Struct Multidisc Optim (2012) 45:515–527
DOI 10.1007/s00158-011-0692-1

RESEARCH PAPER

Mid-range metamodel assembly building based on linear regression for large scale optimization problems

Andrey Polynkin · Vassili V. Toropov

Received: 6 October 2010 / Revised: 29 June 2011 / Accepted: 1 July 2011 / Published online: 9 September 2011

Abstract In this work an approach to building a high accuracy approximation valid in a larger range of design variables is investigated. The approach is based on an assembly of multiple surrogates into a single surrogate using linear regression. The coefficients of the model assembly are not weights of the individual models but tuning parameters determined by the least squares method. The approach was utilized in the Multipoint Approximation Method (MAM) method within the mid-range approximation framework. The developed technique has been tested on several benchmark problems with up to 1000 design variables and constraints. The obtained results show a high degree of accuracy of the built approximations and the efficiency of the technique when applied to large-scale optimization problems.

Keywords Multipoint Approximations Method · Metamodel · Linear regression · Assembly

1 Introduction

This paper is focused on the building of mid-range approximations that has been originated in the work of Haftka et al. (1987) and later developed by Fadel et al. (1990), Wang and Grandhi (1995) as "two-point" approximation methods. This was generalized to multi-point approximations by Toropov (1989), Toropov et al. (1993) and reported by Wang and Grandhi (1996), and Canfield (2004).

The approach undergoes continuous development (van Keulen and Toropov 1997; Polynkin et al. 2008; Shahpar et al. 2008). The objective is to produce better quality approximations that are applicable to large-scale optimization problems.

The present approach is based on the assembly of multiple metamodels. Such an approach was recently studied, for example, by Viana and Haftka (2008) and Acar and Rais-Rohani (2009) where metamodel assembly was based on a weighted sum formulation. In the present work metamodel assembly is built using a linear regression. The regression coefficients of the assembly model are not scaled weights but tuning parameters determined by the least squares method. As a result, the tuning parameters of the assembly model are not restricted to a positive range but may have negative values as well. However, as it will be shown in the paper, for particular cases these parameters may have the meaning of scaled positive weight factors of individual metamodels in the assembly.

The approach was implemented within Multipoint Approximation Method (MAM) based on the mid-range approximation framework. In this paper, the approach was tested on a set of benchmark problems (Svanberg 1987; Fleury 1989; Vanderplaats 1999). The obtained results have shown a high degree of accuracy of the built approximations and the efficiency of the technique when applied to large-scale optimization problems.

2 Outline of Multipoint Approximation Method (MAM)

This technique (Toropov 1989; Toropov et al. 1993) replaces the original optimization problem by a succession of simpler mathematical programming problems. The functions in

A. Polynkin (✉) · V. V. Toropov
University of Leeds, Leeds, LS2 9JT, UK
e-mail: a.polynkin@leeds.ac.uk

V. V. Toropov
e-mail: v.v.toropov@leeds.ac.uk

each iteration are mid-range approximations to the corresponding original functions. The solution of an individual sub-problem becomes the starting point for the next step, the trust region is modified and the optimization is repeated iteratively until the optimum is reached. Each approximation function is defined as a function of design variables as well as a number of tuning parameters. The latter are determined by the weighted least squares surface fitting using the original function values (and their derivatives, when available) at several sampling points of the design variable space. Some of the sampling points are generated in the trust region, and the rest is taken from the extended trust region (as described below).

A general optimization problem can be formulated as

$$\begin{aligned} & min\ F_0(\mathbf{x}),\ F_j(\mathbf{x}) \le 1\ (j = 1, ..., M), \\ & A_i \le x_i \le B_i\ (i = 1, ..., N). \end{aligned} \tag{1}$$

where $\mathbf{x}$ refers to the vector of design variables. The MAM replaces the optimization problem by a sequence of approximate optimization problems:

$$\begin{aligned} & min\ \tilde{F}_0^k(\mathbf{x}),\ \tilde{F}_j^k(\mathbf{x}) \le 1\ (j = 1, ..., M), \\ & A_i^k \le x_i \le B_i^k,\ A_i^k \ge A_i,\ B_i^k \le B_i\ \ (i = 1, ..., N). \end{aligned} \tag{2}$$

where k is the iteration number.

The selection of the approximations $\tilde{F}_j^k(\mathbf{x})\ (j=0, ..., M)$ is such that their evaluation is inexpensive as compared to the evaluation of the original response functions F_j. For example, intrinsically linear functions were successfully used for a variety of design optimization problems in the works of Toropov et al. (1993), van Keulen and Toropov (1997).

The approximations are determined by means of the weighted least squares:

$$min \sum_{p=1}^{P} w_{pj} \left[F_j(\mathbf{x}_p) - \tilde{F}_j^k(\mathbf{x}_p, \mathbf{a}_j) \right]^2. \tag{3}$$

In (3) minimization is carried out with respect to the tuning parameters $\mathbf{a}_j$; w_{pj} are the weight coefficients, and P is the number of sampling points in Design of Experiments (DoE) which must not be less than the number of parameters in the vector $\mathbf{a}_j$.

The weight coefficients w_{pj} strongly influence the difference in the quality of the approximations in different regions of the design variable space. Since in realistic constrained optimization problems the optimum point usually belongs to the boundary of the feasible region, the approximation functions should be more accurate in such domain. Thus, the information at the points located near the boundary of the feasible region is to be treated with greater weights. In a similar manner a larger weight can be allocated to a design with a better objective function, see Toropov et al. (1993), van Keulen and Toropov (1997).

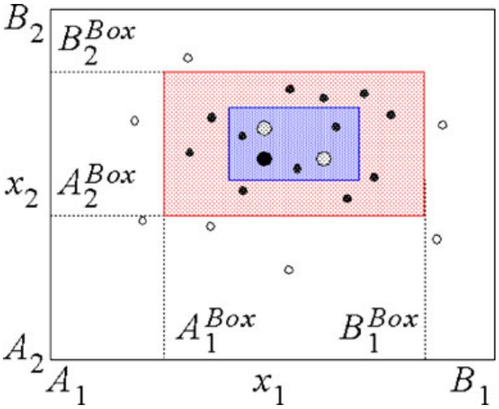

Fig. 1 Current trust region (*smaller box*) and its extension (*larger box*): points outside the *larger* box are not used for building the approximate functions

The approximate functions $\tilde{F}_j^k(\mathbf{x})\ (j = 0, ..., M)$ are intended to be adequate in a current trust region. This is achieved by the appropriate planning of a DoE and definition of a trust region by the side constraints A_i^k and B_i^k. After having solved the approximate optimization problem (2), a new trust region is defined, i.e. its new size and its location are specified. This is done on the basis of a set of parameters that estimate the quality of the approximations ("bad", "reasonable" or "good") and the location of the sub-optimum point in the current trust region. Once the parameters have been determined, the trust region is moved and resized; see van Keulen and Toropov (1997).

As optimization steps are carried out, a database with response function values becomes available. In order to achieve good quality approximations in the current trust region, an appropriate selection of DoE points must be made. Generally, points located far from the current trust region would not contribute to the improvement of the quality of the resulting approximations in the trust region. For this reason only points located in the neighborhood of the current trust region are taken into account, as depicted in Fig. 1. A box in the space of design variables, which is approximately 1.5 to 1.8 times larger than the box representing the current trust region, was found by numerical experimentation to be a reasonable choice for the size of the neighborhood.

3 Design of experiments

In this work, a design of experiments in each trust region is generated randomly. In order to improve the quality of a random DoE, a uniformity constraint (i.e. a constraint on the minimal distance between sampling points) is imposed using the following expression:

$$\frac{dist^p}{Diag} \geq r, \tag{4}$$

where

$$Diag = \sqrt{\sum_{i=1}^{N} \left(B_i^k - A_i^k\right)^2},$$

$$dist^e = \sqrt{\sum_{i=1}^{N} \left(x_i^e - x_i^p\right)^2}; \; e, p = 1, ..., P; e \neq p;$$

In (4) $Diag$ is a characteristic size of a kth trust region (i.e. L^2 distance), $\mathbf{x}^e$ is a new sampling point to be generated, $\mathbf{x}^p$ is a previously generated point, and P is the number of sampling points in a kth trust region.

The parameter r is initially set to 0.95. However if the condition (4) is not satisfied after a prescribed number of randomly generated new points, a value of the threshold ratio r is iteratively reduced, for example, using the following relationship

$$r = r * coeff, \; 0.9 \leq coeff < 1$$

until the constraint (4) is satisfied. It should be noted that the condition (4) is also checked for all sampling points that were generated in the previous trust regions (1,..., $k-1$) and belong to the current trust region.

4 Approximation building

In this work an approach is studied that is based on the assembly of different approximate models $\{\varphi_l\}$ into one metamodel using the following form (note that the indices j and k are suppressed to simplify notation):

$$\tilde{F}(\mathbf{x}) = \sum_{l=1}^{NF} b_l \varphi_l(\mathbf{x}) \tag{5}$$

where NF is the number of regressors in the model pool $\{\varphi_l\}$ and b_l are corresponding regression coefficients.

The use of multiple metamodels has recently been studied, for example, by Viana and Haftka (2008), Viana et al. (2009) and Acar and Rais-Rohani (2009) where coefficients b_l in (5) were treated as weights that reflect the accuracy of the individual surrogates on a set of validation points. Thus, more accurate assembly components φ_l have larger values of the multipliers and vice versa provided that

$$\sum_{l=1}^{NF} b_l = 1. \tag{6}$$

Individual surrogates such as Polynomial Response Surface (PRS), Kriging (KRG), Radial Basis Functions (RBF), Gaussian Process (GP) and Support Vector Regression (SVR) were considered in the above studies.

This work considers an alternative approach to building the expression (5). The idea to use the regression analysis for combining different metamodels instead of calculating the weights for each component was motivated by the early work (Toropov 1989) where the regressors were intended to describe the behavior of separate mechanical (structural) sub-systems. In the present work, as sub-systems we consider individual metamodels.

The proposed procedure consists of two subsequent steps. In the first step, the parameters $\mathbf{a}_l$ of individual functions (regressors) φ_l in (5) are determined by solving a weighted least squares problem using a specified a DoE of P points:

$$min \sum_{p=1}^{P} w_p \left[F\left(\mathbf{x}_p\right) - \varphi_l\left(\mathbf{x}_p, \mathbf{a}_l\right)\right]^2 \tag{7}$$

where minimization is carried out with respect to the tuning parameters $\mathbf{a}_l$.

In the second step, based on the same DoE and keeping the obtained parameters $\mathbf{a}_l$ fixed, a vector $\mathbf{b}$ in (5) is estimated using the following formulation

$$min \sum_{p=1}^{P} w_p \left[F\left(\mathbf{x}_p\right) - \tilde{F}\left(\mathbf{x}_p, \mathbf{b}\right)\right]^2 \tag{8}$$

that leads to solving a linear system of NF equations with NF unknowns b_l where NF is the number of regressors in the model pool $\{\varphi_l\}$. From (8) it follows that the parameters b_l ($l = 1, NF$) in (5) are not conventional weight factors because they are defined as regression coefficients which may get either positive or negative values. The parameters w_p refer to the weights that reflect the inequality of data obtained at different sampling points, $p = 1, P$ (see Section 2).

In order to apply the above two-step regression procedure correctly, it is necessary to satisfy some relations between the number of parameters in vectors $\mathbf{a}_l$ and $\mathbf{b}$, and the number of sampling points P in DoE. The first requirement is that P must not be smaller than the maximum number of tuning parameters contained in either $\mathbf{a}_l$ or $\mathbf{b}$. *(It is worth noting here that in practice we always add a few extra points above the required limit).*

Another essential condition that should be taken into account is the size of the domain (a trust region) where the DoE is built. Too small a size may cause a very similar behavior of the regressors $\{\varphi_l\}$ in the domain (i.e. regressors become almost collinear, Belsley 1991). This may lead to ill-conditioning of the matrix to be inverted in order to obtain a solution of the problem (8). In order to prevent this, in the numerical implementation of the technique a limitation on the smallest size of the trust region is imposed.

The selection of the regressors φ_l is based on the number of sampling points currently located in the trust region. In the mid-range approximation framework that aims at solving large-scale optimization problems, inexpensive (in the sense of the number of sampling points required) approximate models for the objective and constraint functions are built. The simplest case is a linear function of the tuning parameters **a**:

$$\varphi(\mathbf{x}) = a_0 + \sum_{i=1}^{N} a_i x_i . \tag{9}$$

This structure can be extended to an *intrinsically linear* function (Box and Draper 1987). Such functions are non-linear, but they can be led to linear ones by a simple transformation. The most useful function among them is the multiplicative function

$$\varphi(\mathbf{x}) = a_0 \prod_{i=1}^{N} x_i^{a_i} . \tag{10}$$

The advantage of such approximation functions is that a relatively small number ($N+1$, where N is the number of design variables) of tuning parameters a_i is to be determined, this can be done using a relatively small number of DOE points This is the most important feature of such approximations as it allows applying them to large-scale optimization problems.

Other intrinsically linear functions may be considered in the model pool, e.g.

$$\varphi(\mathbf{x}) = a_0 + \sum_{i=1}^{N} a_i / x_i , \tag{11}$$

$$\varphi(\mathbf{x}) = a_0 + \sum_{i=1}^{N} a_i x_i^2 , \tag{12}$$

$$\varphi(\mathbf{x}) = a_0 + \sum_{i=1}^{N} a_i x_i^3 , \tag{12a}$$

$$\varphi(\mathbf{x}) = a_0 + \sum_{i=1}^{N} a_i / x_i^2 . \tag{13}$$

$$\varphi(\mathbf{x}) = a_0 + \sum_{i=1}^{N} a_i / x_i^3 . \tag{13a}$$

As more points are added to the database the approximations may be switched to higher quality models, e.g. a rational model

$$\varphi(\mathbf{x}) = \frac{a_1 + a_2 x_1 + a_3 x_2 + \ldots + a_{n+1} x_n}{1 + a_{n+2} x_1 + a_{n+3} x_2 + \ldots + a_{2n+1} x_n} \tag{14}$$

This type of approximations was studied before by, e.g. Burgee et al. (1994) and Salazar et al. (2007). Due to the rapidly growing number of coefficients for a large N (that is one of the targets of this work), the function structure has to be limited to low degree polynomials, typically linear.

The coefficients in (14) are determined using a least squares approach which reduces to a nonlinear optimization problem with a constraint on the sign of the denominator (positive or negative). The latter is necessary in order to prevent the denominator from crossing the zero axis within a specified trust region. One may note that this formulation may yield the objective function with many local minima. Currently this problem is resolved using optimization restarts from a specified number of initial guesses randomly generated in a trust region.

5 Optimization examples and discussions

The proposed method has been demonstrated on several structural optimization problems. The results obtained for four test cases are presented in order to give an insight into the approach.

5.1 Vanderplaats unconstrained minimization problem

This classic two-dimensional optimization problem, Vanderplaats (1999), with one response function has been chosen mostly for graphical illustration of how the meta-model assembly works.

The objective is to find an equilibrium position of the springs by minimizing the total potential energy of the system

$$\begin{aligned} PE = {} & 0.5^* K_1 \left[\sqrt{x_1^2 + (l_1 - x_2)^2} - l_1\right]^2 \\ & + 0.5^* K_2 \left[\sqrt{x_1^2 + (l_2 + x_2)^2} - l_2\right]^2 \\ & - P_1 x_1 - P_2 x_2 . \end{aligned}$$

The constants K_i are spring stiffnesses, P_i are loads, l_i are the original spring lengths, and x_i are displacements

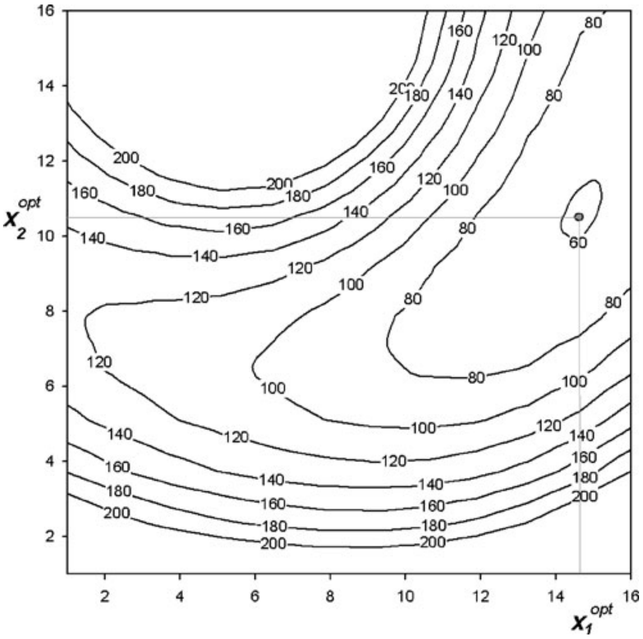

Fig. 2 Two-variable function design space for the spring-force system

where $K_1 = 8$ N/cm, $K_2 = 1$ N/cm, $P_1 = P_2 = 5$ N, $l_1 = l_2 = 10$ cm. The two-variable function space is shown in Fig. 2. In order to consider a positive range of variations for the function and design variables, the following scaling has been applied: $PE = PE + 100$; $x_i = x_i + 6$. The exact minimum of the scaled problem is {14.63; 10.45} cm with $PE = 58.19$ N cm.

Depending on the number of sampling points generated in a trust region, several solutions were obtained. As the initial design, a point {6; 6} cm was used. The initial size of the trust region was 0.25 (i.e. 25% of the search domain). The results are summarised in Table 1. The optima were identified as internal points of trust regions built in a final MAM iteration.

Following the proposed procedure (5)–(8) for building the approximate models, the following five intrinsically linear functions were included in the model pool to solve the optimization problem (Table 2):

Table 1 Optimization results depending on the number of sampling points in the trust region

Sampling points per trust region	x_1^{opt}	x_2^{opt}	PE	Iterations	Calls for function
6	15.0	11.06	58.44	8	49
15	14.77	10.72	58.22	8	121
20	14.65	10.47	58.20	8	161

Table 2 Model pool of five regressors

Regressors order in the pool	Regressor type	Equation number
1	Linear	9
2	Linear in squared variables x_i^2	12
3	Multiplicative	10
4	Linear in reciprocal variables $1/x_i$	11
5	Linear in reciprocal squared variables $1/x_i^2$	13

For illustration, the regression coefficients for the assembly model built in the first and final 8th iteration using 20 sampling points are shown below:

$$b_1 = -5.86 \quad b_2 = 3.87 \quad b_3 = 5.67$$

$$b_4 = -3.40 \quad b_5 = 0.73 \qquad \text{(1st MAM iteration)},$$

$$b_1 = -49603.9 \quad b_2 = 24828.1 \quad b_3 = 63.74$$

$$b_4 = -24755.0 \quad b_5 = 49468.1 \qquad \text{(8th MAM iteration)}.$$

As can be checked, the normalization condition $\sum_{l=1}^{5} b_l = 1$ (6) is implicitly satisfied for the obtained coefficients.

The meaning of the negative coefficients can now be illustrated. Figures 3 and 4 show how the technique defines different (positive and negative) slopes for monotonic functions ϕ_l from the model pool, that are monotonic in the trust region, in order to assemble an adequate approximation with non-monotonic behaviour.

In order to compare the accuracy of different components φ_l with the performance of the assembled model $\tilde{F}$, *RMSE* (root mean squared error) of scaled response values was calculated

$$RMSE = \sqrt{\frac{1}{K_{\text{test}}} \sum_{i=1}^{K_{\text{test}}} \left(\frac{\tilde{F}_i - F_i}{F_i} \right)^2} \qquad (15)$$

where K_{test} is number of validation points randomly generated in the trust region; $\tilde{F}_i$ and F_i are model and actual function values at validation points. For this case 500 validation points were generated.

Values of scaled *RMSE* depending on the number of sampling points are given in Tables 3 and 4.

As can be observed, the accuracy of the assembled model is always higher than the accuracy of its individual

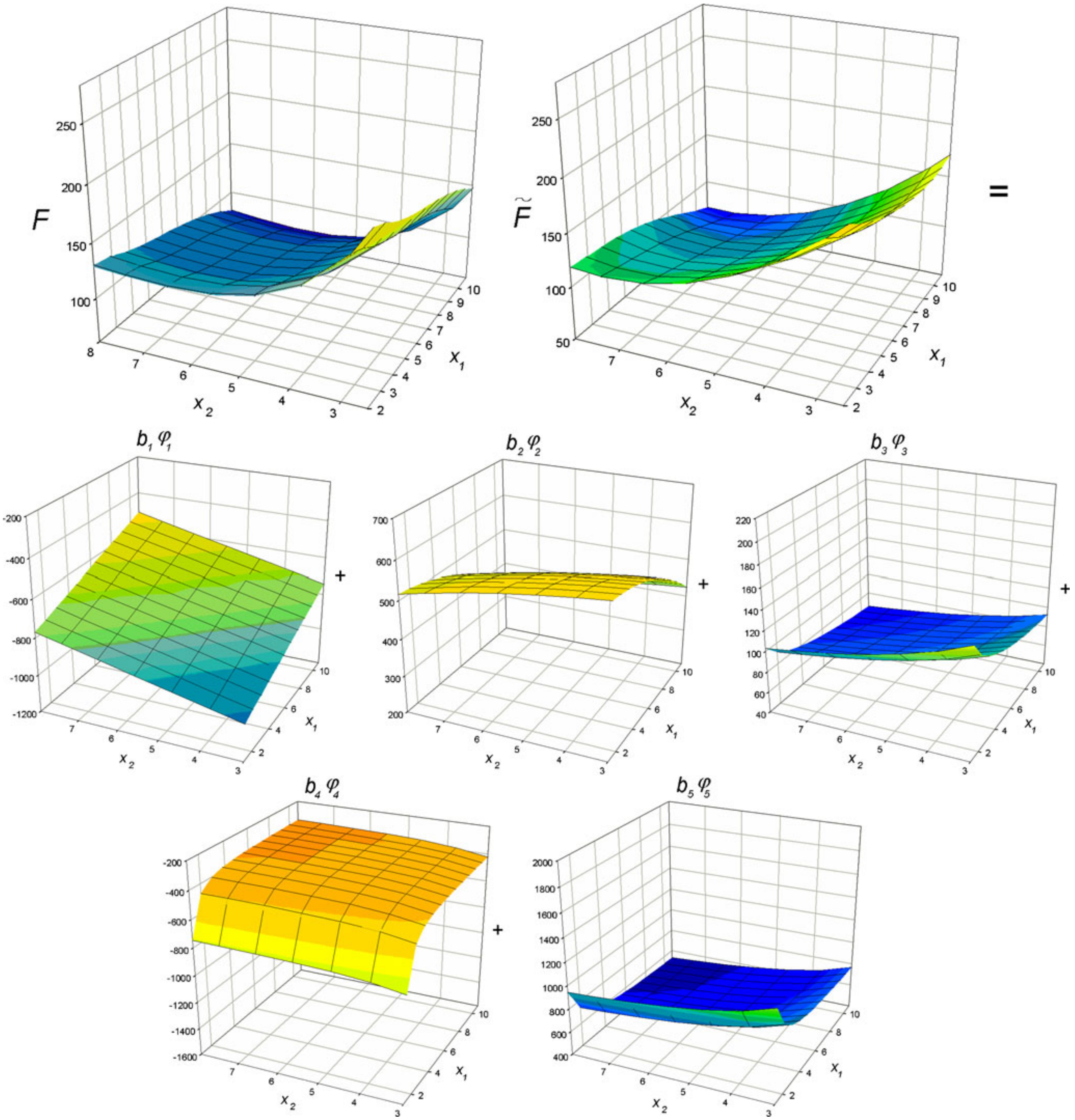

Fig. 3 Actual function and metamodel assembly built of five regressors in the first MAM iteration

components. This trend was found in all iterations using different numbers of sampling points generated in a trust region.

It should be *reminded* that the size of the *trust* regions generated during the MAM optimization search is gradually decreased. This explains why *RMSE* obtained in the 8th (last) iteration is much smaller than *RMSE* corresponding to the 1st (initial) iteration.

5.2 Two-bar truss

The problem is illustrated in Fig. 5. A two-bar truss (Svanberg 1987) is loaded by a force with corresponding components $F_x = 24.8$ kN and $F_y = 198.4$ kN. There are two design variables: the cross-section area of both bars x_1 cm^2 and half of the distance between the supports x_2 m. The objective function is the weight of the structure and

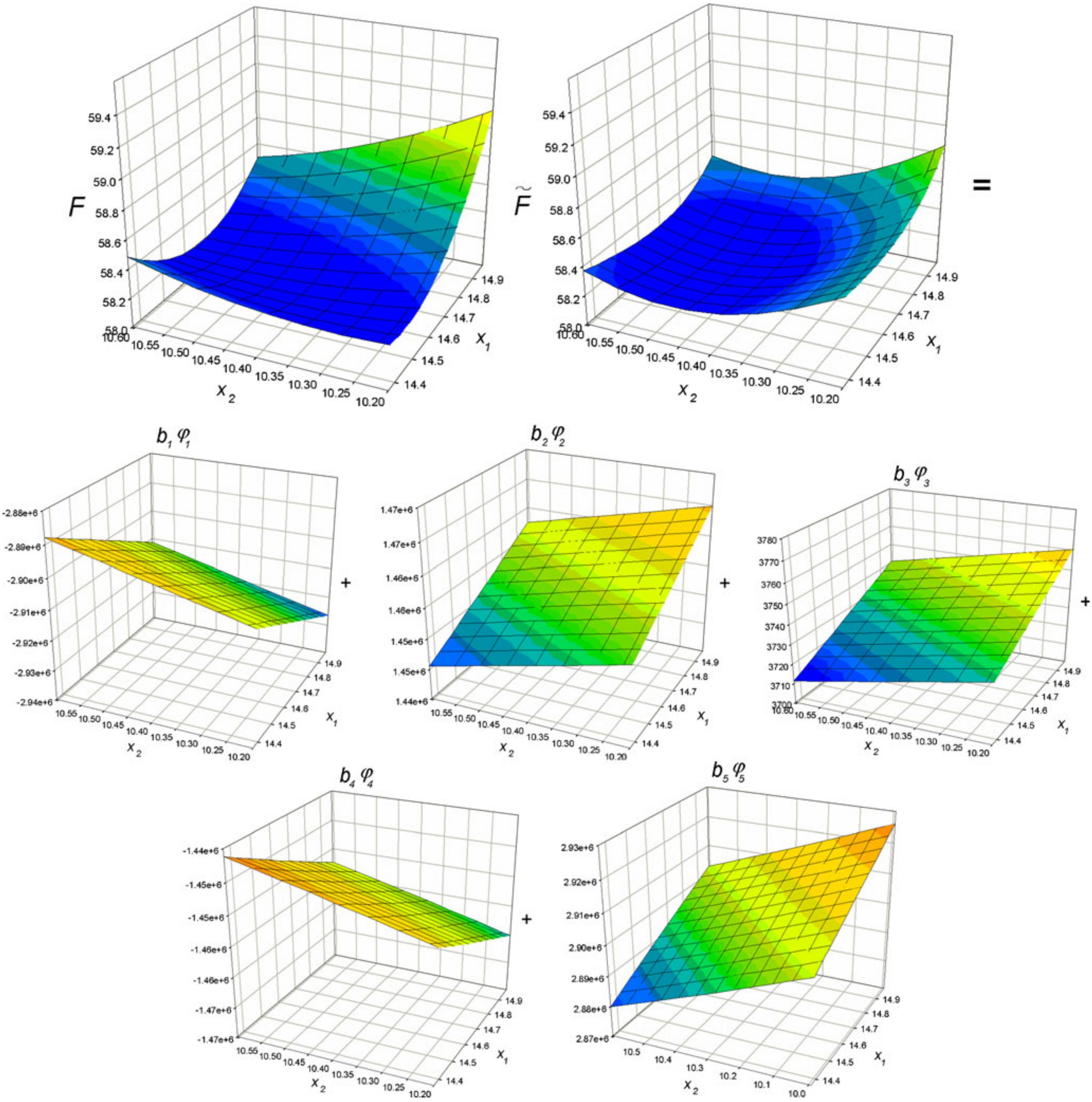

Fig. 4 Actual function and metamodel assembly built of five regressors in the 8th MAM iteration

constraints are stresses (in both bars) which must not exceed 100 N/mm^2. The functions have analytical expressions:

$$F_0\left(\mathbf{x}\right) = c_1 x_1 \sqrt{\left(1 + x_2^2\right)},$$

$$F_1\left(\mathbf{x}\right) = c_2 \sqrt{\left(1 + x_2^2\right)} \left(\frac{8}{x_1} + \frac{1}{x_1 x_2}\right) \leq 1,$$

$$F_2\left(\mathbf{x}\right) = c_2 \sqrt{\left(1 + x_2^2\right)} \left(\frac{8}{x_1} - \frac{1}{x_1 x_2}\right) \leq 1,$$

where $c_1 = 1.0$, $c_2 = 0.124$.

The side constrains are defined by $A_1 = 0.2$; $A_2 = 0.1$; $B_1 = 4.0$; $B_2 = 1.6$. The starting guess is $\{2.5; 1.0\}$.

The problem has been tested by several authors using different approximation techniques (see Svanberg 1987). In our earlier work (Toropov et al. 1993), this test has

Table 3 *RMSE* for metamodel assembly and individual regressors in the 1st MAM iteration

Sampling (building) points per trust region	$\tilde{F}$	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
6	0.18	0.21	0.22	0.19	0.27	0.19
15	0.12	0.21	0.23	0.18	0.19	0.17
20	0.11	0.22	0.23	0.18	0.19	0.17
30	0.12	0.21	0.22	0.18	0.19	0.17
40	0.11	0.21	0.23	0.18	0.19	0.17

been successfully solved using MAM based on one type of approximations, namely the multiplicative function (10).

The purpose of this simple test in the present work is to show the validity of the proposed assembly approach when several types of regressors including non-linear regressor (14) are involved.

For building the approximations, the following set of seven functions in the model pool was used (Table 5):

One may notice that the above functions individually may describe the global behavior rather poorly. However such approximations can be efficient in the mid-range approximation framework of MAM.

The problem has been solved in 4 MAM iterations using 7 sampling points generated in each trust region yielding 29 numerical experiments in total. The obtained solution is given by the vector {1.4132; 0.3736} with the objective $F_0 = 1.50860$ and the constraint $F_1 = 1.00006$. For comparison, the optimum obtained by SQP with a specified tolerance 10^{-10} (after 15 calls for a procedure that calculates functions and its derivatives values) is {1.4119; 0.3766} with the objective $F_0 = 1.50865$ and the constraint $F_1 = 1.00000$.

The way the regression coefficients $\mathbf{a}_i$ $(i = 1, 7)$ and $\mathbf{b}$ of the active constraint F_1 have been evolving during the trust regions move and reduction towards the optimum is shown in the Tables 6, 7, 8 and 9. Each table corresponds to a subsequent MAM iteration. It is worth mentioning here that the optimum has actually been found in the third iteration (as an internal point of the trust region). In the forth iteration, a trust region just shrank around the obtained point leaving the solution essentially the same.

Table 4 *RMSE* for metamodel assembly and individual regressors in the 8th MAM iteration

Sampling (building) points per trust region	$\tilde{F}$	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
6	3.00e–3	6.28e–3	6.23e–3	6.23e–3	6.32e–3	6.38e–3
15	3.41e–3	3.88e–3	3.86e–3	3.88e–3	3.92e–3	3.90e–3
20	3.25e–3	4.57e–3	4.58e–3	4.54e–3	4.54e–3	4.56e–3
30	3.24e–3	3.56e–3	3.58e–3	3.53e–3	3.53e–3	3.54e–3
40	1.02e–3	1.97e–3	1.96e–3	1.97e–3	1.99e–3	1.99e–3

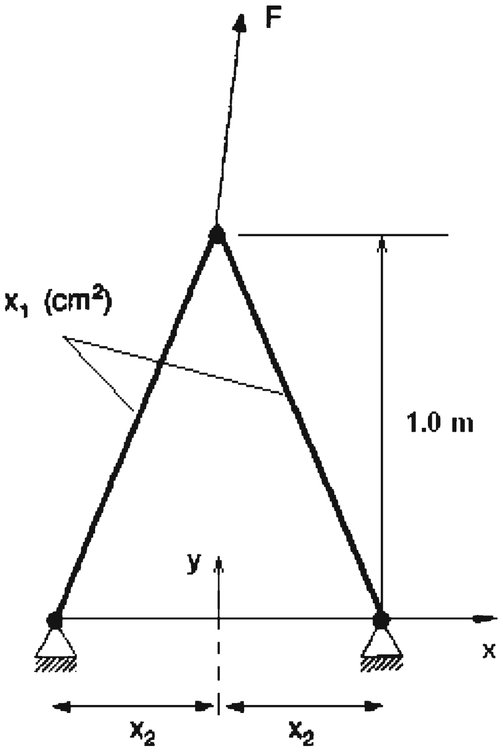

Fig. 5 Two-bar truss

Again, the normalization condition $\sum_{l=1}^{7} b_l = 1$ is satisfied for the obtained coefficients in all iterations.

Note that a non-linear regressor (rational function) has been involved in this example for the illustration purpose

Table 5 Model pool of seven regressors

Regressors order in the pool	Regressor type	Equation number
1	Linear	9
2	Linear in squared variables x_i^2	12
3	Linear in cubed variables x_i^3	12a
4	Linear in reciprocal variables $1/x_i$	11
5	Linear in reciprocal squared variables $1/x_i^2$	13
6	Linear in reciprocal cubed variables $1/x_i^3$	13a
7	Rational	14

Table 6 Parameters of vectors $\mathbf{a}_i$ and $\mathbf{b}$ in the first MAM iteration

$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{b}$
2.9734	2.1097	1.6884	0.3509	0.9390	1.0759	28.296	9.2208
−1.1133	−0.3354	−0.1096	1.4769	0.6256	0.3063	−1.898e–2	−9.8751
−8.969e–2	−6.268e–2	9.082e–3	−0.2521	−0.1425	−7.977e–2	10.501	3.7884
						24.543	−7.8242
						−1.3766	9.1385
							−3.4721
							2.364e–2

Table 7 Parameters of vectors $\mathbf{a}_i$ and $\mathbf{b}$ in the 2nd MAM iteration

$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{b}$
1.8915	1.4207	1.2646	5.460e–3	0.4771	0.6346	315390.2	457.51
−0.6328	−0.2067	−8.809e–2	1.3873	0.9959	0.9361	−27131.1	−395.28
8.450e–3	9.487e–4	−1.645e–2	7.881e–6	9.916e–4	7.706e–4	88386.2	107.50
						200709.3	−398.80
						83016.7	299.38
							−70.01
							0.7040

Table 8 Parameters of vectors $\mathbf{a}_i$ and $\mathbf{b}$ in the 3rd MAM iteration

$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{b}$
2.0917	1.5426	1.3563	−7.685e–2	0.4710	0.6555	44.9334	−333.58
−0.7040	−0.2404	−0.1065	1.4002	0.9539	0.8485	−4.7065	300.57
−0.1860	−0.1963	−0.2648	3.554e–2	7.123e–3	1.805e–3	7.0259	−84.837
						24.1301	261.22
						14.1623	−180.99
							38.349
							0.2611

Table 9 Parameters of vectors $\mathbf{a}_i$ and $\mathbf{b}$ in the final 4th MAM iteration

$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{b}$
2.1215	1.5642	1.3782	−0.1059	0.4513	0.6373	5.7237	−5.3693
−0.7128	−0.2538	−0.1203	1.4001	0.9792	0.9119	−1.943e–2	28.332
−0.3023	−0.3982	−0.6969	4.307e–2	8.079e–3	2.011e–3	−3.6111	−13.979
						3.0019	−39.241
						−2.3976	46.501
							−14.068
							−1.1757

only (as this problem can efficiently be solved without it). In practice, e.g. for large-scale problems, we found it quite expensive each time to determine coefficients of the rational function in each iteration as this involves solving a nonlinear optimization problem with multiple restarts.

On this problem we would like to demonstrate the difference between our two step regression approach and the one-shot nonlinear regression that can be considered as a possible (although inconvenient for the reasons mentioned above) alternative. The results below correspond to the metamodels built for the constraint function F_1 on a 40 point DoE in the first MAM iteration.

In the case of a one-shot nonlinear regression, the approximation model includes 6 intrinsically linear and 1 rational functions (Table 5) that gives 30 coefficients A_i to be defined by solving an optimization problem with multiple restarts to allow for the presence of many local minima:

$$\begin{aligned}
\varphi_1 &= \mathrm{A}_1 + \mathrm{A}_2 x_1 + \mathrm{A}_3 x_2\\
\varphi_2 &= \mathrm{A}_4 + \mathrm{A}_5 x_1^2 + \mathrm{A}_6 x_2^2\\
\varphi_3 &= \mathrm{A}_7 + \mathrm{A}_8 x_1^3 + \mathrm{A}_9 x_2^3\\
\varphi_4 &= \mathrm{A}_{10} + \mathrm{A}_{11}/x_1 + \mathrm{A}_{12}/x_2\\
\varphi_5 &= \mathrm{A}_{13} + \mathrm{A}_{14}/x_1^2 + \mathrm{A}_{15}/x_2^2\\
\varphi_6 &= \mathrm{A}_{16} + \mathrm{A}_{17}/x_1^3 + \mathrm{A}_{18}/x_2^3\\
\varphi_7 &= (\mathrm{A}_{19} + \mathrm{A}_{20} x_1 + \mathrm{A}_{21} x_2)\\
&\quad / \, (1.0 + \mathrm{A}_{22} x_1 + \mathrm{A}_{23} x_2)\\
\tilde{F} &= \mathrm{A}_{24}\varphi_1 + \mathrm{A}_{25}\varphi_2 + \mathrm{A}_{26}\varphi_3 + \mathrm{A}_{27}\varphi_4\\
&\quad + \mathrm{A}_{28}\varphi_5 + \mathrm{A}_{29}\varphi_6 + \mathrm{A}_{30}\varphi_7 \qquad (16)
\end{aligned}$$

In the case of the metamodel assembly building, the corresponding vectors $\mathbf{a}_i$ and $\mathbf{b}$ are as follows:

$$\begin{aligned}
\mathbf{a}_1 &= \{\mathrm{A}_1; \mathrm{A}_2; \mathrm{A}_3\}\\
\mathbf{a}_2 &= \{\mathrm{A}_4; \mathrm{A}_5; \mathrm{A}_6\}\\
\mathbf{a}_3 &= \{\mathrm{A}_7; \mathrm{A}_8; \mathrm{A}_9\}\\
\mathbf{a}_4 &= \{\mathrm{A}_{10}; \mathrm{A}_{11}; \mathrm{A}_{12}\}\\
\mathbf{a}_5 &= \{\mathrm{A}_{13}; \mathrm{A}_{14}; \mathrm{A}_{15}\}\\
\mathbf{a}_6 &= \{\mathrm{A}_{16}; \mathrm{A}_{17}; \mathrm{A}_{18}\}\\
\mathbf{a}_7 &= \{\mathrm{A}_{19}; \mathrm{A}_{20}; \mathrm{A}_{21}; \mathrm{A}_{22}; \mathrm{A}_{23}\}\\
\mathbf{b} &= \{\mathrm{A}_{24}; \mathrm{A}_{25}; \mathrm{A}_{26}; \mathrm{A}_{27}; \mathrm{A}_{28}; \mathrm{A}_{29}; \mathrm{A}_{30}\}
\end{aligned}$$

These two ways of formulating the metamodel building problem result in two completely different sets of model coefficients (Table 10). The quality of the metamodels has been assessed by the *RMSE* values (15) calculated on the DOE points, these are 0.00469 for the one-shot of non-linear regression and 0.0136 for the metamodel assembly. The former is more accurate whereas the latter is much simpler and more suitable for large-scale optimization problems.

Table 10 Coefficients of the non-linear regression (16) estimated in one "shot" (i.e. obtained by minimization of sum of squares) in comparison with the corresponding coefficients obtained by the assembly approach

	Non-linear regression	The assembly
1	2.396457	3.0280750
2	4.449486	−1.3956280
3	−3.382556	0.6016999
4	7.693423	2.3517557
5	0.22267E-02	−0.3902567
6	0.223307	0.2588174
7	3.422376	2.1280342
8	0.902226	−0.1279212
9	0.369356	0.1334303
10	8.080112	0.4558186
11	−17.26829	1.5898862
12	1.148209	−0.4103339
13	2.224118	1.0855063
14	0.987671	0.4156419
15	0.836704	−0.1145523
16	3.297245	1.2604212
17	7.653476	0.1110441
18	0.866966	−4.120782E-002
19	−0.746787	203.88510
20	5.047989	0.8537122
21	−0.224718	108.19648
22	9.903058	195.30600
23	−2.265703	−1.0588928
24	−0.925751	−4.8314931
25	−0.589809	5.9631825
26	1.320733	−2.3860610
27	−0.97340E-01	4.0274174
28	0.71407E-01	−5.0225619
29	−0.74234E-03	2.2107057
30	5.794658	1.039240

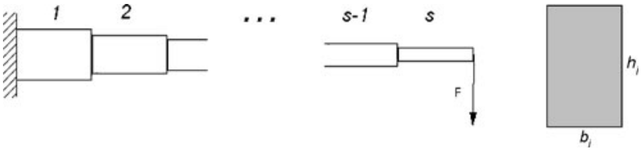

Fig. 6 Scalable beam with rectangular cross sections

5.3 Vanderplaats scalable beam

The problem is formulated as follows: minimize the volume of a cantilever beam (Fig. 6)

$$V = \sum_{i=1}^{S} b_i h_i l_i$$

under stress, aspect ratio, and tip deflection constraints

$$\sigma_i/\bar{\sigma} \le 1;\ \ h_i/(20b_i) \le 1;\ \ y_S/\bar{y} \le 1;$$

with the lower limits on the cross section size

$$b_i \ge 1; h_i \ge 5;\ \ (i = 1, S)\,.$$

The properties of the beam are as given by Vanderplaats (1999): $\bar{\sigma} = 14{,}000$ N/cm^2; $\bar{y} = 0.5$ cm; F = 50,000 N; E = 200 GPa; total length L = 500 cm.

Three cases were considered depending on the number of elements in the beam: (a) $S = 5$ resulting in $N = 10$ design variables and 11 constraints; (b) $S = 50$ resulting in $N = 100$ and 101 constraints; (c) $S = 500$ resulting in $N = 1000$ and 1001 constraints.

For building the approximate models, a set of five intrinsically linear functions specified in Table 2 has been used.

It was found that a multiplicative function (10) was given a preference in the model building for the stress and aspect ratio constraints. As an example, the coefficients b_i ($i = 1 ... 5$) obtained for the stress constraint in the first MAM iteration are shown below:

$$b_1 = -0.86\text{E-}05 \quad b_2 = -0.12\text{E-}04 \quad b_3 = 1.00003$$
$$b_4 = 0.32\text{E-}04 \qquad b_5 = -0.61\text{E-}04$$

As can be seen, all the parameters except b_3 are almost zeros. This means that the algorithm implicitly selects the most suitable model from the pool for a function whose behaviour is closely described by that model.

In contrast to the stress and aspect ratio constraints, the models for the objective function and displacement constraint have non-zero coefficients b_i for all the available regressors in the model pool. An example of coefficients determined for the displacement approximation during the optimization search is given below:

$$b_1 = -0.119 \quad b_2 = 0.15\text{E-}01 \quad b_3 = 0.453$$
$$b_4 = 0.207 \quad b_5 = 0.444 \qquad \left(\sum_{l=1}^{5} b_l = 1\right)$$

MAM's optimization result is $V = 61914.79$ cm^3 with a corresponding vector of design variables {2.992; 2.778; 2.524; 2.205; 1.750; 59.840; 55.551; 50.471; 44.091; 34.995} cm, where the first 5 parameters are element widths, and the rest of them (from 6 to 10) are element heights. All stress and aspect ratio constraints are active at the optimal point except the displacement constraint which is inactive. For comparison, Vanderplaats's solution for this case obtained using the exterior penalty function method is $V = 66169$ cm^3 with a vector {3.24; 2.90; 2.52; 2.26; 2.24; 56.77; 53.81; 50.30; 44.87; 41.71} cm.

The optimization results obtained for all three cases (a–c) using the MAM method are summarized in Fig. 7. The number of DoE points p generated in a trust region is $N + 1$ in each case. The total number of calls for analysis after three MAM iterations is 34, 304, and 3004, respectively (that is p multiplied by the number of iterations plus a starting point). The optimal values of the objective function corresponding to $N = 100$ and $N = 1000$ cases are 54590.65 cm^3 and 53803.01 cm^3.

The solutions obtained by SQP method (directly applied) for this example are quite identical to the above results. For $N = 10$, the optimal value of the objective function 61914.7890 cm^3 was obtained after 16 calls for the subroutine that calculated the functions values and its derivatives (using finite differences). A corresponding vector of design variables is {2.99204240; 2.77756612; 2.52358629; 2.20455569; 1.74975701; 59.8408480; 55.5513224; 50.4717259; 44.0911138; 34.99514023} cm. For $N = 100$, the optimal value of the objective function 54590.6536 cm^3 was obtained also after 16 calls for the subroutine.

5.4 A cantilever scalable thin-wall beam

In this test, a cantilever beam is built up of S elements with hollow square cross sections. The objective function is the weight of the beam that has to be minimized. There is a constraint imposed on the tip displacement. The design

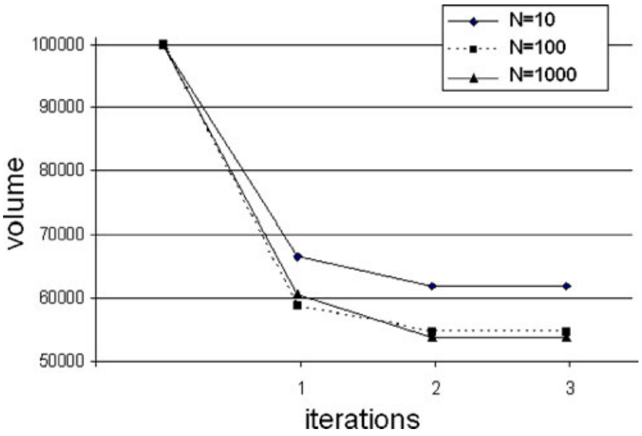

Fig. 7 Convergence plots for the optimization cases with $N = 10$, $N = 100$, and $N = 1000$ design parameters

variables are heights (widths) of the square cross sections, Fig. 8.

Based on the discretization of five elements the optimization problem was formulated by Svanberg (1987) in a closed form:

$$minimize\ F_0(\mathbf{x}) = 0.0624 \sum_{i=1}^{5} x_i$$

subject to

$$F_1(\mathbf{x}) = 61/x_1^3 + 37/x_2^3 + 19/x_3^3 + 7/x_4^3 + 1/x_5^3 \leq 1$$

with a feasible starting point $x_i = 5$ $(i = 1...5)$.

In order to solve the problem, the same set of 5 regressors (Table 2) was used as in the previous test case.

An example of the regression coefficients obtained in a MAM iteration for the displacement constraint is typed below

$$b_1 = 1.828 \quad b_2 = -0.746 \quad b_3 = 0.15\text{E-}02$$
$$b_4 = 3.419 \quad b_5 = -3.515 \quad \left(\sum_{l=1}^{5} b_l = 1\right)$$

For the objective function the algorithm always determined $b_1 = 1$ and $b_i = 0$ $(i = 2...5)$. The solution was obtained after 5 MAM iterations and 31 function evaluations (based on a 6 point DoE generated in each iteration). The optimum point is {6.015; 5.309; 4.493; 3.502; 2.152}. The corresponding value of the objective function F_0 is 1.339. For the reference, the analytical solution of this problem is {6.016; 5.309; 4.494; 3.502; 2.153}, $F_0 = 1.34$ (Svanberg 1987).

It is worth noting that this problem seemed to be rather difficult for solving by approximation techniques. For instance, Svanberg's MMA method converged to $F_0 = 1.34$ after four iterations after some preliminary tuning, while Fleury's CONLIN optimizer didn't converge at all. Using an earlier version of Toropov's MAM (Toropov et al. 1993) with a multiplicative approximation (as a default type) for the constraint function, a solution {6.02; 5.53; 4.75; 3.14; 2.03} with $F_0 = 1.34$ but with a violated constraint $F_1 = 1.01$ was achieved after 17 iterations (103 function evaluations) that we consider as an unsatisfactory result.

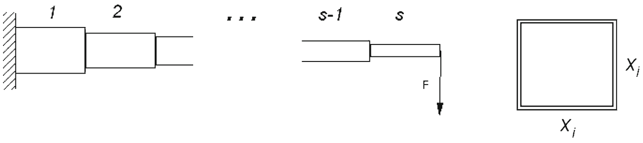

Fig. 8 Scalable beam with hollow square cross-sections

In order to verify the performance of the algorithm on a large-scale level, the problem was extended to 100 and 500 beam elements resulting in $N = 100$ and $N = 500$ design variables. The metamodels were built using 105 ($N = 100$) and 550 ($N = 500$) point DoEs generated in each MAM iteration. The corresponding solutions are shown in Fig. 9. The optimal values of the objective function are 1.3107 ($N = 100$) and 1.3101 ($N = 500$).

Note that including a regressor $\varphi(\mathbf{x}) = a_0 + \sum_{i=1}^{N} a_i/x_i^3$ (13a) in the model pool can considerably improve the performance of the algorithm as a solution of the problem may actually require just one MAM iteration. This is because the algorithm in this case will build approximations that are nearly identical to the expressions for the objective and constraint functions. This however assumes that MAM may still perform a few further iterations shrinking a trust region around the solution obtained in the first iteration until a minimum size of the trust region is reached (a criterion which is used to stop MAM optimization run). The lines below present coefficients of vectors $\mathbf{a}_6$ and $\mathbf{b}$ of the approximation of the constraint F_1 obtained in the first MAM iteration for $N = 5$ and using DoE of 10 points:

$$\mathbf{a}_6 = \{1.5289\text{E-}009; 60.9999; 36.9999; 18.9999; 6.9999; 0.9999\},$$
$$\mathbf{b} = \{-0.4450\text{E-}07; 0.3268\text{E-}07; -0.5265\text{E-}08; 0.5420\text{E-}07; -0.5188\text{E-}07; 1.0000\}.$$

In the above lines the last 6th parameter of the vector $\mathbf{b}$ corresponds to the additionally introduced regressor.

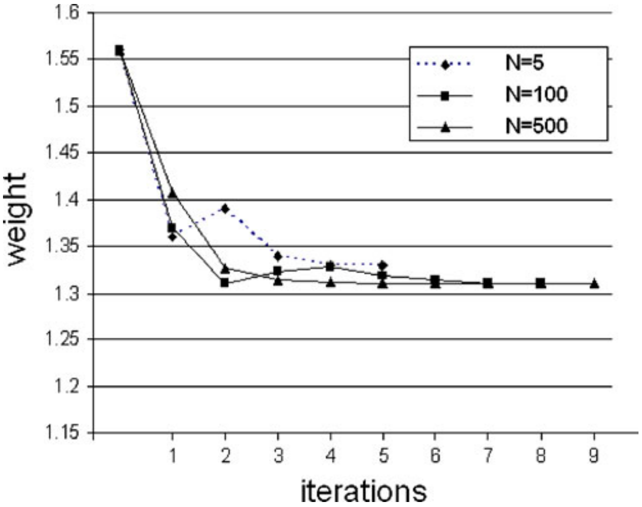

Fig. 9 Convergence plots for the optimization cases with $N = 5$, $N = 100$, and $N = 500$ design parameters. The number of analysis for each case is 31, 841, and 4951, respectively

6 Conclusions

This paper presented an approach for building the approximate functions based on a metamodel assembly. The novelty of the approach is that the construction of the metamodel assembly is based on a linear regression. The parameters of the metamodel assembly are not scaled positive weight factors reflecting the accuracy of the individual components but regression coefficients obtained by the least squares method. In this way, the parameters of the metamodel assembly may get both positive and negative values. As was illustrated, the different signs of the parameters may be interpreted as the different slopes of monotonic functions defined to produce non-monotonic behaviour. It has also been shown that in the particular cases the approach may yield the regression coefficients with the values of conventional weight factors (i.e. scaled to the range [0,1]).

The approach was utilized in the Multipoint Approximation Method (MAM) method within the mid-range approximation framework. The results obtained in the paper show that the technique is economical in calls for function evaluations and is capable of solving optimization problems with a large number of design variables.

Acknowledgments The authors are grateful to the UK Department of Trade and Industry, Rolls-Royce plc and Airbus UK Ltd for the support of this work within the CFMS R&D programme.

References

Acar E, Rais-Rohani M (2009) Ensemble of metamodels with optimized weight factors. Struct Multidisc Optim 37:279–294

Belsley DA (1991) Conditioning diagnostics: collinearity and weak data in regression. Wiley, New York

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

Burgee SL, Watson LT, Giunta AA, Grossman B, Haftka RT, Mason WH (1994) Parallel multipoint variable-complexity approximations for multidisciplinary optimization. In: Proceedings of the IEEE scalable high-performance computing conference, June 1994, pp 734–740

Canfield RA (2004) Multipoint cubic surrogate function for sequential approximate optimization. Struct Multidisc Optim 27:326–336

Fadel GM, Riley MF, Barthelemy JM (1990) Two point exponential approximation method for structural optimization. Struct Optim 2:117–124

Fleury C (1989) CONLIN: an efficient dual optimizer based on convex approximation concepts. Struct Optim 1:81–89

Haftka RT, Nachlas JA, Watson LT, Rizzo T, Desai R (1987) Two-point constraint approximation in structural optimization. Comp Meth Appl Mech Eng 60:289–301

Polynkin A, Toropov V, Shaphar S (2008) Adaptive and parallel capabilities in the Multipoint Approximation Method. In: AIAA 2008-5803, Proceeding of 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. Proceedings. AIAA/ ISSMO Multidisciplinary Analysis and Optimization Conference. Victoria, British Columbia, Canada

Salazar C, Cuyt A, Verdonk B (2007) Rational approximation of vertical segments. Numer Algorithms 45:375–388

Shahpar S, Polynkin A, Toropov VV (2008) Large scale optimization of transonic axial compressor rotor blades. In: AIAA 2008-2056, Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials conference. Schaumburg, IL

Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Methods Eng 24:259–373

Toropov VV (1989) Simulation approach to structural optimization. Struct Optim 1:37–46

Toropov VV, Filatov AA, Polynkin AA (1993) Multiparameter structural optimization using FEM and multipoint explicit approximations. Struct Optim 6:7–14

Vanderplaats GN (1999) Numerical optimization techniques for engineering design. 3rd edn. VR&D, Colorado Springs

van Keulen F, Toropov VV (1997) New developments in structural optimization using adaptive mesh refinement and multi-point approximations. Eng Optim 29:217–234

Viana FAC, Haftka RT (2008) Using multiple surrogates for metamodeling. In: Proceedings of 7th ASMO-UK/ISSMO international conference on engineering design optimization, Bath, UK, 7–8 July 2008, pp 132–137

Viana FAC, Haftka RT, Steffen Jr V (2009) Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. Struct Multidisc Optim 39:439–457

Wang LP, Grandhi RV (1995) Improved two-point function approximations for design optimization. AIAA J 33:1720–1727

Wang LP, Grandhi RV (1996) Multi-point approximations: comparisons using structural size, configuration and shape design. Struct Optim 12:177–185