

# Implementation of Discrete Capability into the enhanced Multipoint Approximation Method for solving mixed integer-continuous optimization problems

Dianzi Liu<sup>1</sup>, Vassili Toropov<sup>2</sup>

<sup>1</sup>Schools of Mathematics, University of East Anglia, Norwich NR4 7TJ, UK

<sup>2</sup>School of Engineering and Materials Science, Queen Mary University of London, London E1 4NS, UK

## Abstract

Multipoint approximation method (MAM) focuses on the development of meta models for the objective and constraint functions in solving a mid-range optimization problem within a trust region. To develop an optimization technique applicable to mixed integer-continuous design optimization problems in which the objective and constraint functions are computationally expensive and could be impossible to evaluate at some combinations of design variables, a simple and efficient algorithm, coordinate search, is implemented in the MAM. This discrete optimization capability is examined by the well established benchmark problem and its effectiveness is also evaluated as the discreteness interval for discrete design variables is increased from 0.2 to 1. Furthermore, an application to the optimization of a lattice composite fuselage structure where one of design variables (number of helical ribs) is integer is also presented to demonstrate the efficiency of this capability.

**Keywords:** Multipoint approximation method; Integer-continuous optimization; Metamodel; Coordinate search

## 1. Introduction

Lots of real-life structural optimization problems have the same common features: the objective and constraint functions are computationally expensive using the finite element methods or could be impossible to evaluate at some combinations of design variables (e.g., nodal values of displacements, stresses). In order to improve accuracy and computational efficiency, it is necessary to develop the robust approximation techniques. The Multipoint Approximation Method (MAM) is a technique replacing the original optimization problem by a succession of simpler mathematical programming. It utilizes high quality explicit approximations in order to reduce the total number of calls for analysis needed to solve large-scale optimization problems. This approach has been influenced by the previous work [1-3] on two-point approximation methods. This was later generalized to multi-point approximations [4-6].

Metamodel assembly is an approach, which is based on an assembly of multiple surrogates into a single surrogate using linear regression [7, 8]. The regression coefficients of the assembly model are not scaled weights but tuning parameters determined by the least squares method. Therefore, the tuning parameters of the assembly model are not restricted to a positive range but may have negative values as well. In the current work, following the previous research for a case of continuous variables [6], this approach is utilized in the MAM for design optimizations with integer-continuous variables.

In this paper the mixed integer-continuous design optimization problems are solved using MAM with enhanced discrete capability. In these cases, some of, or all, design variables are discrete. It is assumed that it is allowed to perform a response function evaluation only for points that have discrete values of the design variables [9]. This makes it impossible to initially ignore the discrete nature of the design variables, solve a continuous problem and adjust the result to the given set of the discrete values, as sometimes suggested in [10]. Thus, new procedures for sampling, metamodel building, their use for solving an optimization problem with discrete properties within a trust region, and the trust region adaptation strategy are required. In current research, a discrete form of the coordinate search algorithm [11] is implemented within the MAM to search for the solution in the sub-space of the discrete variables only starting from the optimal continuous values obtained by the Sequential Quadratic Programming method (SQP) on the approximated functions in a current trust region. This is compared to the use of a simple rounding-off strategy for the discrete variables. Based on the research by authors [12], the new mixed integer-continuous capability is implemented within the MAM and tested by two examples. The first example is the well-established benchmark ten-bar truss problem [13] and the effect of the size of

---

<sup>1</sup>Corresponding author. Tel: +44 01603592594

*E-mail address:* dianzi.liu@uea.ac.uk

discreteness on the optimal results obtained by MAM with the implementation of coordinate search algorithm and the simple rounding-off algorithm is examined as well. The second case study relates to the optimization of a lattice composite fuselage structure [14] where one of the design variables, the number of helical ribs, is integer. This composite lattice fuselage barrel is detailed in the EU FP7 research project ALaSCA in [15]. The obtained results are compared with the solutions obtained by a binary GA for the lattice fuselage barrel design. The efficiency of this technique is demonstrated by these two real-life industrial applications.

## 2. Multipoint Approximation Method (MAM)

The MAM is based on the building of mid-range approximations [4-6] and is suitable to solve large-scale optimization problems by producing better quality approximations that are sufficiently accurate in a current trust region and inexpensive in term of computational costs required for their building. These approximation functions have a relatively small number ( $N+1$  where  $N$  is number of design variables) of regression coefficients to be determined and the corresponding least squares problem can be solved easily. This feature of such approximations allows applying them to large scale optimization problems with the number of design variables in the order of hundreds [6].

The technique replaces the original optimization problem by a succession of simpler mathematical programming problems. The functions in each iteration present mid-range approximations to the corresponding original functions. The solution of an individual sub-problem becomes the starting point for the next step and the trust region is also changed. The optimization process is repeated iteratively until the optimum is reached. Each approximation function is defined as a function of both design variables and a number of tuning parameters. The tuning parameters 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 are taken from the extended trust region, i.e. the pool of points considered in the previous iterations that are located not too far from the current trust region.

A general optimization problem can be formulated as

$$\min F_0(\mathbf{x}), \quad F_j(\mathbf{x}) \leq 1 \quad (j=1, \dots, M), \quad A_i \leq x_i \leq B_i \quad (i=1, \dots, N) \quad (1)$$

where  $\mathbf{x}$  refers to the vector of design variables;  $A_i$  and  $B_i$  are the given lower and upper bounds on the design variable  $x_i$ ;  $N$  is the total number of the design variables;  $F_0(\mathbf{x})$  is an objective function;  $F_j(\mathbf{x})$  is the constraint function and  $M$  is the total number of the constraint functions.

In order to present the detailed model of a structure by the response functions and reduce the number of calls for the response function evaluations, the MAM replaces the optimization problem by a sequence of approximate optimization problems:

$$\min \tilde{F}_0^k(\mathbf{x}), \quad \tilde{F}_j^k(\mathbf{x}) \leq 1 \quad (j=1, \dots, M), \quad A_i^k \leq x_i \leq B_i^k, \quad A_i^k \geq A_i, \quad B_i^k \leq B_i \quad (i=1, \dots, N) \quad (2)$$

where  $\tilde{F}_0^k(\mathbf{x})$  and  $\tilde{F}_j^k(\mathbf{x})$  are the functions which approximate the functions  $F_0(\mathbf{x})$  and  $F_j(\mathbf{x})$  of the initial optimization problem defined in Eq.(1).  $A_i^k$  and  $B_i^k$  are the side constraints of a trust region when searching the solution of the approximate optimization problem,  $k$  is the iteration number.

The selection of the approximate response functions  $\tilde{F}_j^k(\mathbf{x})$  ( $j=0, \dots, M$ ) is such that their evaluation is inexpensive as compared to the evaluation of the response functions  $F_j(\mathbf{x})$ , although they are not necessarily explicit functions of the design variables. The approximate response functions are intended to be adequate in a current trust region. This is achieved by appropriate planning of numerical experiments and use of the trust region defined by the side constraints  $A_i^k$  and  $B_i^k$ . The Figure 1 illustrates the main steps of the optimization procedure using MAM.

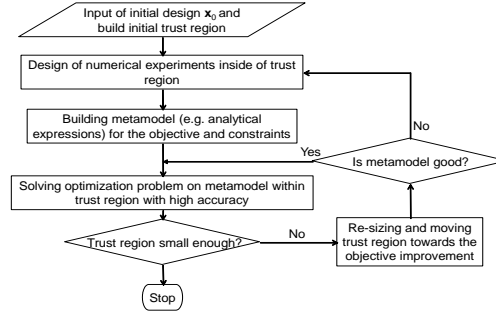


Fig. 1. Main steps of the optimization procedure using MAM

### 3. Design of Experiments

In the MAM algorithm, the Design of Experiments (DoE) is intentionally simple and is based on a random generation of sampling points taking into account the discrete nature of some of (or all) the design variables. This ensures that the calls for the response function evaluations are only made for the points that have discrete values of the design variables. The added points are checked for calculability of the response function and, if the check fails (i.e. the simulation software did not return the response function values), a new set of points is generated until a required number of sampling points (all passing the check) are obtained. To improve the quality of the random DoE, a constraint on the minimal distance between the points is imposed [6] using the following expression:

$$\frac{dist^p}{Diag} \geq r \quad (3)$$

where

$$Diag = \sqrt{\sum_{i=1}^N (B_i^k - A_i^k)^2}, dist^e = \sqrt{\sum_{i=1}^N (x_i^e - x_i^p)^2} \quad (e, p = 1, \dots, P; e \neq p),$$

$Diag$  is a characteristic size of the trust region,  $e$  is a number of a new sampling point,  $p$  is a number of a previously generated point,  $P$  is the total number of sampling points in the trust region,  $N$  is the total number of the design variables,  $A_i^k$  and  $B_i^k$  are the side constraints of a trust region when searching the solution of the approximate optimization problem,  $k$  is the iteration number. The parameter  $r$  is initially set to 0.9 and if the assessment on Eq. (3) fails after a prescribed number of random generations for a new point, the value  $r$  will be iteratively reduced by multiplying a positive constant ( $r = r * coeff$ ,  $0.9 \leq coeff < 1$ ) until the Eq. (3) is satisfied. Imposing such a mechanism on the selection of points, the uniform distribution of sampling points across the space is guaranteed.

A seed for random numbers is used to create the sampling points in this paper. To show the influence of the randomness of the sampling points on minimal distances between points in DoE, Example 1 in Section 7.1.1 is selected to demonstrate the indication of this scatter shown in Figure 2. Note that the seed for random numbers is taken as 1 for all the examples in this paper due to its capability to generate more uniformly distributed sampling points. As an example indicating a good quality of the DoE, 30 points is generated for a two-dimensional test problem. Using the proposed technique, the comparison of patterns of sampling points without and with the consideration of a constraint imposed on the minimum distance is shown in Figure 3.

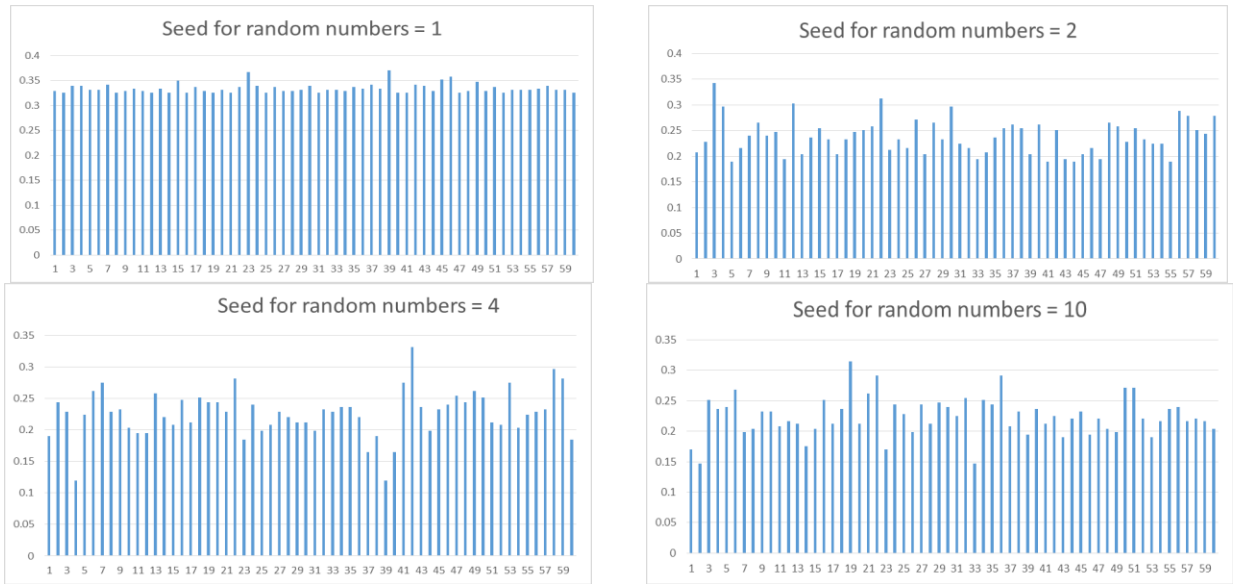


Fig. 2. Effect of randomness of sample points on minimal distance between points in 60–point design of experiments

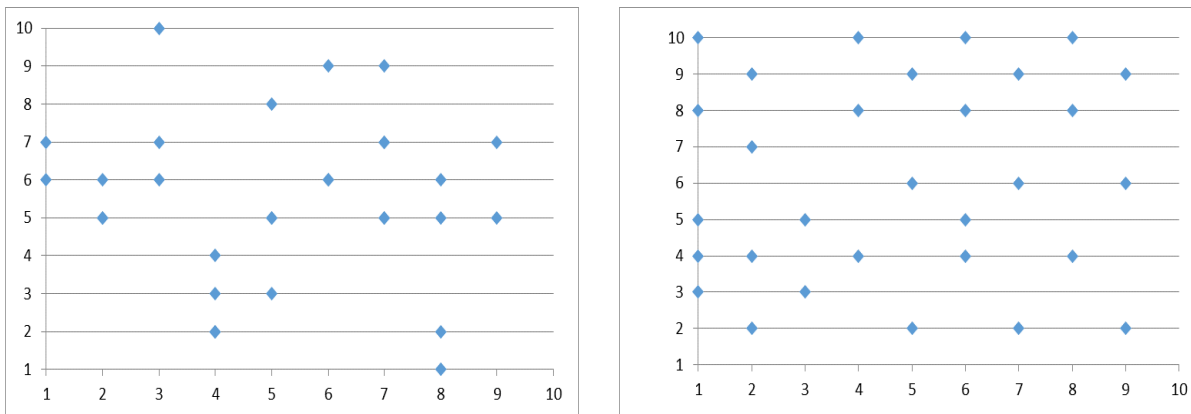


Fig. 3. 30-point patterns without (left) and with (right) a constraint imposed on the minimal distance when design variables only take discrete values

#### 4. Trust Region Strategy

The aim of the trust region strategy is to control the quality of a metamodel. When the approximation gets better, the trust region will be further reduced. The track of the trust regions also indicates a path of the direction from the initial starting point to the optimum over the entire searching domain. On each MAM iteration, a new trust region must be defined, i.e. its new size and its location have to be specified. It should be noted that if the sub-optimum point does not pass the check for calculability of the response functions, the trust region is reduced and the approximated problem is solved again. The only essential assumption here is that all functions of the optimization problem exist at the starting point. Note that for the discrete variables there is a restriction on the trust region size reduction, i.e. the corresponding size of the trust region has to contain at least three levels of a discrete variable. The Figure 4 summarizes the trust region strategy in MAM algorithm.

Approximations 'good'					
'Internal'			'Boundary'		
'Small'	'Large'		'Backwards'	'Forward'	
Stop: Convergence found	'Close'	'Far'	Reduce	'Straight'	'Curved'
	Reduce	Reduce		Keep model & Enlarge	Keep model & Keep size

Approximations 'reasonable'					
'Internal'			'External'		
'Small'	'Large'		'Backwards'	'Forward'	
Stop: Convergence found	'Close'	'Far'	Reduce	'Straight'	'Curved'
	Reduce	Reduce		Enlarge	Keep size

Approximations 'bad'			
'Small'	'Large'		
Stop: No convergence found	'Straight'		'Curved'
	Reduce moderate or even enlarge		Reduce

“Internal, Boundary, External” – location of sub-optimal point  $x_k^*$  in respect to trust region

“Small, Large” – specified minimal relative size of trust region and current size that is above Small

“Close, Far” – distance from the obtained point  $x_k^*$  to the boundary of feasible region: depending on that, the option “Reduce” is applied with different coefficients

“Backwards, Forward (Straight/Curved) - directions of trust region movement

“Good, Reasonable, Bad” – quality of the metamodel estimated by the expression:

$$r_k = \left| [F(\mathbf{x}_k^*) - \tilde{F}_k(\mathbf{x}_k^*)] / F(\mathbf{x}_k^*) \right|$$

Fig. 4. Trust region indicators

## 5. Using an Assembly Technique to Build Approximations

Based on the research on multiple metamodels [7, 8], a two-step regression procedure is applied to build the metamodel [6] represented by the assembly of different approximations. This was introduced in the form

$$\tilde{F}(x) = \sum_{l=1}^{NF} b_l \varphi_l(x), \quad (4)$$

where  $\varphi_l(x)$  means different approximate models, which are assembled into one metamodel;  $NF$  is the number of regressors in the approximate models  $\{ \varphi_l(x) \}$ ;  $b_l$  is the corresponding regression coefficient and reflects the accuracy of the individual surrogates on a set of validation points.

In order to build the expression in Eq. (4), the optimization of the minimization of the expression

$$\min \sum_{p=1}^P w_{pj} [F_j(\mathbf{x}_p) - \tilde{F}_j(\mathbf{x}_p, \mathbf{b}_j)]^2 \quad (5)$$

is formulated to determine the regression coefficients  $\mathbf{b}_j$ . This weighted least squares problem leads to solving a linear system of  $NF$  equations with  $NF$  unknowns  $\mathbf{b}_j$ . Here the regression coefficients  $\mathbf{b}_j$  should not be considered as weight factors, e.g. could be either positive or negative. The parameters  $w_{pj}$  refer to the weights that reflect the inequality of data obtained at different sampling points [4, 5].

The functions  $\varphi_l$  in (4) are determined in the similar manner

$$\min \sum_{p=1}^P w_{pj} [F_j(\mathbf{x}_p) - \varphi_j(\mathbf{x}_p, \mathbf{a}_j)]^2 \quad (6)$$

where the minimization is carried out with respect to the tuning parameters  $\mathbf{a}_j$ ,  $P$  means the number of points used in a specified DoE (see Section 3). Note that the tuning parameters  $\mathbf{a}_j$  in Eq. (6) should be determined before the procedure (5) has been applied.

The evaluation of the regressors  $\varphi_l$  is based on the data from the sampling points currently located in the trust region. In the mid-range approximation framework, inexpensive approximate models for objective and constraint functions are built using minimum required number of sampling points. The simplest case is a linear function and more complex ones are intrinsically linear functions [16] that have been successfully used for a variety of design optimization problems [4-6]. Intrinsically linear functions are nonlinear but they can be led to linear ones by simple transformations. In this paper, the intrinsically linear functions considered in the model bank  $\{\varphi_l(x)\}$  are:

$$\begin{aligned}
\varphi(\mathbf{x}) &= a_0 + \sum_{i=1}^N a_i x_i \\
\varphi(\mathbf{x}) &= a_0 \prod_{i=1}^N x_i^{a_i} \\
\varphi(\mathbf{x}) &= a_0 + \sum_{i=1}^N a_i / x_i \\
\varphi(\mathbf{x}) &= a_0 + \sum_{i=1}^N a_i x_i^2 \\
\varphi(\mathbf{x}) &= a_0 + \sum_{i=1}^N a_i / x_i^2
\end{aligned} \tag{7}$$

## 6. Local Optimization Procedure

In each iteration of MAM, a continuous solution is initially sought in a current trust region using the SQP method applied to the approximated optimization problem (2). The discrete variables are initially obtained by rounding the continuous variables to the nearest discrete ones, fixing these values and solving a continuous optimization problem again (with the reduced number of design variables to include only continuous variables). This becomes a starting guess for the discrete form of the optimization problem (2). Two approaches to solving discrete optimization problems have been examined, accepting the solution obtained after the rounding-off, and the discrete form of the coordinate search method.

The coordinate search algorithm examines points near the current point by perturbing design variables - one variable at a time - until an improved point is found. There is a similarity between this technique and one previously suggested in [9] although for a different type of approximations. The algorithm begins with the starting point, as well as  $2N_d$  coordinate points, where  $N_d$  is the number of discrete design variables. The  $i^{\text{th}}$  pair of coordinate points differs from the starting point only in the  $i^{\text{th}}$  coordinate. The next test point will be placed by a perturbation  $\Delta_i$  (positive or negative) along the  $i^{\text{th}}$  coordinate. The size of  $\Delta_i$  is determined by the spacing of the  $i^{\text{th}}$  discrete design variable. The point with the lowest objective function value (that is penalized in case of violated constraints) along  $i^{\text{th}}$  coordinate will be selected as the new starting point for the next  $(i+1)^{\text{th}}$  coordinate search. As direct search algorithms [11] are known as unconstrained optimization techniques, an exterior penalty function is used to accommodate the constraints by penalizing unfeasible solutions as follows:

$$f(\mathbf{x}) = \frac{F_0(\mathbf{x})}{F_0^*} + \alpha \sum_{i=1}^m \max[1, F_i(\mathbf{x})]^\beta \tag{8}$$

where  $f(\mathbf{x})$  is the objective function penalized in case any of the constraints is violated,

$F_0(\mathbf{x})$  is the objective function,

$F_0^*$  is the initial value of objective function at the starting point,

$\alpha, \beta$  are the penalty parameters, here  $\alpha = 0.5$  and  $\beta = 1$  are suggested

$F_i(\mathbf{x})$  is the  $i$ -th constraint function,  $i = 1, \dots, m$ ,

$m$  is the total number of constraints.

Every time when the discrete variables are modified by the discrete optimizer, it is necessary to make an adjustment of the remaining, continuous design variables. This is carried out by the use of SQP in the space of continuous variables only. Therefore, the overall local optimization process that is carried out on the approximated response functions, is carried out in two levels. The outer optimization loop deals with the discrete variables only and is based on the discrete form of the two search algorithms, and the inner optimization loop adjusts the continuous design variables using SQP.

## 7. Numerical Examples:

### 7.1 Ten-bar Truss Structure

#### 7.1.1 Example 1

The developed approach was tested on the well known ten-bar truss benchmark problem [13] shown in Figure 5 that has been used by many authors. The minimum weight design is obtained by changing the cross-sectional areas (from 0.1 in<sup>2</sup> to 12.7 in<sup>2</sup> with the increment of 0.2) of the truss members subject to stress constraints and minimum gage constraints of 0.1 in<sup>2</sup>. The allowable stress in each truss member is the same in tension and compression and is set to 25 ksi for all members except member 9 for which it is 75 ksi. The density of the truss material is 0.1 lb/in<sup>3</sup>, the member size  $L=360$  in, the loads  $P_1 = P_2 = 100$  Kips and  $P_3 = 0$ .

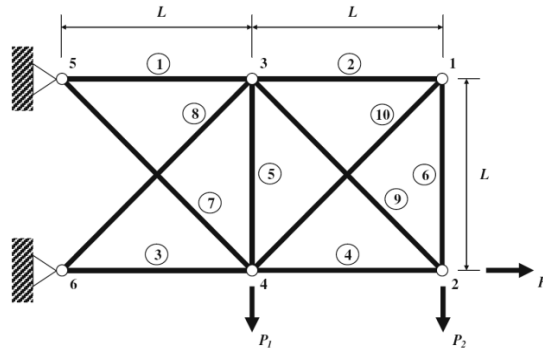


Fig. 5. Ten-bar truss structure

The results for weight optimization of the ten-bar truss structure are presented in Table 1 and information on the constraint values for the optimal design is listed in Table 2. In order to compare to the known results [13], the weight optimization of the ten-bar truss with continuous properties (cross-sectional areas) is performed by MAM. The number of sampling points used in each iteration is 15. The optimal result (1497.1) of weight optimization by MAM with continuous variables is almost the same as of the benchmark [13] (1497.0) and of the optimization by SQP [17] (1497.6). All these three optimal designs are (almost) feasible solutions with only a small violation of constraints (by 0.4%), see Table 2. For the discrete optimization of ten-bar truss, optimization by MAM with the implementation of coordinate search algorithm has achieved the same optimal weight as the one by the simple rounding-off, but the number of iterations (24) for the coordinate search algorithm is only half of the total number required by the simple rounding-off algorithm (48). For the number of response analyses in both algorithms, the same conclusion could be made. Hence, the computational efficiency of the coordinate search algorithm is higher than the simple rounding-off method for the discrete optimization using MAM, whereas both obtained the same result. The maximum constraint violation for the either implementation of MAM was 0.8% that was deemed acceptable.

Next, the obtained results were compared to the results obtained by a binary genetic algorithm (GA). Since a GA is naturally suitable to the discrete optimization and it is also a global search algorithm, a better result from GA could be expected. On the contrary, a greater weight (1560.4) was obtained, that is the best result out of five separate GA runs. This optimal design is a feasible solution with no violation of constraints (see Table 2) and also gives the explanation why this slightly heavier truss design is obtained. As expected, the number of iterations and response analyses in the optimization by binary genetic algorithm are significantly increased.

Table 1. Results of continuous and discrete optimizations

Design variables	Continuous variables, in <sup>2</sup> [13]	Continuous variables, in <sup>2</sup> (SQP, [17])	Continuous variables, in <sup>2</sup> (MAM)	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)	Discrete variables, in <sup>2</sup> (GA, [17])
$x_1$	7.9	7.9	7.9	7.5	7.5	7.7
$x_2$	0.1	0.1	0.1	0.5	0.5	0.5
$x_3$	8.1	8.1	8.1	8.5	8.5	8.5
$x_4$	3.9	3.9	3.9	3.5	3.5	3.7
$x_5$	0.1	0.1	0.1	0.1	0.1	0.1
$x_6$	0.1	0.1	0.1	0.5	0.5	0.5
$x_7$	5.8	5.8	5.8	6.3	6.3	6.3
$x_8$	5.51	5.52	5.52	5.1	5.1	5.1
$x_9$	3.68	3.68	3.66	3.3	3.3	3.7
$x_{10}$	0.14	0.14	0.14	0.7	0.7	0.7
Obj. (lb)	1497.0	1497.6	1497.1	1525.6	1525.6	1560.4
No. of iterations	N/A	13	22	24	48	57
No. of response analyses	N/A	144	331	361	721	7157

Table 2. Constraints results of the continuous and discrete optimizations (violated constraint values shown in bold)

Constraints (ksi)	Continuous variables, in <sup>2</sup> [13]	Continuous variables, in <sup>2</sup> (SQP,[17])	Continuous variables, in <sup>2</sup> (MAM)	Discrete variables, in <sup>2</sup> (MAM)	Discrete variables, in <sup>2</sup> (GA, [17])
$G_1$	25.0	25.0	25.0	25.0	24.5
$G_2$	24.9	25.0	<b>25.1</b>	24.6	22.6
$G_3$	25.0	25.0	25.0	25.0	24.8
$G_4$	25.0	25.0	25.0	25.0	24.0
$G_5$	0.07	6.5e-7	0.05	1.5	1.0
$G_6$	24.9	25.0	<b>25.1</b>	24.6	22.6
$G_7$	25.0	25.0	25.0	<b>25.2</b>	25.0
$G_8$	25.0	25.0	25.0	24.4	24.6
$G_9$	37.5	37.5	37.5	37.5	33.9
$G_{10}$	<b>25.1</b>	25.0	25.0	25.0	22.8

7.1.2 Example 2

In this example all the design parameters are the same as the ones in *Example 1*, but the allowable stress for the member 9 is set to 25 ksi. The optimal results and constraint values for weight optimization of this ten-bar truss structure are presented in Table 3 and 4, respectively. The optimal result (1593.1) of weight optimization with continuous variables by MAM is the same as of the optimization by SQP. For the discrete optimization of this ten-bar truss structure, optimization by MAM with the implementation of coordinate search algorithm has achieved the best result (the lightest weight, 1610.1) and a slightly heavier weight (1617.3) is obtained by MAM with the simple rounding-off. These two results are much better than the discrete results from [17-19] shown in Table 3. The numbers of response analyses for both coordinate search algorithm and simple rounding-off algorithm have been reduced by an order magnitude as



compared to the results from other published paper in Table 3. It is concluded that the computational efficiency of the coordinate search algorithm and the simple rounding-off method for the discrete optimization using MAM are quite higher than GA. All the optimal designs obtained by discrete optimizations are feasible solutions with no violation of constraints (see Table 4).

Table 3. Results of continuous and discrete optimizations

Design variables	Continuous variables, in <sup>2</sup> (SQP,[17])	Continuous variables, in <sup>2</sup> (MAM)	GA, in <sup>2</sup> [18]	Discrete variables, in <sup>2</sup> [19]	Discrete variables, in <sup>2</sup> (GA, [17])	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)
$x_1$	7.94	7.94	N/A	8.1	7.7	7.9	7.5
$x_2$	0.10	0.10		0.1	0.5	0.1	0.3
$x_3$	8.06	8.06		8.1	8.5	8.1	8.3
$x_4$	3.94	3.94		4.1	3.7	4.1	3.7
$x_5$	0.10	0.10		0.1	0.1	0.1	0.1
$x_6$	0.10	0.10		0.1	0.5	0.1	0.5
$x_7$	5.74	5.74		5.9	6.3	5.7	6.1
$x_8$	5.57	5.57		5.7	5.1	5.7	5.3
$x_9$	5.57	5.57		5.7	3.7	5.7	5.3
$x_{10}$	0.10	0.10		0.1	0.7	0.1	0.5
Obj. (lb)	1593.1	1593.1	1635	1627.5	1627.5	1610.1	1617.3
No. of iterations	20	21	N/A	120	57	30	27
No. of response analyses	211	316	17906	5190	7206	451	406

Table 4. Constraints results of the continuous and discrete optimizations

Constraints (ksi)	Continuous variables, in <sup>2</sup> (SQP, [17])	Continuous variables, in <sup>2</sup> (MAM)	Discrete variables, in <sup>2</sup> [19]	Discrete variables, in <sup>2</sup> (GA, [17])	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)
$G_1$	25.0	25.0	24.5	24.5	25.0	25.0
$G_2$	15.5	15.5	15.1	15.1	14.9	22.8
$G_3$	25.0	25.0	24.8	24.9	24.9	24.9
$G_4$	25.0	25.0	24.0	24.0	24.0	25.0
$G_5$	1.7e-8	9.0e-4	0.14	0.1	4.4e-2	1.2e-2
$G_6$	15.5	15.5	15.1	15.1	14.9	13.7
$G_7$	25.0	25.0	24.3	24.3	25.0	24.8
$G_8$	25.0	25.0	24.4	24.4	24.5	24.8
$G_9$	25.0	25.0	24.4	24.4	24.5	24.8
$G_{10}$	22.0	22.0	21.3	21.3	21.0	19.3

### 7.1.3 Example 3

A mixed integer-continuous design problem has been solved in this example. The optimization formulation has been defined as:

Objective: *Minimize Weight*

Constraint: The allowable stress  $G_i$ ,  $G_i \leq 25ksi, i = 1, \dots, 10$  except for  $G_9 \leq 75ksi$ ;

Design Variables: Cross-sectional areas are changing from 0.1 in<sup>2</sup> to 12.7 in<sup>2</sup>; Continuous cross-sectional areas for truss member 1 to 6 (horizontal and vertical members); Discrete cross-sectional areas for truss member 7 to 10 (diagonal members) with the increment of 0.2.

In Table 5, the optimal designs of the ten-bar truss structure by GA, MAM with the implementation of coordinate search algorithm and the simple rounding-off algorithm are presented. Since the optimal design obtained by MAM with the coordinate search algorithm (1506.7) has reduced the weight by 5% as compared to the one by GA (1583.3), more constraints have been activated or become critical in the converged solution shown in Table 6. As expected, the number of iterations and response analyses in the mixed integer-continuous optimization by MAM are significantly reduced. The number of response analyses for GA (14901) has been reduced by 96% as compared to the result by the coordinate search algorithm (526), while the lighter weight is also achieved by the latter. The similar conclusion could be made for the results obtained by MAM with the simple rounding-off algorithm and GA. The lightest design shown in Table 5 is 1506.7 by MAM with the coordinate search algorithm and this requires acceptable more numbers of iterations (35) and response analyses (526) than the ones for the slightly heavier design (1534.3) by MAM with the simple rounding-off algorithm. In terms of the computational efficiency, both the coordinate search algorithm and the simple rounding-off method for the optimization of this truss structure are much higher than GA. MAM, enhanced by the discrete optimization capability, outperforms a GA not only in the efficiency but also in the quality of the obtained solution. In Table 6, all the solutions are feasible and no constraint violation happens in these three discrete optimizations.

Table 5. Results of the mixed integer-continuous optimizations (discrete design variables shown in bold)

Design variables	Discrete variables, in <sup>2</sup> (GA, [17])	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)
$x_1$	7.75	7.86	7.43
$x_2$	0.21	0.21	0.63
$x_3$	8.25	8.15	8.59
$x_4$	3.80	3.92	3.45
$x_5$	0.22	0.1	0.1
$x_6$	0.27	0.21	0.64
$x_7$ (discrete)	<b>6.7</b>	<b>5.9</b>	<b>6.5</b>
$x_8$ (discrete)	<b>5.3</b>	<b>5.5</b>	<b>4.9</b>
$x_9$ (discrete)	<b>4.3</b>	<b>3.5</b>	<b>3.1</b>
$x_{10}$ (discrete)	<b>0.3</b>	<b>0.3</b>	<b>0.9</b>
Obj. (lb)	1583.3	1506.7	1534.3
No. of iterations	114	35	11
No. of response analyses	14901	526	166

Table 6. Constraints results of the mixed integer-continuous optimizations (discrete design constraints shown in bold)

Constraints (ksi)	Discrete variables, in <sup>2</sup> (GA, [17])	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)
$G_1$	25.0	24.9	24.8
$G_2$	24.6	25.0	25.0
$G_3$	25.0	25.0	25.0
$G_4$	25.0	24.7	24.4
$G_5$	5.1	7.0e-2	7.4e-2
$G_6$	19.0	25.0	24.7
$G_7$	<b>22.4</b>	<b>25.0</b>	<b>25.0</b>
$G_8$	<b>25.0</b>	<b>24.4</b>	<b>24.4</b>
$G_9$	<b>31.2</b>	<b>38.9</b>	<b>38.4</b>
$G_{10}$	<b>24.4</b>	<b>24.7</b>	<b>24.8</b>

#### 7.1.4 Effect of the discreteness interval on the results

In above three examples, design variables have values with the discreteness interval of 0.2 in the optimization process. Such small discreteness interval might be of the mildest possible form in the discrete optimization problems. It remains to be determined how the accuracy and efficiency of the proposed two methods (MAM with the implementation of coordinate search algorithm and the simple rounding-off algorithm) will be influenced as the discreteness interval is relatively large, for example 1. To examine the potential of MAM with the developed discrete capability, the optimal results in the above three examples with the discreteness interval of 1 for discrete design variables are compared with the ones by GA.

The optimal designs of the ten-bar truss structure in the preceding Example 1 by GA, MAM with the implementation of coordinate search algorithm and the simple rounding-off algorithm are presented in Table 7 and the associated constraint values are listed in Table 8. Optimization by MAM with the implementation of coordinate search algorithm has achieved the lightest weight (1612.6) compared with the result (1684.6) by the simple rounding-off and the solution (1663.5) by GA. Although the numbers of iterations (86 and 120) are required to converge the optimization analysis using the coordinate search algorithm and the simple rounding-off algorithm are more than the one by GA (57), the associated numbers of response analyses (1291 and 1801) are far less than the number required by GA (6900). It indicates that the computational efficiency of the coordinate search algorithm and the simple rounding-off method is much higher than GA. By comparing weights of the optimal designs obtained by the either implementation of MAM and GA, the quality of the obtained designs is well kept, whereas the maximum constraint violation for both enhanced MAM is 0.8% that falls within an acceptable tolerance.

The optimal designs and constraint values are shown in Table 9 and 10 for the foregoing Example 2 in the optimization process with the discreteness interval of 1. In Table 9, better designs by both coordinate search algorithm (1786.4) and simple rounding-off method (1801.3) can be identified as compared to the one by GA (1858.4). The numbers of response analyses are much reduced from 6764 (by GA) to 1711 (simple rounding-off method) or 376 (by coordinate search algorithm). As expected, implementation of MAM by the coordinate search algorithm can solve the optimization problem more efficiently and produce a better design than the simple rounding-off method. Either implementation of MAM outperforms GA to obtain an improved feasible design (see Table 10) in terms of the structural weight and number of analyses.

Finally, the mixed integer-continuous design optimization problem (Example 3) has been solved with respect to the size of discreteness of 1 for discrete design variables. The very expensive computational efforts of GA method are considerably reduced by the methods of either implementation of MAM shown in Table 11. The number of response analyses by GA (12375) has been reduced by 95.7% as compared to the result by the coordinate search algorithm (526), while the lightest weight (1600.4) is also achieved by the latter. With the comparison of MAM implemented with the simple rounding-off method, GA can generate a slightly lighter design (1666.6), but the computational cost in search for

the optimal solution has significantly increased to 12375 from 1396 in terms of the number of analyses. Again, all three optimal designs are feasible solutions and no constraint violation has been observed in Table 12.

Based on the above six examples, the effect of the discreteness interval on the optimal designs in terms of computational efficiency and the quality of solutions has been evaluated by comparing the results from either implementation of MAM and GA. When the size of the discreteness is increased from 0.2 to 1 for discrete design variables, the heavier weights of the optimal designs are always obtained. The reason for that is obviously because of the reduced design space in the optimization process. When the discreteness interval is set to the mildest value (0.2), there is no distinct difference in the optimal designs between the coordinate search algorithm and the simple rounding-off method. Either implementation of MAM outperforms GA (see Table 1, 3 and 5). As the discreteness interval is increased to 1, the best designs by MAM with the implementation of coordinate search algorithm are given at a lowest computational cost as compared to MAM with the implementation of simple rounding-off method and GA (see Table 7, 9 and 11). MAM with the implementation of simple rounding-off method can produce the same level of quality of the results as GA, while it requires much fewer response analyses and less computational expense. It turns out that when the size of the discreteness is increased, the simple rounding-off method can not solve the optimization problem as efficiently and precisely as the coordinate search algorithm due to its simple search mechanism. Furthermore, either implementation of MAM can obtain the better or similar designs as GA does, but the computational efficiency to achieve the optimal designs has been significantly improved. In summary, comparing with the results by GA, the coordinate search algorithm is a more robust method as compared with the simple rounding-off method in the discrete optimization problems, especially when the relatively large value is assigned to the discreteness interval. The designs can be significantly improved by MAM with the implementation of coordinate search algorithm by comparing with the results by GA as the discreteness interval is increased in the optimization problems. For MAM with the implementation of simple rounding-off method, the optimal results can be slightly better or almost the same level of quality as GA.

Table 7. Optimal results for ten-bar truss structure designs (Example 1) with the size of discreteness of 1.0 for design variables

Design variables	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)	Discrete variables, in <sup>2</sup> (GA, [17])
$x_1$	7.1	7.1	7.1
$x_2$	1.1	1.1	1.1
$x_3$	9.1	9.1	9.1
$x_4$	3.1	3.1	3.1
$x_5$	0.1	2.1	0.1
$x_6$	1.1	1.1	1.1
$x_7$	7.1	7.1	7.1
$x_8$	4.1	4.1	5.1
$x_9$	3.1	3.1	3.1
$x_{10}$	2.1	2.1	2.1
Obj. (lb)	1612.6	1684.6	1663.45
No. of iterations	86	120	57
No. of response analyses	1291	1801	6900

Table 8. Constraints results of discrete optimizations with the size of discreteness of 1.0 for design variables (violated constraint values shown in bold)

Constraints (ksi)	Discrete variables, in <sup>2</sup> (MAM with coordinate search)	Discrete variables, in <sup>2</sup> (MAM with rounding off)	Discrete variables, in <sup>2</sup> (GA, [17])
$G_1$	24.3	24.4	24.6
$G_2$	25.0	<b>25.2</b>	23.7
$G_3$	25.0	25.0	24.8
$G_4$	23.3	23.2	23.8
$G_5$	1.6	0.5	8.6
$G_6$	25.0	<b>25.2</b>	23.7
$G_7$	<b>25.2</b>	<b>25.2</b>	25.0
$G_8$	25.0	25.0	20.7
$G_9$	33.0	32.9	33.7
$G_{10}$	18.7	18.8	17.6

Table 9. Optimal results for ten-bar truss structure designs (Example 2) with the size of discreteness of 1.0 for design variables

Design variables	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)	Discrete variables, in <sup>2</sup> (GA, [17])
$x_1$	7.1	7.1	9.1
$x_2$	2.1	1.1	1.1
$x_3$	9.1	9.1	8.1
$x_4$	3.1	3.1	4.1
$x_5$	1.1	1.1	2.1
$x_6$	1.1	1.1	1.1
$x_7$	8.1	8.1	6.1
$x_8$	4.1	4.1	6.1
$x_9$	4.1	5.1	5.1
$x_{10}$	2.1	2.1	1.1
Obj. (lb)	1786.4	1801.3	1858.4
No. of iterations	25	114	57
No. of response analyses	376	1711	6764

Table 10. Constraints values of discrete optimizations with the size of discreteness of 1.0 for design variables

Constraints (ksi)	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)	Discrete variables, in <sup>2</sup> (GA, [17])
$G_1$	24.0	24.3	21.7
$G_2$	13.1	20.7	13.0
$G_3$	25.0	25.0	25.0
$G_4$	23.4	25.0	20.9
$G_5$	1.9	4.5	5.6

$G_6$	25.0	20.7	13.0
$G_7$	22.7	22.3	23.8
$G_8$	24.3	25.0	22.6
$G_9$	25.0	21.4	23.8
$G_{10}$	18.6	15.4	18.4

Table 11. Optimal results for ten-bar truss structure designs (Example 3) with the size of discreteness of 1.0 for design variables (discrete design variables shown in bold)

Design variables	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)	Discrete variables, in <sup>2</sup> (GA, [17])
$x_1$	7.30	7.86	6.73
$x_2$	0.76	0.54	1.43
$x_3$	8.80	8.57	9.29
$x_4$	3.28	3.65	2.72
$x_5$	0.36	0.84	0.1
$x_6$	0.76	0.54	1.42
$x_7$ (discrete)	<b>7.1</b>	<b>7.1</b>	<b>8.1</b>
$x_8$ (discrete)	<b>5.1</b>	<b>5.1</b>	<b>4.1</b>
$x_9$ (discrete)	<b>3.1</b>	<b>4.1</b>	<b>3.1</b>
$x_{10}$ (discrete)	<b>1.1</b>	<b>1.1</b>	<b>2.1</b>
Obj. (lb)	1600.4	1677.7	1666.6
No. of iterations	35	93	95
No. of response analyses	526	1396	12375

Table 12. Constraints results of discrete optimizations with the size of discreteness of 1.0 for design variables (constraints associated with discrete cross-sectional areas of bars shown in bold)

Constraints (ksi)	Discrete variables, in <sup>2</sup> (MAM with coordinate search algorithm)	Discrete variables, in <sup>2</sup> (MAM with a rounding-off algorithm)	Discrete variables, in <sup>2</sup> (GA, [17])
$G_1$	25.0	23.7	25.0
$G_2$	24.8	25.0	22.3
$G_3$	24.8	25.0	25.0
$G_4$	24.8	23.7	25.0
$G_5$	1.6	0.8	0.8
$G_6$	24.5	24.9	22.5
$G_7$	<b>23.5</b>	<b>22.7</b>	<b>23.0</b>
$G_8$	<b>22.8</b>	<b>23.8</b>	<b>23.4</b>
$G_9$	<b>37.1</b>	<b>29.9</b>	<b>31.0</b>
$G_{10}$	<b>24.0</b>	<b>17.3</b>	<b>21.5</b>

7.2 A lattice composite fuselage structure

Another mixed integer-continuous example, presented in Figure 6, relates to the optimization of a lattice composite fuselage structure where one of the design variables, the number of helical ribs, is integer. A schematic of a composite lattice barrel, considered in the ALaSCA EU FP7 research project [14] is shown in Figure 7. The bounds on the design variables used for the FE modelling of the barrel model in the weight optimization are listed in Figures 8, 9 and Table 13. The optimization formulation has been defined as (see [15]):

Objective: *Minimize* weight of the fuselage barrel;

Constraint: Subject to normalized strain constraints on the fuselage barrel;

Design Variables: six continuous design variables and one discrete design variable (see Table 13).

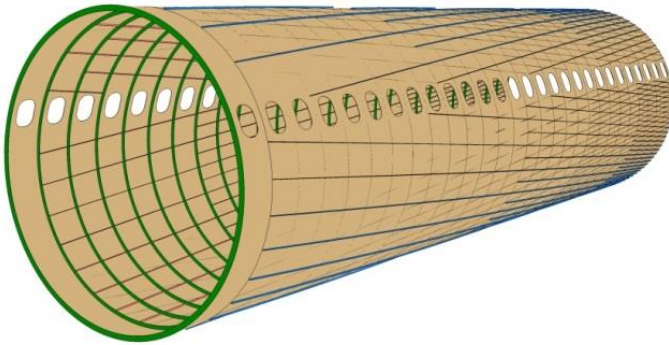


Fig. 6. The lattice composite fuselage structure

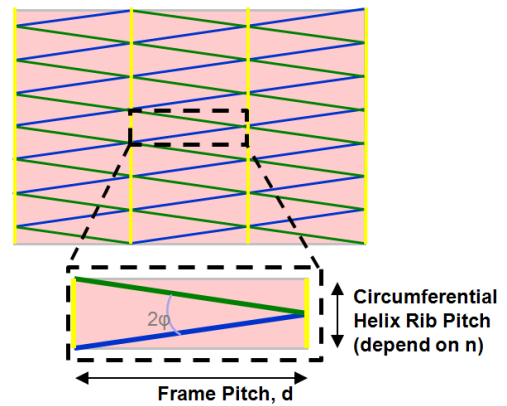


Fig. 7. A schematic of the composite lattice barrel (frames are shown in yellow; helical ribs in green and blue; skin in red)

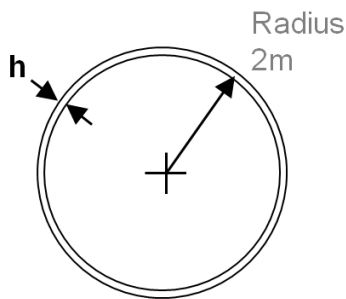


Fig. 8. Barrel cross section

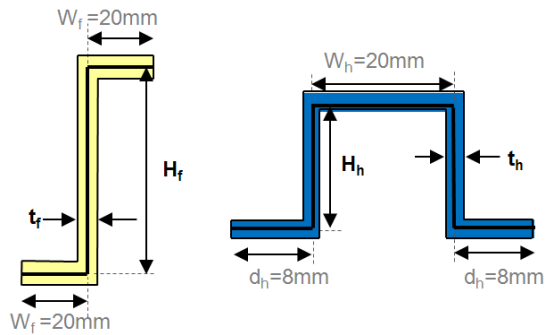


Fig. 9. Circumferential ribs and helical ribs

Table 13. Design variables properties  
(discrete design variable shown in bold)

Design variables	Lower bound	Upper bound
Skin thickness ( $h$ )	0.6 (mm)	4.0 (mm)
<b>Number of helix rib pairs around the circumference, (<math>n</math>)</b>	<b>50</b>	<b>150</b>
Helix rib (hat-section) thickness, ( $t_h$ )	0.6 (mm)	3.0 (mm)
Helix rib (hat-section) height, ( $H_h$ )	15 (mm)	30 (mm)
Frame pitch, ( $d$ )	500 (mm)	650 (mm)
Frame (z-section) thickness, ( $t_f$ )	1.0 (mm)	4.0 (mm)
Frame (z-section) height, ( $H_f$ )	50 (mm)	150 (mm)

Optimal design of the fuselage barrel leads to a mixed discrete-continuous optimization problem where one parameter (number of helical ribs) is integer and other parameters, e.g. helical rib height, can be considered as continuous. In order to perform weight optimization of this barrel structure efficiently, metamodel-based optimization approach is applied in this paper. Genetic Programming (GP) [20, 21] is used to build the metamodels for structural responses of interest obtained by the FE analysis. For more details on GP and FE simulations, see [15]. Since GP metamodels are analytical expression that provides the best fit into the data from the FE runs, the prohibitive computational effort is avoided to solve this parametric optimization problem. Once the metamodels of all structural responses have been built, the weight optimization is then performed by a Genetic Algorithm (GA), MAM with a rounding-off algorithm and MAM with the coordinate search algorithm, respectively. It should be noted that the validation of predicted optimal results by FE analysis has also been carried out for the final design.

The optimal results by GA, MAM with a rounding-off algorithm and MAM with coordinate search algorithm are shown in Table 14. It can be seen that the designs obtained by both MAM with a rounding-off algorithm and coordinate search algorithm are similar but the latter has produced a slightly better result in terms of structural weight. An alternative design by binary GA is also given with the acceptable weight (0.0980). In comparison with these three designs, the correlation between number of helical rib pairs around the circumference ( $n$ ) and frame pitch ( $d$ ) can be identified: when  $n$  is increased in the design,  $d$  is increased for generating large skin bays. Also,  $H_h$  is decreased for the light weight of helical ribs so that the same weight of the fuselage section can be kept. It is seen that all these three optimal designs have some common features: The helical ribs have a tall and slender hat-shaped cross section with a thickness of 0.60 mm, which is the minimal allowed value of 0.60, and a height close to the maximum allowed value of 30 mm. This had led to a large moment of inertia and thus to a high bending stiffness. The circumferential frames, which are less instrumental in preventing fuselage bending have become thin and small, with both of their dimensions reaching the minimal bounds of 1.0 mm and 50.0 mm respectively. This case study has shown that the fuselage barrel is generated with large skin bays; few thin tall helical ribs; and few thin small circumferential frames. By altering the frame pitch, the height of the triangular skin bay is affected. Similarly, the number of helical ribs changes the width of the base of these triangular bays, see Figure 6. Consequently, these two variables change the area and the angle  $2\phi$  of these skin bays, and thus are - along with the skin thickness - instrumental in influencing the buckling behaviour of the barrel structure. In Table 15, the validation by FE analysis has been given for the final optimal design. The optimal result by MAM with coordinate search algorithm is validated by FE analysis in terms of normalized strains and weight. It is concluded that the predicted optimal design and FE simulation have a high degree of agreement.

Table 14. Results of weight optimization of the lattice fuselage barrel

Design variable	GA [17]	MAM with coordinate search algorithm	MAM with a rounding-off algorithm
Skin thickness ( $h$ )	2.08	2.09	2.10
Number of helical rib pairs around the circumference, ( $n$ )	60.00	66.00	67.00
Helix rib (hat-section) thickness, ( $t_h$ )	0.60	0.60	0.60
Helix rib (hat-section) height, ( $H_h$ )	27.90	24.34	23.77
Frame pitch, ( $d$ )	627.70	650.00	650.00
Frame (z-section) thickness, ( $t_f$ )	1.00	1.00	1.00
Frame (z-section) height, ( $H_f$ )	50.00	50.00	50.00
Objective (normalized weight)	0.0980	0.0978	0.0979



Table 15. Optima obtained using normalized strain constraints

Model	Normalized Tensile strain	Normalized Compressive strain	Normalized Shear strain	Normalized mass
MAM with coordinate search algorithm	0.00	0.00	1.42	0.0978
Validation by finite element analysis	0.02	0.00	1.42	0.1000

## 8. Conclusions

MAM with the discrete capability for solving mixed integer-continuous optimization problems was successfully demonstrated on a benchmark problem of a ten-bar truss structure and weight optimization of a lattice composite fuselage structure. The proposed method in this paper, MAM with the implementation of the coordinate search algorithm, has successfully solved the benchmark problem and applied for an aerospace application. By evaluating the effect of the size of discreteness for discrete design variables on the optimal designs, it indicates that MAM with the implementation of the coordinate search algorithm is a robust method and it outperforms a GA not only in the efficiency but also in the quality of the obtained solution; MAM with the implementation of the simple rounding-off method can not solve the optimization problem as efficiently and precisely as the coordinate search algorithm due to its simple search mechanism, but it produces the same level of quality of the results as GA, while requiring much fewer response analyses and less computational cost. The developed discrete capability in MAM is currently used in industrial projects where it is essential to consider the mixed integer-continuous problem formulations.

## References

- [1] R.T. Haftka, J.A. Nachlas., L.A. Watson, T. Rizzo, R. Desai, Two Point Constraint Approximations in Structural Optimization, *Comput. Methods Appl. Mech. Engrg.*, vol. 60, pp. 289-301, 1987.
- [2] G.M. Fadel, M.F. Riley, J.M. Barthelemy, Two Point Exponential Approximation Method for Structural Optimization, *Struct. Multidisc. Optim.*, vol. 2, pp. 117-124, 1990.
- [3] L.P. Wang, R.V. Grandhi, Improved Two-point Function Approximations for Design Optimization, *AIAA J.*, vol. 33, pp. 1720-1727, 1995.
- [4] V.V. Toropov, Simulation Approach to Structural Optimization, *Struct. Multidisc. Optim.*, vol. 1, pp. 37-46, 1989.
- [5] F. van Keulen, V.V. Toropov, New Development in Structural Optimization Using Adaptive Mesh Refinement and Multipoint Approximations, *Eng. Opt.*, vol. 29, pp. 217-234, 1997.
- [6] A. Polynkin, V.V. Toropov, Mid-range Metamodel Assembly Building Based on Linear Regression for Large Scale Optimization Problems, *Struct. Multidisc. Optim.*, vol. 45, pp. 515-527, 2012.
- [7] F.A.C. Viana, R.T. Haftka, Using Multiple Surrogates for Metamodeling, *Proceedings of 7th ASMO-UK/ISSMO International Conference on Engineering Design Optimization*, pp. 132-137, Bath, UK, July 7-8, 2008.
- [8] E. Acar, M. Rais-Rohani, Ensemble of Metamodels with Optimized Weight Factors, *Struct. Multidisc. Optim.*, vol. 37, pp. 279-294, 2009.
- [9] V.O. Balabanov, G. Venter, Response Surface Optimization with Discrete Variables, *Proc. 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Palm Springs, CA, AIAA 2004-1872, April 19-22, 2004.
- [10] M. Stolpe, To be or Not To Bee - comments on Discrete Optimum Design of Truss Structures Using Artificial Bee Colony Algorithm, *Struct. Multidisc. Optim.*, vol. 44, pp. 707-711, 2011.
- [11] T.G. Kolda, R.M. Lewis, V. Torczon, Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods, *SIAM Rev.*, vol. 45, pp. 385-482, 2003.
- [12] D. Liu, V.V. Toropov, Multipoint Approximation Method for Design Optimization with Discrete Variables, *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization (MAO) Conference*, Indianapolis, IN, 17-19 September, 2012.
- [13] R.T. Haftka, Z. Gürdal, *Elements of Structural Optimization*, 3rd ed., Kluwer Academic Publishers, 1992.
- [14] EU FP7 Research Project, Advanced Lattice Structures for Composite Airframes (ALasCA), Project Reference: 265881, 2010-2013, Available: [http://cordis.europa.eu/projects/rcn/97744\\_en.html](http://cordis.europa.eu/projects/rcn/97744_en.html).
- [15] H. Lohse-Busch, C. Hühne, D. Liu, V.V. Toropov, U. Armani, Parametric Optimization of a Lattice Aircraft Fuselage Barrel Using Metamodels Built with Genetic Programming, in: *Proceedings of the Fourteenth International Conference on Civil, Structural and Environmental Engineering Computing*, Paper 230, Civil-Comp Press, Stirlingshire, UK, 2013.
- [16] G.E.P. Box, N.R. Draper, *Empirical Model-building and Response Surfaces*, New York: John Wiley and Sons, 1987.

- [17] HyperStudy V12.0, Altair Engineering Inc, USA, 2012.
- [18] P. Hajela, J. Yoo, Constraint Handling in Genetic Search Using Expression Strategies, AIAA J., vol. 134, pp. 2414-2420, 1996.
- [19] S.Y. Mahfouz, Design Optimization of Structural Steelwork, PhD thesis, University of Bradford, Bradford, UK, 1999.
- [20] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT press, Cambridge, USA, 1992.
- [21] V.V. Toropov, A. Polynkin, U. Armani, L.F. Alvarez, Application of Metamodel Building by Genetic Programming to Industrial Optimization Problems, IV European Congress on Computational Mechanics (ECCM IV): Solids, Structures and Coupled Problems in Engineering, Paris, May, 2010.