9}$$

where $\boldsymbol{a} = [a_0 \quad a_1]^T$

Type 2 Correction Functions

The type 2 metamodels are based on an explicit correction function $C(\boldsymbol{x}, \boldsymbol{a})$ that depends on the design variables and tuning parameters, such as:

multiplicative function:

$$\tilde{F}(\boldsymbol{x}, \boldsymbol{a}) = f(x)\ C(\boldsymbol{x}, \boldsymbol{a}) \tag{10}$$

where $C(\boldsymbol{x}, \boldsymbol{a}) = a_0 \prod_{l=1}^{\mathrm{N}} x_l^{a_1}$;
or a linear function:

$$\tilde{F}(\boldsymbol{x}, \boldsymbol{a}) = f(\boldsymbol{x}) + C(\boldsymbol{x}, \boldsymbol{a}) \tag{11}$$

where $C(\boldsymbol{x}, \boldsymbol{a})$ can be a low order polynomial in $\boldsymbol{x}$ with coefficients $\boldsymbol{a}$, e.g. a linear, quadratic or a cubic function.

Type 3 Use of model inputs as tuning parameters

A low fidelity model, $f(\boldsymbol{x})$, which is used as a basis for building an metamodel depends not only on the design variables $\boldsymbol{x}$ but also on other parameters, e.g. geometrical and material properties, etc. Some of these parameters can be considered as tuning parameters $\boldsymbol{a}$ so that a metamodel function can be represented in the following form:

$$\tilde{F}(\boldsymbol{x}, \boldsymbol{a}) = f(\boldsymbol{x}, \boldsymbol{a})\,. \tag{12}$$

Once the metamodel type is chosen, the tuning parameters $\boldsymbol{a}$ included in the metamodel are obtained by minimizing the sum of squares of the errors over the p sampling points at which both high fidelity and low fidelity models have been run:

$$\text{Minimize}: \quad G(\boldsymbol{a}) = \sum_{p=1}^{P} w_p \left(F_p - \tilde{F}(\boldsymbol{x}_p, \boldsymbol{a})\right)^2 \tag{13}$$

where $\boldsymbol{x}_p$ is pth sampling point, w_p is a weight that determines the relative contribution of the information at the pth point and F_p is the corresponding value of response from the high fidelity model, for more details see Toropov (2001).

The linear and multiplicative functions of type 1 and 2 have been successfully used for a variety of design optimization problems (Toropov 2001). The advantage of these metamodels is that a relatively small number of tuning parameters $\boldsymbol{a}$ are required. For higher order metamodels of type 2, such as full quadratic and cubic polynomials, the error measure reduces but metamodels can become prone to overfitting. In addition, these metamodels would require a significantly larger number of designs to be evaluated. The type 3 metamodel similarly to mechanistic metamodels, is based on deeper understanding of a process being modelled, which can be useful (Toropov 2001) but is problem-dependent.

3.1.2 Design of experiments

The selection of sampling points in the design variable space, where the response must be evaluated, is commonly referred to as Design of Experiments (DoE). DoE is an essential part of metamodel building process, and allows the designer to build metamodels more efficiently by employing set points at which to evaluate the response function(s). The planning of DoE (i.e. the choice of points in the design variable space) can have a considerable effect on the accuracy and the efficiency of the metamodel building. There are many schemes available in the literature for generating plans of experiments (Simpson et al. 1997). In this study the scheme suggested by Audze and Eglais (1977) is adopted. It is a space-filling DoE that is based on a Latin hypercube design.

The Latin hypercube (LH) approach McKay et al. (1979) is independent of the mathematical model of a problem. It can be considered as an N-dimensional extension of the traditional Latin square design (Montgomery 1997). In each level of every design variable only one point is placed. In this approach the number of design points to be analysed can be defined without an explicit dependence on the number of design variables. There are two variations of LH approach, the random sampling and the optimal LH method. The optimal LH method is based on optimizing the uniformity of the distribution of the design points. There are a number of methods are available for obtaining optimal LH, e.g. Bates et al. (2004). Audze and Eglais (1977) introduced an optimality criterion using an analogy with the potential energy of the system of points in the design space for obtaining a uniform distribution of design points that is used in this work.

3.2 System level optimization

Constraints at the system level are equality constraints (discrepancy functions $\boldsymbol{C} = 0$) in (2) that are much more computationally expensive as compared to discipline level constraints. Values of the system level constraints are obtained by solving disciplinary optimization problems and correspond to a measure of disagreement between the targets given to a discipline by the system level optimizer. Hence, they are non-smooth at the transition from a plane of zero values (discipline compliance with given targets) to a region of non-zero values (discipline non-compliance). This feature can cause slow convergence of the CO system level optimization. Also, these characteristics of the system level optimization in CO make it more demanding in terms of a choice of a metamodelling technique.

Figure 2a and b show a 50-point uniform Latin Hypercube plan, obtained corresponding to the system level design variables for disciplines 1 and 2 for the test problem used in this study (see Section 4 below). In these figures, larger dots indicate points at which the corresponding disciplinary optimizer returned non-zero values of the objective function. The remaining points correspond to zero values of the disciplinary optimization runs.

An initial study was carried out using a cubic polynomial response surface to build a metamodel for the system level optimization. Figure 3 shows the discipline 1 objective function (a constraint at the system level) represented by a cubic polynomial response surface with negative values removed by forcing them to zero.

The cubic metamodel captures the basic behavior of the function, but can exhibit an unrealistic non-zero domain ("hump" in Fig. 3) in the region of the zero-level plateau. It is therefore necessary to employ a metamodel strategy, suitable for the characteristics of the discrepancy function, for more accurate and efficient modelling within a CO framework. Jang et al. (2005) used kriging-based metamodels to approximate the system level constraints and Zadeh et al. (2004) used the moving least squares method (MLSM) for the same purpose In this study the use MLSM is demonstrated on a test example.

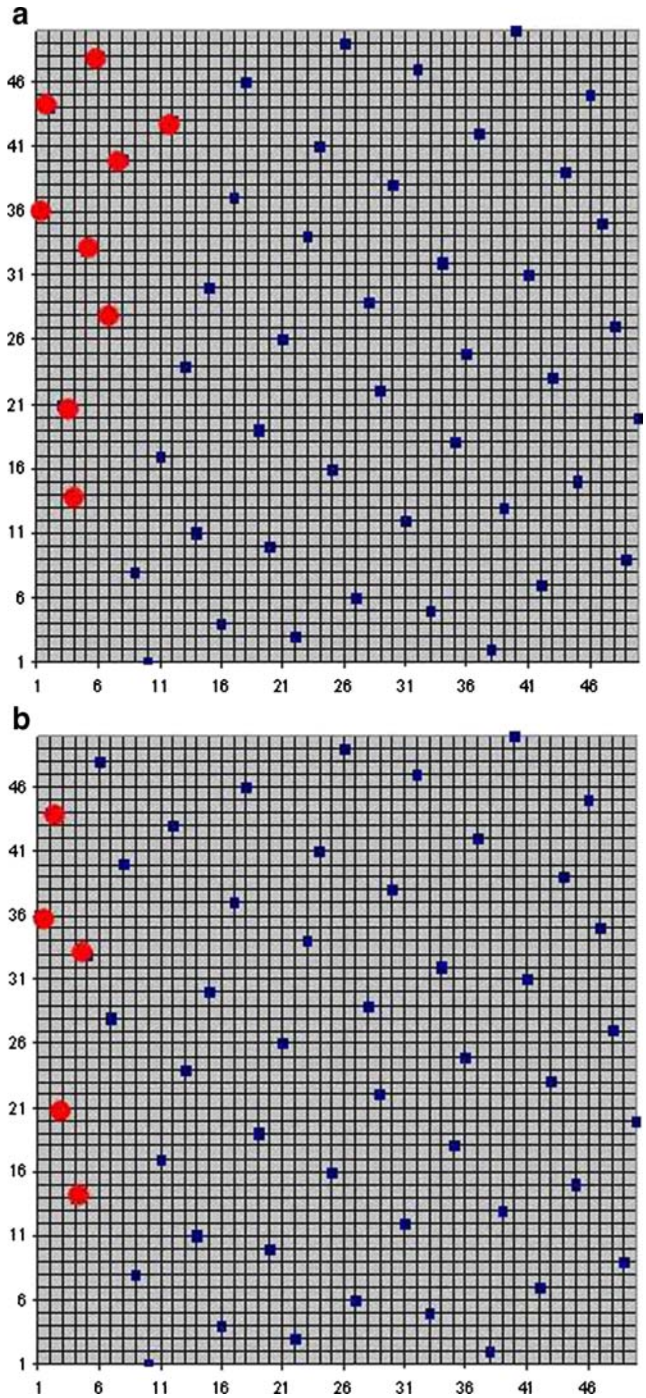

Fig. 2 Characteristics of the system level equality constraints shown by the 50 point uniform Latin Hypercube DoE for the test problem. Larger circular dots correspond to violated constraints and remaining dots indicate that constraints are satisfied

3.2.1 Moving least squares method

The moving least squares method (Wang and Yang 2000; Choi et al. 2001) is a method for metamodel building in design optimization. It can be thought of as a weighted least squares method with weight functions that vary with respect to the location at which the metamodel is evaluated. In the space of all system level variables $\boldsymbol{z}$ that includes $\boldsymbol{s}$, $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ in (1), (2) the

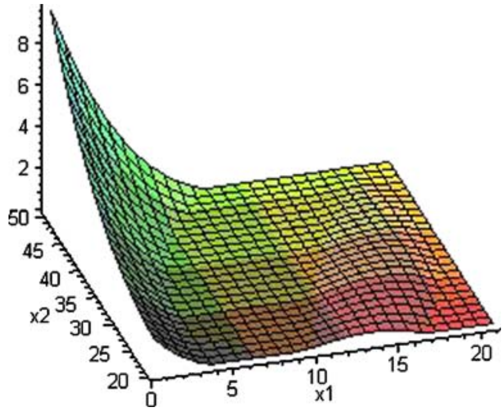

Fig. 3 Cubic metamodel, discipline 1 objective function (with negative values removed)

weight associated with a particular sampling point, z_i, decays as a prediction point z moves away from z_i so the metamodel obtained by the least squares fit is termed a moving least squares metamodel of the original function $F(z)$. Since the weights w_i are functions of z, the polynomial basis function coefficients are also dependent on z. This means that it is not possible to obtain an analytical form of the function $\tilde{F}(z)$ but its evaluation is computationally inexpensive. It is possible to control the "closeness of fit" of the metamodel to the sampling data set by changing a parameter in a weight decay function $w_i(r)$, where r is the distance from the ith sampling point. Such a parameter defines the rate of weight decay or the radius of a sphere beyond which the weight is assumed to be zero (sphere of influence of a sampling point z_i).

There are several types of weight decay function that can be used to control the closeness of fit. Some of these are described by Toropov et al. (2005). The most frequently used function is the Gaussian function, expressed by $w_i = \exp\left(-\theta\, r_i^2\right)$ where θ defines the closeness of fit. The case $\theta = 0$ is equivalent to conventional least squares regression. When θ is large it is possible to obtain a very close fit through the sampling points (approaching interpolation).

4 Numerical example

This section presents a demonstration of the proposed metamodel-based CO framework for solving MDO problems with high fidelity models used in individual disciplines on an intentionally simple problem. It is attempted to mimic some of the essential features of real-life problems such as multidisciplinarity and also potential for the use of high and low fidelity FE models

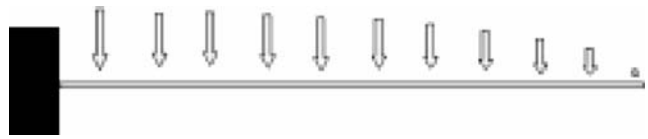

Fig. 4 Beam subjected to a parabolic distributed load $p(\xi)$

in the disciplinary simulation without involving a massive computing effort. This test problem deals with the weight minimization of a continuous fiber-reinforced composite cantilever beam subject to a parabolic distributed load $p(\xi)$ as shown in Fig. 4. The objective function (to be minimized) is the weight of the beam. The design constraints include a limitation on the maximum deflection at the free end of the beam $\delta_{\max}$, a constraint on the maximum bending stress in the beam $\acute{a}_{max}$ and the geometric requirement that the depth of the beam h does not exceed ten times the width w to avoid torsional lateral buckling, as well as the side constraints on the design variables. The design data for this problem are listed in Table 1.

The maximum stress and deflection of the beam can be calculated as follows:

$$\sigma_{\max} = \frac{M_{\max} h}{2I} = \frac{3}{2}\frac{p_0 L^2}{wh^2} = \frac{p_0 L^2 h}{8I} \tag{14}$$

$$\delta_{\max} = \frac{19 p_0 L^4}{360 EI} \tag{15}$$

where the parabolic load $p(\zeta)$ over the beam is defined as:

$$q\,(z) = p_0 \left(1 - \frac{\zeta^2}{L^2}\right) \tag{16}$$

and the second moment of area $I = \frac{wh^2}{12}$. Based on the rule of mixtures for a continuous fiber-reinforced composite material with a fiber volume fraction v_f and a matrix volume fraction v_m, the following relationship must be satisfied:

$$v_{\mathrm{f}} + v_{\mathrm{m}} = 1 \tag{17}$$

Table 1 Design data

Description (notation)	Unit	Value
Parameter in the parabolic distributed load p_0	N/mm	1.0
Length of the beam L	mm	1,000
Elastic modulus, graphite fiber E_{f}	N/mm^2	2.3×10^5
Elastic modulus, epoxy resin E_{m}	N/mm^2	3.45×10^3
Weight density, graphite fiber ρ_{f}	N/mm^3	1.72×10^{-5}
Weight density, epoxy resin ρ_{m}	N/mm^3	1.2×10^{-5}
Stress limit σ^*	N/mm^2	166.667
Displacement limit δ^*	mm	12.9387

The longitudinal (fiber direction) Young's modulus E and the composite weight density, ρ, in terms of the fiber volume fraction using the rule of mixtures can be expressed as:

$$E = E_f v_f + E_m (1 - v_f) \tag{18}$$

$$\rho = \rho_f v_f + \rho_m (1 - v_f) \tag{19}$$

where E_f and E_m are the elastic moduli for graphite fiber and epoxy resin respectively, ρ_f and ρ_m are the weight density values of the graphite fiber and epoxy resin, respectively.

The fiber volume fraction can vary from zero (no fiber) to the maximum value of v_f^{max}. The maximum possible fiber volume fraction using the packing geometry provided in Fig. 5 is calculated as follows:

$$v_f^{max} = \frac{A_{fiber}}{A_{total}} = \frac{3\pi R^2}{6\sqrt{3}R^2} = \frac{\pi}{2\sqrt{3}} = 0.9069. \tag{20}$$

4.1 Conventional optimization problem formulation

The objective is to minimize the weight of a composite cantilever beam. The design variables are shown in the Table 2. The function to be minimized is the weight of the beam:

$$\text{Objective}: \ \text{weight} = AL\rho$$

where $A = \dfrac{12I}{h^2}$ (mm^2), L = 1,000 (mm) and $\rho = \rho_m + v_f(\rho_f - \rho_m)$.

The constraints are:

$$\mathbf{g}_1(I, h) = \frac{\sigma_{max}}{\sigma^*} = \frac{p_0 L^2 h}{8I\sigma^*} \le 1 \tag{21}$$

$$g_2(I, v_f) = \frac{\delta_{max}}{\delta} = \frac{19 q_0 L^4}{360 EI\delta^*} = \frac{19 p_0 L^4}{360 I\left[E_m + v_f\left(E_f - E_m\right)\right]\delta^*} \le 1 \tag{22}$$

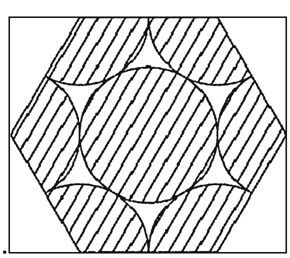

Fig. 5 Geometry of fiber packing in a unit volume

Table 2 Design variables

Description (notation)	Unit	Lower limit	Upper limit
Second moment of area I	mm^4	0.333×10^4	20.833×10^4
Height of the beam h	mm	20	50
Fiber volume fraction v_f		0.4	0.9069

$$g_3(I, h) = \frac{h}{10w} = \frac{h^4}{120I} \le 1. \tag{23}$$

The inequalities (21)–(23) represent, respectively, the constraints on the maximum stress, maximum deflection at the end of the beam and the geometric requirements that the depth of the beam must be equal or greater than ten times of the width of the composite beam. The limiting values of stress and displacement are shown in Table 1, and they correspond to the values for the baseline design. The side constraints on the design variables are obtained by the lower and upper limits, see Table 2.

Conventional (all-at-once) optimization is carried out using a sequential quadratic programming (SQP) algorithm. Results are shown in Table 3.

4.2 Collaborative optimization formulation

The composite beam test problem is now posed to suit the collaborative optimization framework. The test problem is decomposed into two disciplines (the stress-constrained and the deflection-constrained problems) and a system level optimizer to coordinate the overall optimization procedure. The design variables shown in Table 3 are grouped into two disciplines 1 and 2, shown in Table 4.

The system level design variables s_1, s_2 and s_3 represent the second moment of area I, depth of the beam h and the fiber volume fraction v_f, respectively. In the discipline 1 the variable $\boldsymbol{\psi}_{1,1}$ is a local copy of s_1 and $\boldsymbol{\psi}_{1,2}$ is that of s_2, and in the discipline 2 the variable $\boldsymbol{\psi}_{2,1}$ is a local copy of s_1 and $\boldsymbol{\psi}_{2,2}$ is that of s_3. An optimization run at the discipline level finds optimum values of the design variables which satisfy the discipline's own constraints and minimize the discrepancy

Table 3 Results of optimization (All-At-Once) using SQP

Design variables	Unit	Baseline design	Optimum
$I = x_1$	mm^4	2.25×10^4	3.361×10^4
$h = x_2$	mm	30	44.814
$v_f = x_3$		0.785	0.5205
Objective $F(\boldsymbol{x})$	N	4.8246	2.9535
$C_1(\boldsymbol{x})$		1.0	1.0
$C_2(\boldsymbol{x})$		1.0	1.0
$C_3(\boldsymbol{x})$		0.3	1.0

Table 4 Design variables and constraints of the test problem using (CO)

Discipline levels	Design variables	Constraints
Discipline 1	$\psi_{1,1}, \psi_{1,2}$	$g_{1,1} = g_1, g_{1,2} = g_3$
Discipline 2	$\psi_{2,1}, \psi_{2,2}$	$g_{2,1} = g_2$
System level	s_1, s_2, s_3	C_1, C_2

from the target values generated by the system level optimizer. At the system level the constraints are:

$$C_1(\boldsymbol{s}) = 0, \quad C_2(\boldsymbol{s}) = 0 \tag{24}$$

where $C_1(\boldsymbol{s})$ and $C_2(\boldsymbol{s})$ are the discrepancies between the actual and target values returned from discipline 1 and discipline 2, respectively. The system level optimization must satisfy these consistency requirements among disciplines 1 and 2 by enforcing the equality constraints (24).

4.3 Discipline level optimization

This section demonstrates the multi-fidelity modelling strategy in the discipline level optimization using the test problem and focusing on the use of a tuned low fidelity model in place of a high fidelity model in the optimization process. The models used here involve two levels of fidelity: a coarse FE model consisting of 2 beam elements as a low fidelity model and a fine-meshed FE model consisting of 100 beam elements as a high fidelity model. In order to make this simple example mimic an important feature of real-life problems, a commercial FE software ANSYS was used. The tuned low fidelity model is used in optimization. The discrepancy function to be minimized in the first discipline is

$$\left(s_1 - \psi_{1,1}\right)^2 + \left(s_2 - \psi_{1,2}\right)^2. \tag{25}$$

The constraints in discipline 1 are:

$$g_{1,1} = \frac{p_0 L^2 \psi_{1,2}}{8\psi_{1,1}\sigma^*} \leq 1 \tag{26}$$

$$g_{1,2} = \frac{\psi_{1,2}^4}{120\psi_{1,1}} \leq 1 \tag{27}$$

The discrepancy function of discipline 2 can be expressed as:

$$\text{Minimize}: \quad \left(s_1 - \psi_{2,1}\right)^2 + \left(s_3 - \psi_{2,2}\right)^2 \tag{28}$$

The second discipline has one constraint on the maximum deflection of the beam:

$$g_{2,1} = \frac{19 p_0 L^4}{360\psi_{2,1}\left[E_{\mathrm{m}} + \psi_{2,2}\left(E_{\mathrm{f}} - E_{\mathrm{m}}\right)\right]\delta^*} \leq 1. \tag{29}$$

4.4 Implementation of multi-fidelity modelling methodology in CO

The main steps in the multi-fidelity modelling methodology applied to creation of a metamodel utilised in the discipline level optimization in CO are described below, following Zadeh and Toropov (2002).

Step 1 *Choice of a design of experiments*. In this step, the scheme suggested by Audze and Eglais (1977) is used to uniformly plan P points in the design variable space. In order to find what number of points is sufficient for the construction of high quality metamodel models, five separate designs of experiments of ten, five, four, three and two points were studied on the above test problem.

Step 2 *Run low and high fidelity simulation models*. Response values are computed at the selected DoE points using a low fidelity (two beam elements) and a high fidelity (100 beam elements) FE model.

Step 3 *Choice of metamodel function*. In this step, the original high fidelity simulation model is replaced by the tuned low fidelity model and used as a metamodel. Six types of a metamodel (linear and multiplicative of types 1 and 2, quadratic and cubic of type 2) were examined on the above test problem. The selection of the metamodel type was based on the root mean square (RSM) error and the maximum deviation error of the tuned low fidelity model as compared to the high fidelity model at points of another DoE, referred to as a verification DoE.
The five-point plan was selected as an appropriate one for building metamodels for constraints in both disciplines. This was checked against the verification plan. Based on the results the linear metamodel type 1 was selected for the use in the collaborative optimization framework.

Step 4 *Tuning low fidelity model.* Since the approximation of the original high fidelity model $F(\boldsymbol{x})$ by the simplified numerical model $f(\boldsymbol{x})$ is not exact, the discrepancy between them (7) is to be minimized using the tuning parameters $\boldsymbol{a}$.

This procedure was used in the metamodel type selection in the previous step.

Step 5 *Run optimization using corrected low fidelity model*. In this step, the tuned low fidelity model is used in place of the original high fidelity model in the optimization process. In the composite beam test problem the discrepancy between the results obtained using the high fidelity model and the metamodel is negligible.

4.5 System level optimization using MLSM

This section focuses on the system level optimization in the test problem, which is described by:

$$\text{Minimize}: \quad f = \frac{12 s_1}{s_2^2} L\left[\rho_{\mathrm{m}} + s_3\left(\rho_{\mathrm{f}} - \rho_{\mathrm{m}}\right)\right] \tag{30}$$

$$\text{subject to}: \quad C_1(\boldsymbol{s}) = 0, \quad C_2(\boldsymbol{s}) = 0 \tag{31}$$

$$0.333 \times 10^4 \leq s_1 \leq 20.833 \times 10^4,$$
$$20.0 \leq s_2 \leq 50.0, \; 0.4 \leq s_3 \leq 0.9069 \tag{32}$$

where s_1, s_2 and s_3 are system level design variables; $C_1(\boldsymbol{s})$ and $C_2(\boldsymbol{s})$ are the system level equality compatibility constraints. The values of functions $C_1(\boldsymbol{s})$ and $C_2(\boldsymbol{s})$ are produced as a result of the disciplinary optimization runs and can be expensive to evaluate, hence they are replaced by inexpensive metamodels. The main steps are outlined below.

Step 1 *Choice of a design of experiments*. The selection of points in the design variable space is based on a Audze–Eglais uniform Latin hypercube shown in Fig. 2a and b for disciplines 1 and 2, respectively.

Step 2 *Compute the values of compatibility constraints C_1 and C_2*. The values of C_1 and C_2 are computed by the disciplinary optimization runs where targets correspond to the DoE points established in Step 1.

Step 3 *Construct a metamodel*. Compatibility constraint and, if needed, objective function values are used to build global metamodels for all disciplines.

In the moving least squares method there are several parameters such as the order of the base polynomial (e.g. linear, quadratic, etc), the size of the domain of influence and the type of the weight decay function that can be used to control the quality of a metamodel. In this study the Gaussian function, $w_i = \exp\left(-\theta\, r_i^2\right)$, is used as a weight decay function and a quadratic polynomial is used as the base function with $\theta = 10$. Zadeh et al. (2005) investigated on the same test problem the effects of the order of the base polynomial considering linear, quadratic and cubic ones, and also of the value of θ. They concluded that settings mentioned above produced the best quality of the MLSM metamodel. It should be mentioned that these conclusions are not universal, e.g. for a case of a larger number (say, over 10) of design variables even a quadratic base function could lead to an excessive amount of data needed to build a metamodel. Zadeh et al. (2005) demonstrated that a good quality metamodel can be also obtained using a linear base polynomial, this can be a reasonable choice for such a case of a larger number of design variables. As related to the best value of the parameter θ, several measures of the metamodel quality can be used in an optimization process in which θ is treated as a variable. This can be done either on the same data (e.g. using a "leave-K-out" approach, see, e.g. Meckesheimer et al. 2002) or data points from a separate design of experimenrs, see Wang and Shan (2007) and Narayanan et al. (2007). Figure 6 shows the global metamodel obtained for the constraint C_1 related to the discipline 1.

Step 4 *Solve the system level optimization problems*. Metamodels constructed in Step 3 are used in the system level optimization. The process treats response values below 2×10^{-4} (due to the metamodel error which is corrected during the optimization process) as zero. A genetic algorithm (GA) was used to solve the system level optimization in this example.

Step 5 *Trust region strategy*: After solving the optimization problem with global metamodels,

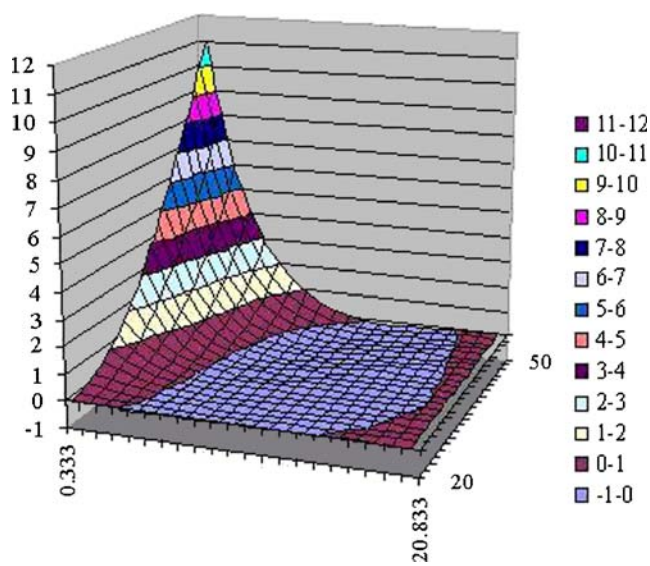

Fig. 6 Global metamodel built by MLSM for constraint C_1 (discipline 1)

Table 5 Results of metamodel-based collaborative optimization and comparison with all-at-once method

Iteration number	Number of plan points used in disciplines 1 and 2		System level						Discipline levels				
			Design variables			Constraints		Objective	Discipline 1			Discipline 2	
	Discipline 1	Discipline 2	s_1 ($\times 10^4$)	s_2	s_3	C_1	C_2	f	Objective	$g_{1,1}$ (g_1)	$g_{1,2}$ (g_3)	Objective	$g_{2,1}$ (g_2)
Global range	50	50	6.771	42.889	0.401	0	0	6.221	0	1.53	1.58	0	1.36
Trust region 1	35	30	2.620	36.577	0.402	0	0	3.312	0	1.0	1.45	0.07	1.0
Trust region 2	39	28	3.314	43.350	0.497	0	0	3.085	0	1.02	1.11	0	1.0
Trust region 3	25	24	3.359	44.819	0.524	0	0	2.955	0	1.0	1.0	0	1.0
All-at-once			3.361	44.814	0.521			2.954		1.0	1.0		1.0

construct a new sub-region (trust region) in the design space, add new DoE points and return to Step 2. This step focuses on the localised search for an optimum solution. The new sub-region is centred on the design point obtained in Step 4, which is resized to be smaller part (here, a half of the range of each of the design variables) of the original size of the design variable domain. The new DoE points are generated in such a way as to ensure the homogeneous distribution of the points inside a search sub-region. The following strategy was adopted: the initial DoE had 50 uniformly spaced points (see Fig. 2a and b) and each subsequent trust region had 20 new DoE points added and used together with existing points from the previous iterations. The new DoE points were added randomly whilst controlling the minimum distance ε from the already planted points. In the test problem above the value of ε is chosen as 5% of the diagonal of a current trust region. If the distance from a new point to any of the existing points is smaller than ε then such a candidate point is abandoned and another random point is chosen. This is repeated until all the new points have been allocated. The optimization problem is solved and checked for convergence of the solution (stop if convergence is obtained, otherwise the trust region allocation process is continued until the optimum solution is reached). The acceptance criteria for convergence include constraint violation and objective function improvement. The result of CO based on the multi-fidelity modelling is shown in Table 5.

Table 6 Evaluation of predictive capabilities of metamodels constructed during CO runs for the system level (discipline 1)

Iterations	RMSE	R-square	RMAE	RAAE
Global meta-model	0.5355	0.9021	1.3235	0.1402
Trust region 1	0.2895	0.9267	0.7582	0.1590
Trust region 2	0.3207	0.9914	0.2255	0.0711
Trust region 3	0.0903	0.9980	0.1173	0.0323

5 Optimization algorithms used in CO

In the discipline level optimization runs various optimization techniques can be used (e.g. SQP) in conjunction with metamodels. At the system level, however, the use of equality constraints to represent the disciplinary feasible regions introduces numerical and computational difficulties that hinder the application of gradient-based optimization algorithms. In this study, three different optimization algorithms SQP, Nelder–Mead (Nelder and Mead 1965) and GA were compared for the system level in the optimization on the above test problem. Due to the special features of the system level optimization discussed above, both SQP and Nelder–Mead algorithms failed to provide a converged solution and, consequently, a more robust optimization algorithm (GA) was used for solving the CO system level optimization. SQP was successfully used for solving the optimization problems in the disciplines 1 and 2.

6 Evaluation of predictive capabilities of the metamodels

The construction of highly accurate metamodels is a requirement for system level optimization within a CO

Table 7 Evaluation of predictive capabilities of metamodels constructed during CO runs for the system level (discipline 2)

Iterations	RMSE	R-square	RMAE	RAAE
Global metamodel model	0.1924	0.7984	2.1062	0.1801
Trust region 1	0.0147	0.9300	0.6985	0.1716
Trust region 2	0.1591	0.9789	0.3612	0.1215
Trust region 3	0.0571	0.9909	0.2858	0.0611

framework and it is therefore important to evaluate the predictive capabilities of such models. An accuracy estimation using several statistical criteria was used in this work. These include root mean square error (RMSE), R-square, relative average absolute error (RAAE) and relative maximum absolute error (RMAE) over DoE points. The larger R-square and smaller RMSE and RAAE values indicate a more accurate metamodel. Note that the parameters of the MLSM-based metamodels were fixed in all runs.

In Tables 6 and 7, R-square values for disciplines 1 and 2 increases from 0.9021 to 0.998 and from 0.7984 to 0.9909, respectively. This is indicative of the fact that as the trust region reduces the accuracy of the metamodel increases.

7 Conclusions

This paper described a metamodel-based CO approach to multidisciplinary optimization with high fidelity simulation models. The obtained results (Table 5) show a high degree of accuracy in discipline level optimization using multi-fidelity modelling. It has been observed that a tuned low fidelity model can have virtually the same accuracy as the high fidelity model. The discrepancy of the optimum solutions using the high fidelity and the tuned low fidelity models is negligible in the test example.

Construction of metamodels poses significant difficulties because of the peculiar characteristics of the system level constraints. The results obtained here show that the MLSM can be used effectively for the construction of metamodels at the system level to capture the behavior of highly non-linear surfaces (in particular the transition from a plateau of zero to non-zero values on the equality constraint surfaces constructed at the system level). MLSM predicted the response with sufficient accuracy and required only four iterations to reach an optimum solution. Good agreement was found between the results obtained using conventional all-at-once optimization and metamodel-based collaborative optimization using high fidelity models for the test problem (Table 5). In terms of reducing computing time, it was observed that with the implementation of metamodels in CO it was reduced from around 10 h (when no metamodels were used at either discipline level or system level) to several minutes of computing time on a typical Windows PC. This is an encouraging result for future applications to complex MDO problems with high fidelity models (e.g. for vehicle crash simulation, aerospace design, etc.).

References

Alexandrov NM, Lewis RM (2000) Analytical and computational properties of distributed approaches to MDO. In: AIAA-2000-4718, Proceedings of the 8th AIAA/NASA/USAF/ISSMO symposium on multidisciplinary analysis and optimization. Long Beach, CA

Alexandrov N, Dennis JE, Lewis RM, Torczon V (1998) A trust region framework for managing the use of approximation models in optimization. Struct Multidisc Optim 15:16–23

Alexandrov NM, Lewis RM, Gumbert CR, Green LL, Newman PA (1999) Optimization with variable-fidelity models applied to wing design. Technical report no. 99-49. ICASE, Institute for Computer Applications in Science and Engineering. NASA Langley Research Center, Hampton, VA

Audze P, Eglais V (1977) New approach for planning out of experiments. Problems of Dynamics and Strengths 35:104–107 (in Russian)

Barthelemy J-FM, Haftka RT (1993) Approximation concepts for optimum structural design—a review. Struct Optim 5(3):129–144

Bates SJ, Sienz J, Toropov VV (2004) Formulation of the Audze–Eglais uniform Latin hypercube design of experiments using a permutation genetic algorithm. In: Proceedings of the 45th AIAA/ASME/ASE/AHS/ASC structural dynamics and materials conference. Palm Springs, CA, 19–22 April 2004

Box GEP, Draper NR (1987) Empirical model-building and response surfaces. Wiley, New York

Braun RD, Gage P, Kroo IM, Sobieski I (1996) Implementation and performance issues in collaborative optimization. In: Proceedings of the 6th AIAA/NASA/USAF/ISSMO symposium on multidisciplinary analysis and optimization. Bellevue, WA

Choi KK, Youn BD, Yang R-J (2001) Moving least square method for reliability-based design optimization. In: Proceedings of the 4th world cong. Structural and multidisciplinary optimization. Dalian

Cramer EJ, Frank PD, Shubin GR, Dennis JE, Lewis RM (1992) On alternative problem formulation for multidisciplinary optimization. In: Proceedings of the 4th AIAA/USAF/NASA/OAI symposium on multidisciplinary analysis and optimization. Cleveland, OH, AIAA 92-4752

Giunta A (1997) Aircraft multidisciplinary optimization using design of experiments theory and response surface modelling methods. Ph.D. thesis, Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA

Giunta AA, Balabanov V, Burgee S, Grossman B, Haftka RT, Mason WH, Watson LT (1995) Variable-complexity multidisciplinary design optimization using parallel computers. In: Alturi SN, Yagawa G, Cruse TA (eds) Computational mechanics '95—theory and applications. Springer, Berlin, pp 489–494

Jang BS, Yang YS, Jung HS, Yeun YS (2005) Managing approximation models in collaborative optimization. Struct Multidisc Optim 30:11–26

Kim HM, Rideout DG, Papalambros PY, Stein JL (2003) Analytical target cascading in automotive vehicle design. J Mech Des 125:481–489

Kodiyalam S, Sobieszczanski-Sobieski J (2001) Multidisciplinary design optimization: some formal methods, framework requirements and application to vehicle design. Int J Vehicle Design 25(special issue):3–32

Kroo I (1995) Decomposition and collaborative optimization for large scale aerospace design. In: Proceedings of the

ICAS/LaRC/SIAM workshop on multidisciplinary optimization. Hampton, VA
Kroo I, Altus S, Braun R, Gage P, Sobieski I (1994) Multidisciplinary optimization method for aircraft preliminary design. In: AIAA-94-4333-CP, Proceedings of the 5th AIAA/NASA/USAF/ISSMO symposium on multidisciplinary analysis and optimization. Panama City, FL, pp 697–707, 7–9 September
Mason WH, Haftka RT, Johnson ER (1994) Analysis and design of composite channel frames. In: Proceedings of the 5th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, vol 2. AIAA Paper 94-4346-CP, Panama City, Florida, pp 1023–1040, September 7–9
McKay MD, Bechman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245
Meckesheimer M, Booker AJ, Barton RR, Simpson TW (2002) Computationally inexpensive metamodel assessment strategies. AIAA J 40(10):2053–2060
Montgomery DC (1997) Design and analysis of experiment. Wiley, New York
Narayanan A, Toropov VV, Wood AS, Campean IF (2007) Simultaneous model building and validation with uniform designs of experiments. Eng Optim 39(5):497–512
Nelder JA, Mead A (1965) A simplex method for function minimization. Comput J 7:308–313
Rodriguez JF, Renaud JE, Watson LT (1997) Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data. In: Proceedings of the 2nd world congress of structural and multidisciplinary optimization. Zakopane, Poland, 26–30 May
Simpson TW, Peplinski J, Mitchell TJ, Allen JK (1997) On the use of statistics in design and the implications for deterministic computer experiment. In: Proceedings of the ASME, DETC97/DTM-3881, design theory and methodology—DTM'97. Sacramento, CA
Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang R-J (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. Struct Optim 27:302–313
Sobieski IP, Kroo IM (1996) Aircraft design using collaborative optimization. In: Proceedings of the aerospace sciences meeting. AIAA Paper 96-0715, Reno, NV, January
Sobieszczanski-Sobieski J (1988) Optimization by decomposition: a step from hierarchic to non-hierarchic systems. In: Proceedings of the 2nd NASA/Air force symposium on recent advances in multidisciplinary analysis and optimization. Hampton, VA, NASA-CP-3031
Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace design optimization: survey of recent developments. Struct Optimiz 14(1):1–23
Sobieszczanski-Sobieski J, Agte J, Sandusky R Jr (1998) Bi-level integrated system synthesis (BLISS). In: Proceedings of the 7th AIAA/USAF/NASA/ISSMO symposium multidisciplinary analysis and optimization (St. Louis, MO). AIAA 98-4916
Toropov VV (2001) Modelling and approximation strategies in optimization-global and mid-range metamodels, response surface methods, genetic programming, and low/high fidelity models. In: Blachut J, Eschenauer HA (eds) Emerging methods for multidisciplinary optimization, CISM courses and lectures, no. 425, international center for mechanical science. Springer, Berlin, pp 205–256
Toropov VV, Markine VL (1996) The use of simplified numerical models as mid-range approximations. In: Proceedings of the 6th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization. pp 952–958
Toropov VV, van der Giessen E (1993) Parameter identification for nonlinear constitutive models: finite element simulation–optimization–nontrivial experiments. In: Pederson P (ed) Optimal design with advanced materials—the Frithiof Niordson volume. Proceedings of IUTAM symposium. Elsevier, Amsterdam, pp 113–130
Toropov VV, Schramm U, Sahai A, Jones RD, Zeguer T (2005) Design optimization and stochastic analysis based on the Moving Least Squares method. In: Proceedings of the 6th world congresses of structural and multidisciplinary optimization. Rio de Janeiro, Brazil, 30 May–3 June
Venkataraman S, Haftka RT, Johnson TF (1988) Design of shell structures for bucking using correction response surface approximations. In: Proceedings of the 7th AIAA/USAF/NASA/ISSMO symposium multidisciplinary analysis and optimization. St. Louis, MO, AIAA 98-4855, September 2–4
Vitali R, Haftka RT, Sankar BV (1998) Correction response surface approximations for stress intensity factors of a composite stiffened plate. In: Proceedings of the 39th AIAA/ASME/ASCE/AHS/ ASC structural dynamics, and materials conference. Long Beach, CA, AIAA-98-2047, pp 2917–2922, 20–23 April
Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. J Mech Des Trans ASME 129(4):370–380
Wang BP, Yang RJ (2000) Structural optimization using moving least squares approximation. In: Proceedings of the 24th national mechanics conference. CYCU, Chungli, Taiwan, 9–10 Dec. 2000
Zadeh PM, Toropov VV (2002) Multi-fidelity multidisciplinary design optimization based on collaborative optimization framework. In: AIAA 2002-5504, Proceedings of the 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization. Atlanta, GA
Zadeh PM, Toropov VV, Wood AS (2004) Use of global approximations in the collaborative optimization framework. In: AIAA 2004-4654, Proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization conference. Albany, NY
Zadeh PM, Toropov VV, Wood AS (2005) Use of MLSM in collaborative optimization. In: Paper 6411, Proceedings of the 6th world congresses of structural and multidisciplinary optimization. Rio de Janeiro, Brazil, 30 May–3 June