Int. J. Simul. Multidisci. Des. Optim. **2**, 177–185 (2008)

DOI: 10.1051/ijsmdo:2008024

Available online at:
http://www.ijsmdo.org

Computational system for multi disciplinary optimization at conceptual design stage

P. Fantini[1], L.K. Balachandran[2] and M.D. Guenov[2,a]

[1] Computational Aerodynamics, Aircraft Research Association Ltd., Bedford, UK
[2] Aerospace Engineering, Cranfield University, Bedford, UK

Received 30 March 2008, Accepted 15 Juin 2008

Abstract – Presented is a computational system for multidisciplinary design optimization (MDO) at the conceptual design stage. During this phase, hundreds of low-fidelity models such as equations and compiled code, and thousands of variables are used to describe a complex product such as aircraft. In this context the paper presents a novel computational approach associated with the complete MDO process. The first aspect of the proposed approach is the dynamic derivation of the optimal computational plan for each design study, given the designer's choice of independent variables. The second aspect is the effectiveness with which the trade-off landscape is obtained. This is crucial from an engineering point of view, since such information will be used for selecting a baseline design. The approach is demonstrated with an aircraft design test case consisting of 96 models and 120 variables.

Key words: Multidisciplinary Design and Optimization (MDO); multi-objective optimization, computational process planning; design structure matrix (DSM), model based design, aircraft conceptual design.

1 Introduction

The aim of the conceptual phase of the design of complex products such as aircraft is to deliver a sound baseline which is then 'frozen' and passed onto the subsequent more detailed design stages. This is a crucial phase since the decisions taken at this stage commit the majority of the product lifecycle costs. It is therefore essential that the underpinning computational processes offer maximum flexibility in creating appropriate performance studies. This includes the dynamic assembly of a computational plan according to the designer's choice of input (independent) variables and also performing effective trade-off analysis for selecting the baseline design.

In this context the objective of this paper is to present a novel computational approach associated with this process. The first aspect of the proposed approach is the dynamic derivation of the optimal computational plan for each design study, given the designer's choice of independent variables. This is not a trivial task since during the conceptual phase, hundreds of low-fidelity models such as equations and compiled code, and thousands of variables are used to describe the complex product. The second aspect is the effectiveness with which the trade-off landscape is obtained. This is crucial from an engineering point of view since the decision maker (DM) needs this information for selecting the baseline design. It also implies that the DM should be able to consider all solutions of potential interest, even local ones, since these could become very attractive when taking into account decision factors not initially taken into consideration, for example, design robustness.

The following section briefly describes the developed procedure for producing an optimal computational process plan. Section three presents a novel method for generating well distributed points describing the entire Pareto frontier, as well as the local Pareto frontiers. Section four presents test results from our ongoing work, aiming at integrating the computational process with multi-objective optimization and uncertainty analysis. The example uses an aircraft design test case consisting of 96 models and 120 variables. Finally conclusions are drawn and future work outlined.

2 Computational process plan

Computational process modelling is the process of organising a complex system of models, in order to compute the output variables, according to specific input variables. An optimised computational plan for executing the models has to be attained each time the designer wishes to perform a computation associated with a different choice of input variables. To obtain such an optimal computational plan a novel procedure has been developed which is briefly explained as follows (Fig. 1):

1. Once the designer has chosen the independent variables, variable flow modelling is performed using the

[a] Corresponding author: m.d.guenov@cranfield.ac.uk

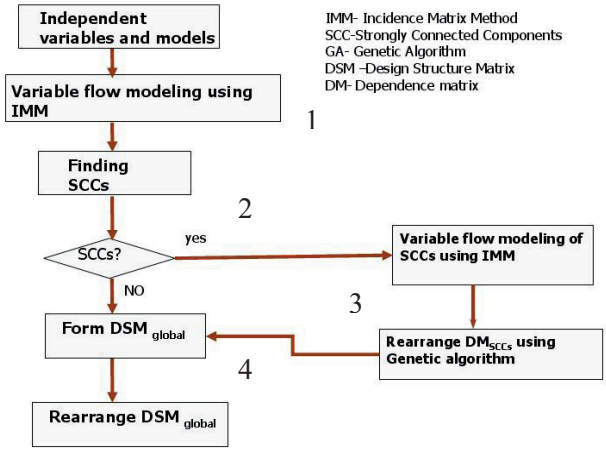


Fig. 1. Flow chart for process management.

incidence matrix method (IMM) to determine the information (data) flow between the models.
2. The next step is to establish the existence of any strongly connected components (SCCs). A SCC is a set of models which are strongly coupled through shared variables.
3. If SCCs exist then variable flow modelling of the models belonging to each SCC is performed. The different variable flow models obtained for each SCC are then populated in corresponding dependence matrices also known as design structure matrices or DSM [1]. Each dependence matrix is rearranged using a genetic algorithm, with feedback length as the objective function to be minimized. The variable flow model with the minimum feedback length is selected for further solving.
4. The SCCs (if any) and the remaining models are now populated in a global DSM. This matrix is further rearranged into upper triangular form, which determines the optimal sequence for the execution of the models.

The following sub-sections explain the procedure in more detail.

2.1 Variable flow modelling using Incidence matrix method

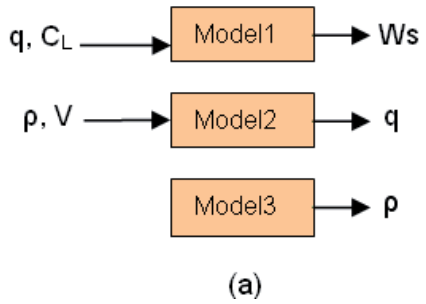
Variable flow modelling is the process of identifying the information flow between the models in the system, according to the input (independent) variables selected by the designer. The information flow between the models is required in order to determine the output variables to be computed, according to the input variables provided by the designer. This subsection explains a novel incidence matrix method (IMM) which allows to obtain the information flow dynamically.

The incidence matrix is used to represent the relationship between the models and their associated variables. The rows of the matrix represent the models, while the columns denote the variables. The association of a variable in a column with a model in a row is denoted by '*' in the corresponding entry (cell). For solving the incidence matrix, the *'s in each cell have to be substituted either with an 'i' (input) or with an 'o' (output), depending on whether the variable in the column should be an input to or an output from the model in the row. The incidence matrix is solved or populated according to the rules stated below.

1. An independent variable should be always an input to a model. (This implies that all the '*'s in the column of the independent variable should be replaced by 'i's.)
2. If a variable is associated only with one model and if it is not an independent variable, it should be an output from that model. In the same way if a model is associated with only one variable, that variable should be the output from that model (This implies that if a column/row has only one '*' marked in it and if the corresponding data variable in the column is not an independent variable, then the '*' should be replaced with an 'o'.)
3. Each variable should be output from only one model. (This implies that except for the columns of the independent variables, all the other columns should have an 'o' in only one cell.)
4. The number of outputs identified for a particular model as a result of variable flow modelling should correspond to the number of outputs from the original model. (This implies that every row should have the same number of 'o's as the number of outputs from the associated model)
5. After the four rules above have been applied to a SCC it is possible that there will be some models remaining, for which not all '*' have been replaced with either an 'i' or an 'o'. Of these, the models with inputs differing from the original ones are selected for further population. If no such a model exists, the incidence matrix is populated with the original inputs and outputs from the model.

A basic example demonstrating the above rules is shown in Figure 2 where a simple set of models for balancing the weight and lift of an aircraft is converted into its corresponding initial incidence matrix. Data variables entering the models are the inputs and data variables leaving the models are the outputs. In this example Ws and V are considered the independent variables selected by the designer.

The final populated incidence matrix is shown in Figure 3, after applying the rules stated earlier. The numbers in the curly brackets represent the sequence in which the matrix was populated: {1}-Rule1, {2}-Rule2, {3}-Rule-3, {4}-Rule4, {5}-Rule3 and {6}-Rule4. Thus the final outcome of incidence matrix method is: model1 has Ws and q as input and C_L as output, model2 has ρ and V as input and q as output, and finally, model3 produces ρ as output.



(a)

	Ws	C_L	q	ρ	V
Model1	*	*	*		
Model2			*	*	*
Model3				*	

(b)

Fig. 2. (a) Models balancing the weight of aircraft with its lift, (b) Initial incidence matrix.

	Ws	C_L	q	ρ	V
Model1	i {1}	o {6}	i {5}		
Model2			o {4}	i {3}	i {1}
Model3				o {2}	

Fig. 3. Final populated incidence matrix.

2.2 Variable flow modelling in strongly connected components

A strongly connected component (SCC) is a set of models which are strongly coupled through shared variables. The term SCC is derived from graph theory and more formally, an SCC is a subset of nodes in a directed graph such that there is a path from every node to every other node [2]. In the context of a system of models, a SCC is a cluster of models in which each model requires input from one or more models from the same cluster. Presence of SCCs in a system adds complexity to the computational process modelling and also to the execution and solving of the models belonging to the same SCC.

In the above simple example only the first four rules were applied to populate the matrix. Figure 4 shows a system of models with its corresponding populated incidence matrix with X_3 chosen as the independent variable. It can be noted that there are some cells in which the '*' is not yet replaced with either an 'i' or an 'o'. This situation arises as a result of the presence of a SCC. In the figure, the models which belong to the SCC are those which have at least one variable undefined (i.e. still labelled as '*'), after applying the first four rules. Thus in this example, models 1, 2, 3, 5 and 6 are strongly connected and are coupled through variables X_1, X_4, X_5, X_6, and X_7. To generalise, the models of a SCC have their own unique set of data variables through which they are coupled. A design problem can have more than one mutually exclusive SCCs.

Returning to the example (Fig. 4), the solution of the incidence matrix of the SCC is obtained by choosing the outputs from the constituent models according to rule 5. According to this rule model6 is chosen for further solving, since X_3 is an output of the original model (Fig. 4a), but after applying the four rules X_3 has become an input (Fig. 4b). Given the choice of model6, then variables X_1, X_5 or X_6 can now be selected as the preferred output. The set of variable flow models obtained for each case is shown in Figure 5.

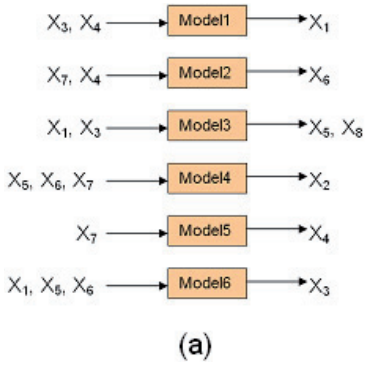
2.3 Rearrangement of SCCs using genetic algorithm

The next step following the variable flow modelling is to schedule the model execution sequence while accounting for any feedback loops. Feedback loops are generated when models require inputs from other models which are run further down in the execution sequence. Thus the aim of the rearrangement is to reduce the length and number of feedback loops or to eliminate them altogether if possible. In complex design systems a complete elimination of iterative feedback loops is not always possible, especially in the SCCs. Here we propose an optimisation based approach for rearranging the SCCs which utilises a genetic algorithm. The objective function which the genetic algorithm minimises is shown in equation (1), where $DX(i,j)$ denotes the value of the (i,j)th element of the dependence matrix DX, J is the feedback length and n is the number of models.

$$J = \sum_{i=2}^{n} \sum_{j=1}^{i=1} DX(i,j) \cdot (i-j) + \sum_{k=1}^{n} DX(k,k). \qquad (1)$$

The feedback length of the dependence matrix represents an approximate estimate of the time required for solving the SCCs if a fixed point iterative scheme is applied. Figure 6 shows the dependence matrix for the variable flow model in Figure 5a.

In the matrix, '1' marked in a cell above the main diagonal symbolises a feed forward loop and a '1' below the main diagonal represents a feedback loop. In addition, '0.5' marked in a cell on the main diagonal itself denotes that the model in the corresponding row has its inputs and outputs interchanged as a result of variable flow modelling. In such cases numerical methods have to be applied to solve those particular models and the value '0.5' signifies the additional time required for solving these. Each of the dependence matrices for the variable flow models in Figure 5 is rearranged using genetic algorithm with equation (1) as the fitness function to be minimized. The value of the objective function is considered as the criteria for choosing the variable flow model which leads to faster convergence of the SCC. After rearranging the dependence matrix of each of the four variable flow models



	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	*		i	*				
Model2				*		*	*	
Model3	*		i		*			o
Model4		o			i	i	i	
Model5				*			*	
Model6	*		i		*	*		

(b)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	i		i	o				
Model2				i		o	i	
Model3	i		i		o			o
Model5				i			o	
Model6	o		i		i	i		

(a)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	i		i	o				
Model2				i		o	i	
Model3	o		i		i			o
Model5				i			o	
Model6	i		i		o	i		

(b)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	o		i	i				
Model2				o		i	i	
Model3	i		i		o			o
Model5				i			o	
Model6	i		i		i	o		

(c)

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Model1	o		i	i				
Model2				i		i	o	
Model3	i		i		o			o
Model5				o			i	
Model6	i		i		i	o		

(d)

Fig. 5. Incidence matrices with the following variables chosen as output from model6: (a) X_1; (b) X_5; (c) X_6 then X_4; (d) X_6 then X_7.

from the example, the corresponding minimised feedback lengths are: 5.5, 6, 5.5 and 5. This shows that the fourth variable flow model (Fig. 5d) will converge faster than the rest and therefore will be chosen as the solution. The final rearranged dependence matrix of the fourth variable flow model is shown in Figure 7.

2.4 Global rearrangement

After the rearrangement of a SCC is completed, it is considered as single sub-system and is reintroduced into the global Design Structure Matrix (DSM) along with the remaining models which were not part of any SCC. By applying the graph theoretical algorithm of Tang et al. [3], the global DSM is rearranged into an upper-triangular form, thus ensuring all loops are feed forward. Hence, the final model computation sequence obtained for the above example will be: SCC (1→3→5→2→6) → 4. A more detailed description of the procedure with examples can be found in [4].

3 Double hyper cone boundary intersection method

When performing a multi-objective optimization the first task of the DM is that of setting up the optimization problem. This involves defining design variables, constraints

	Model1	Model2	Model3	Model5	Model6
Model1	0.5	1	0	1	0
Model2	0	0	0	0	1
Model3	0	0	0	0	1
Model5	0	1	0	0.5	0
Model6	1	0	1	0	0.5

Fig. 6. Dependence matrix for the incidence matrix from Figure 5a.

and objectives. An optimization algorithm will then be applied in order to obtain a set of Pareto solutions. Finally the DM will have to evaluate the results, eventually making additional considerations not originally taken in account when setting up the optimization problem. When the DM is unsure of how to define the optimization problem it can be useful to identify all the solutions of potential interest, rather than only those that are optimal with respect to a particular setup of the optimizer. The local Pareto frontier concept will be used for this purpose.

3.1 The algorithm

The Double Hyper-cone Boundary Intersection (DHCBI) method has been developed for obtaining the global as well as the local Pareto frontiers [5]. The DHCBI method shares conceptual similarities with the NBI and NC methods [6–10] for which an optimization is performed for each

	Model1	Model3	Model5	Model2	Model6
Model1	0	1	0	0	1
Model3	0	0	0	0	1
Model5	1	0	0	1	0
Model2	0	0	1	0.5	0
Model6	0	0	0	1	0.5

Fig. 7. Rearranged dependence matrix of variable flow model of Figure 5d.

of the Pareto points sought. However, as it will be described below, the DHCBI is designed to improve the effectiveness of the search.

For the optimization problem:

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathbf{F}(\mathbf{x}) \\ &\text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}) \leqslant 0, \\ &\qquad\qquad k = 1, 2, ..., K \\ &\text{and } P \text{ equality constraints: } h_p(\mathbf{x}) = 0, p = 1, 2, ..., P \end{aligned} \tag{2}$$

where $F_i : R^N \rightarrow R^M$, the NBI and NC algorithms can be summarised as follows:

1. A single-objective minimization is performed for each of the M objectives yielding, as termed by Mattison et al. [11], *anchor points*, $\boldsymbol{\mu}_i \in R^M$, $i \in \{1, ..., M\}$. Das and Dennis [6] refer to the polygon of vertices $\boldsymbol{\mu}_i$, $i \in \{1, ..., M\}$ as *Convex Hull of Individual Minima* (CHIM). Messac et al. [8] terms *utopia plane*, the hyper-plane to which the M anchor points belong.
2. In the NBI method, K evenly distributed *utopia plane points* $\mathbf{p}_k$, belonging to the polygon of vertices $\boldsymbol{\mu}_i$, are defined as:

$$\mathbf{p}_k = \sum_{i=1}^{M} \alpha_{ki} \boldsymbol{\mu}_i, \quad k = \{1, 2, ..., K\}. \tag{3}$$

In the NBI method α_{ki} is such that $0 \leqslant \alpha_{ki} \leqslant 1 \quad \forall k, i$ and $\sum_{i=1}^{M} \alpha_{ki} = 1$. In contrast, the NC method takes in consideration the fact that for $M > 2$ the projection of portions of the Pareto front on the utopia plane is external to the polygon of vertices $\boldsymbol{\mu}_i$ [6, 9]. This region will be referred to as *peripheral region*. In order to deal with this problem Messac et al. solve a set of computationally benign optimizations, which allow them to obtain the lower and upper limits of α_{ki} while still imposing $\sum_{i=1}^{M} \alpha_{ki} = 1$.

3. For utopia plane point $\mathbf{p}_k$, the optimization problem is reformulated and solved yielding a Pareto point.

In the NBI method the kth optimization problem is reformulated as follows:

$$\begin{aligned} &\min_{\mathbf{x}} \ t(\mathbf{x}) \\ &\text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}) \leqslant 0, \\ &\qquad\qquad k = 1, 2, ..., K, \\ &\text{subject to } P \text{ equality constraints: } h_p(\mathbf{x}) = 0, \\ &\qquad\qquad p = 1, 2, ..., P \\ &\text{and subject to the additional equality constraint:} \\ &\qquad\qquad \mathbf{p}_k + t\mathbf{n} = \mathbf{F}(\mathbf{x}) \end{aligned} \tag{4}$$

where $\mathbf{n}$ is the normal to the utopia plane and t is the projection of $\mathbf{F}(\mathbf{x})$ on $\mathbf{n}$. The kth optimization problem for the NC method is instead the following:

$$\begin{aligned} &\min_{\mathbf{x}} \ f_M(\mathbf{x}) \\ &\text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}) \leqslant 0, \\ &\qquad\qquad k = 1, 2, ..., K, \\ &\text{subject to } P \text{ equality constraints: } h_p(\mathbf{x}) = 0, \\ &\qquad\qquad p = 1, 2, ..., P \\ &\text{with the M} - 1 \text{ additional equality constraint:} \\ &\qquad \mathbf{q}_j . (\mathbf{p}_k - \mathbf{F}(\mathbf{x})) \leqslant 0, \forall j : \ j \in \{1, 2, ..., M-1\} \end{aligned} \tag{5}$$

where $\mathbf{q}_j = \frac{\boldsymbol{\mu}_M^* - \boldsymbol{\mu}_j^*}{\|\boldsymbol{\mu}_M^* - \boldsymbol{\mu}_j^*\|}$ are $M-1$ unit vectors.

Even though the DHCBI approach is similar to the NBI and NC methods, it differs in two main aspects, one relative to the reformulation of each optimization problem and one relative to the *anchor points*.

3.2 DHCBI reformulation for solving sub-problem k

We propose the following alternative reformulation for sub-problem $k$with respect to utopia plane point $\mathbf{p}_k$ (Fig. 8):

$$\begin{aligned} &\min_{\mathbf{x}} \left[t(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) + q(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) \right] \\ &\text{subject to } K \text{ inequality constraints:} \\ &\qquad\qquad g_k(\mathbf{x}) \leqslant 0, k = 1, 2, ..., K, \\ &\text{subject to } P \text{ equality constraints:} \\ &\qquad\qquad h_p(\mathbf{x}) = 0, p = 1, 2, ..., P \\ &\text{and subject to the additional constraint: } c \leqslant 0 \end{aligned} \tag{6}$$

where $\bar{f}_i(\mathbf{x}) = (f_i(\mathbf{x}) - \mu_{ii}^*)/(\max\{\mu_{1i}^*, \ldots, \mu_{Mi}^*\} - \mu_{ii}^*)$,

$$\gamma_{min} = \gamma_{min}\left(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k\right) = \min\{\gamma_1, \gamma_2\} \tag{7}$$

with

$$\begin{aligned} \gamma_i &= \arccos\left(\frac{\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k}{\left\|\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k\right\|} \cdot \mathbf{l}_i \right), \\ t(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) &= \left\|\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k\right\| \cos(\gamma_{min}), \\ n(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) &= \left\|\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k\right\| \sin(\gamma_{min}), \\ q(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) &= q(n) \end{aligned} \tag{8}$$

and

$$c = \begin{cases} t_c \frac{e^{\frac{\ln(\tan\gamma_c + 1)}{t_c} t} - 1}{\ln(\tan\gamma_c + 1)} - t + n_c & t \leqslant t_c \\ (t - t_c)\tan\gamma_c + t_c \left(\frac{\tan\gamma_c}{\ln(\tan\gamma_c + 1)} - 1 \right) + n_c & t > t_c \end{cases} \tag{9}$$

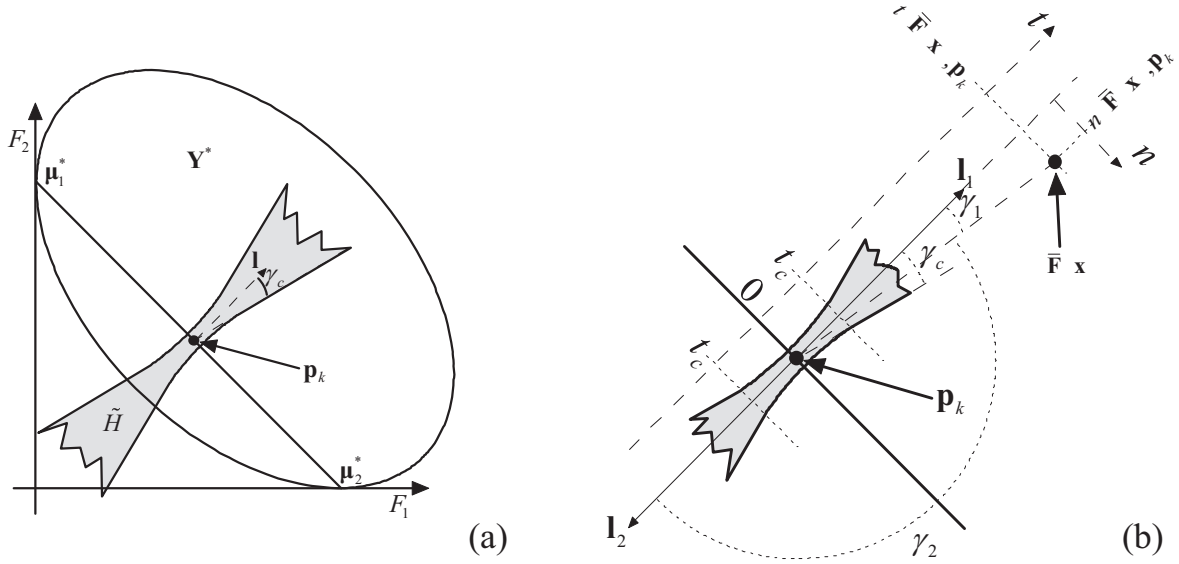


Fig. 8. DHCBI: a) problem formulation; b) detail of the DHCBI formulation.

where n_c is a fraction of the Euclidean distance between two contiguous utopia plane points, γ_c is the cone angle and $t_c = \frac{n_c}{\tan\gamma_c}$ is the distance from $\mathbf{p}_k$ along $\pm\mathbf{l}$ at which the constraint becomes a hyper-cone. The formulation of the additional constraint, c, allows maintaining the continuity of the first derivative $t = 0$ and $t = t_c$:

$$c'(t) = \begin{cases} e^{\ln(\tan\gamma_c+1)\frac{t}{t_c}} - 1 & t \leqslant t_c \\ \tan\gamma_c & t > t_c. \end{cases} \quad (10)$$

In its formulation the DHCBI method (6) differs from the NBI method (4), in that the additional equality constraint of the NBI method is substituted with the inequality constraint $c(9)$.This results in an increased size of the feasible region with respect to the NBI approach. Furthermore, while the NC method 3.2 requires $M-1$ additional constraints, the DHCBI requires just one.

Finally function $q(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k)$ in (6) and (8) is chosen as:

$$q(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) = q\,(n) = \begin{cases} \frac{n^{c+1}}{n_c^c}, & n \leqslant n_c \\ (1-c)n + \frac{cn^2}{n_c}, & n > n_c \end{cases} \quad (11)$$

to direct the solver towards the axis of the hyper-cone.

For obtaining a complete representation of the Pareto frontier it is necessary to obtain also those Pareto points that belong to the peripheral region. Here we propose an alternative approach for obtaining such points with respect to the one proposed by Messac and Mattson [9]. Since the points belonging to the peripheral region are external to the polygon of vertices $\boldsymbol{\mu}_i^*$, a possible solution is that of generating the peripheral region utopia plane points $\mathbf{p}^+$ from the utopia plane points belonging to the edges of the polygon.

Let us consider an utopia plane point $\mathbf{p}_k^*$ belonging to the kth edge of the polygon identified by the two anchor points μ_{j+1} and μ_j. Then $\mathbf{p}_k^* = \sum_{i=1}^{M} \alpha_{ki}\mu_i^*$ with $\alpha_{ki} = 0$ for $i \neq j, j+1$ and $0 < \alpha_{ki} < 1$ for $i = j, j+1$. Similarly to [10], a vector $\mathbf{s}$ lying in the utopia plane, orthogonal to the kth edge and pointing outwards with respect to the polygon, can be defined as:

$$\mathbf{s} = \frac{\boldsymbol{\nu}_{j-1} + \beta_j\boldsymbol{\nu}_j}{|\boldsymbol{\nu}_{j-1} + \beta_j\boldsymbol{\nu}_j|}, \quad \text{with} \quad \beta_j = -\frac{(\boldsymbol{\nu}_{j-1}, \boldsymbol{\nu}_j)}{(\boldsymbol{\nu}_j, \boldsymbol{\nu}_j)} \quad (12)$$

where $\boldsymbol{\nu}_j = \boldsymbol{\mu}_{j+1}^* - \boldsymbol{\mu}_j^*$.

Then for each of the utopia plane points $\mathbf{p}_k^*$ belonging to an edge, the peripheral region utopia plane points can be obtained as:

$$\mathbf{p}^+ = \mathbf{p}_k^* + kn_d\mathbf{s}, \quad k = \{1, 2, ...\} \quad (13)$$

where n_d corresponds to the distance between to adjacent anchor points $\mathbf{p}^+$ and k is increased until the optimizer fails to obtain a solution. By following this approach, each new peripheral region utopia plane point $\mathbf{p}^+$ is generated from an initial utopia plane point $\mathbf{p}_k^*$, moving orthogonally to the edge of the polygon to which $\mathbf{p}_k^*$ belongs and parallel to the utopia plane.

3.3 Local Pareto frontiers

With regard to the local and global minima of a single-objective optimization problem, we can define as local Pareto frontier, a Pareto frontier within a particular neighbourhood of the design space. The procedure presented bellow shows how such frontiers can be obtained. The procedure has two main advantages, the first is that a more complete investigation of the design space is performed improving the chances of obtaining the global Pareto frontier. Secondly, it allows the DM to obtain and analyse a set of solutions that can be of interest from an engineering point of view, for example, when accounting for additional considerations such as design robustness.

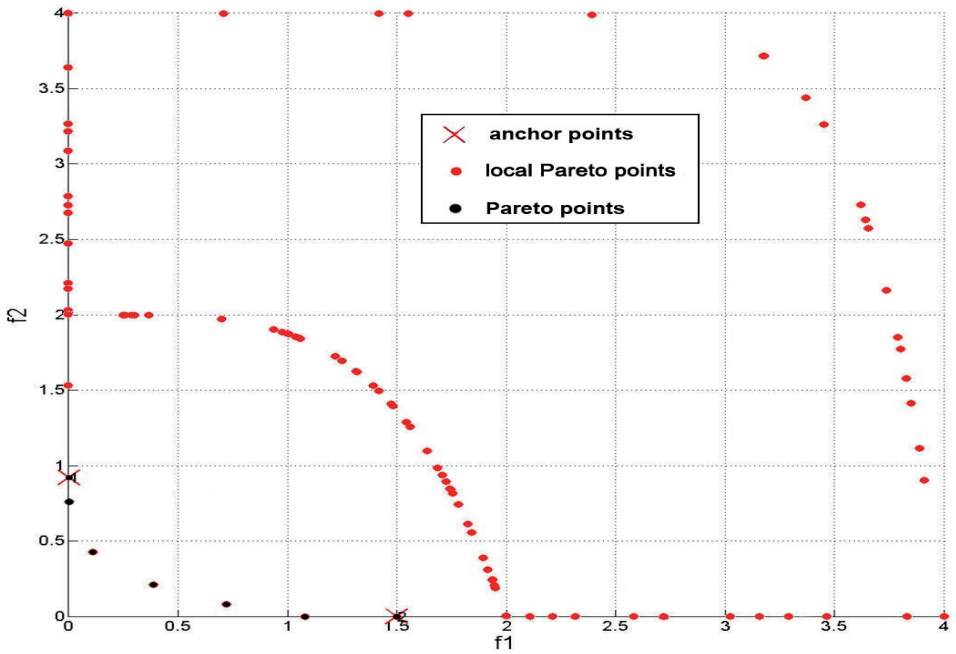


Fig. 9. Criterion space.

The procedure for obtaining the local Pareto frontiers is the following:

1. A Latin-hypercube sampling procedure is used for obtaining L well distributed points in the design space.
2. L single-objective minimizations are performed for each of the M objectives, yielding the *local anchor points* $\boldsymbol{\mu}_{ij} \in R^M$, $i \in \{1, ..., M\}$, $j \in \{1, ..., L_i\}$, where L_i is the number of local minima found for objective i.
3. Among the $\sum_{i=1}^{M} L_i$ local minima, the M *global anchor points* $\boldsymbol{\mu}_i \in R^M$, $i \in \{1, ..., M\}$ are identified.
4. For objective i and local minima j, *local anchor points* $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{ij}, ..., \boldsymbol{\mu}_M$ are used for defining the K utopia plane points:

$$\mathbf{p}_{kij} = \sum_{l=1, l\neq i}^{M} \alpha_{kl}\boldsymbol{\mu}_l + \alpha_{ki}\boldsymbol{\mu}_{ij}, \quad k = \{1, 2, ..., K\}. \tag{14}$$

5. For utopia plane point $\mathbf{p}_{kij}$ sub-problem k is reformulated and solved.
6. Steps 4 through 6 are repeated L_i times for each of the M objectives.

Similarly to the NBI and NC methods, the order in which the sub-problems are solved in the DHCBI is chosen to minimize the distance between each pair of subsequent utopia plane points considered. This allows for using the solution of the previous utopia plane point as a starting point for the subsequent sub-problem, thus minimizing the number of iterations required for convergence and producing a solution in its neighbourhood.

Starting the sequence of optimizations from the utopia plane point closest to the local anchor point $\boldsymbol{\mu}_{ij}$ and moving towards the other $M-1$ global anchor points allows to obtain the local Pareto in the neighbourhood of $\boldsymbol{\mu}_{ij}$.

3.4 Example of solution obtained with the DHCBI

The following test case is a multimodal multi-objective optimization problem developed by Deb [12] which has been widely used for evaluating evolutionary algorithms.

$$\min[F_1(x_1, x_2), F_2(x_1, x_2)] \quad \text{for} \quad 0 < x_i < 1 \tag{15}$$

where

$$F_1 = 4x_1, \quad F_1 = g_1 g_2,$$

$$g_1 = \begin{cases} 4 - 3e^{-\left(\frac{x_2 - 0.2}{0.02}\right)^2} & (\mathrm{x}_2 < 0.4) \\ 4 - 3e^{-\left(\frac{x_2 - 0.7}{0.02}\right)^2} & (\mathrm{x}_2 \geqslant 0.4) \end{cases} \quad \text{and}$$

$$g_2 = \begin{cases} 1 - \left(\frac{F_1}{g_1}\right)^{0.25 + 3.75(g_1 - 1)} & F_1 \leqslant g_1 \\ 0 \; F_1 > g_1. \end{cases} \tag{16}$$

Figure 9 shows the criterion space, including the various local Pareto frontiers obtained with the DHCBI method. The global Pareto frontier is identified by the black dots (lower left corner). It must be noted that weak Pareto points have not been removed. In this test case it can be demonstrated that the global Pareto frontier is extremely sensitive to variations of variable x_2. In such a case, if the DM wished to consider robustness, then solutions from the second best local Pareto frontier could be considered as a better choice.

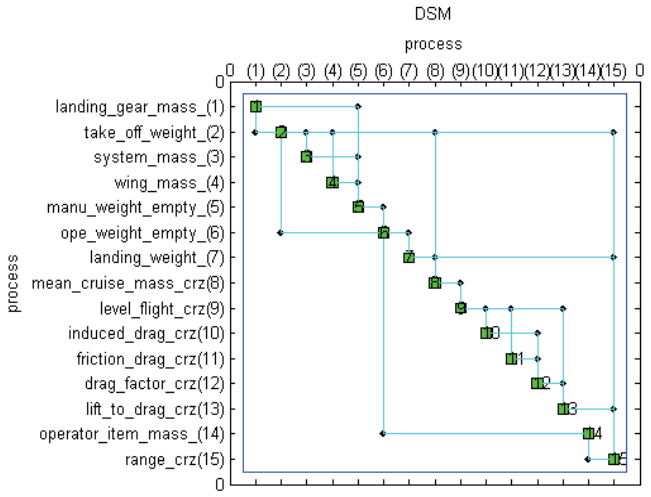


Fig. 10. Results from the computational flow modelling process. Due to the size of the problem, shown is only the Design Structure Matrix of the Strongly Connected Component with the shortest feedback length.

4 Test results

A simplified set of models which represent a real aircraft is considered for testing the computational process concept developed in this research. The test case contains 96 models and 126 variables. The multi-objective optimization problem is set as follows:

- Objectives: range (RA) to be maximized, takeoff weight (MTOW) to be minimized and climb rate (vzclb) to be maximized.
- Constraints: takeoff field length $\leqslant$ 2000 m, approach speed $\leqslant$ 120 kts, climb rate $\geqslant$ 500 ft/min, cruise thrust coefficient $\leqslant$ 1 and wing fuselage fuel ratio $\geqslant$ 0.75.
- Independent variables: engine thrust, wing span, sweep angle, thickness to chord, fuel quantity and wing span.
- Constant variables: number of first class passengers, number of standard class passengers, number of aisles, engine bypass ratio, number of engines, cruise altitude, cruise mach, takeoff altitude and approach altitude.

Following the procedure described in Figure 1, an optimal computational plan (process) for the test case was obtained. It was found that, out of the 96 models, 15 were strongly connected thus forming a SCC with 15 models. For doing this several candidate variable flow models were rearranged using a genetic algorithm to minimize the feedback length. Figure 10 shows the SCC with minimum feedback. Figures 11 and 12 show two views of the 30 Pareto points (each one representing an aircraft configuration) generated by the application of DHCBI on the optimal computational process.

While the emphasis of the current work on the DHCBI has been on the effectiveness (quality) of the solution, the focus of the work on process modeller has been on flexibility and efficiency. The extensive testing of the process modeller demonstrated that the selections which it made were always amongst the best in terms of convergence.

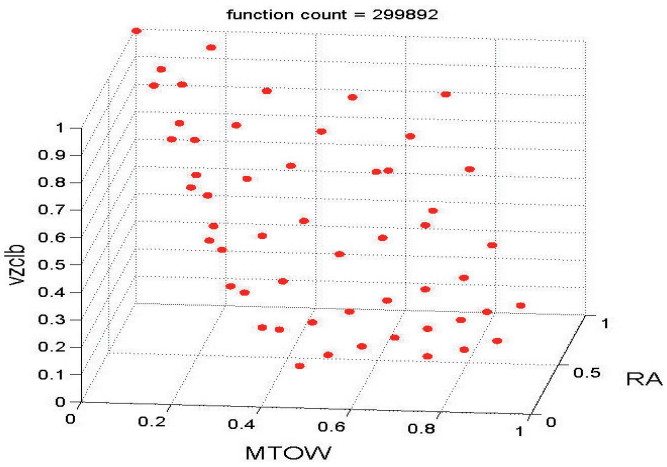


Fig. 11. Normalised Pareto frontier.

5 Conclusions

A novel approach for obtaining optimal computational plans for conceptual design studies is proposed in this paper. It incorporates a novel variable flow method based on the incidence matrix concept. Unlike other existing methods the variable flow method can handle multiple outputs of a particular constituent model as well as the identification of strongly connected components (SCCs) in the entire multidisciplinary set of models and equations describing the aircraft. Traditionally, variable flow modelling and rearrangement of SCCs are performed separately and the only link is the transfer of variable flow model results to the rearrangement process. Our approach is capable of exploring a number of feasible variable flow models according to the objectives of the particular design study. The computational process is being integrated with the proposed novel Double Hyper Cone Intersection Method (DHCBI) for obtaining the Pareto frontier. The advantage of this method is that the number of analyses to be performed increases automatically with the number of local minima of the objective functions. Thus the DHCBI algorithm adapts to the complexity of the problem to be solved. Since a local (gradient based) optimizer such as the sequential quadratic programming (SQP) algorithm needs to be used for obtaining local Pareto solutions, the determination of the gradients would increase the number of analysis to be performed. This problem could be alleviated by the application of automatic differentiation (AD) for obtaining partial derivatives.

Future work will advance further the research into convergence issues associated with the SCCs and the application of AD for robust optimization. Also a significant effort is underway to create an object-oriented conceptual design framework incorporating workflow management, multi-objective optimisation and uncertainty analysis.

Acknowledgements

This research was conducted as part of VIVACE (Value Improvement through a Virtual Aeronautical Collaborative

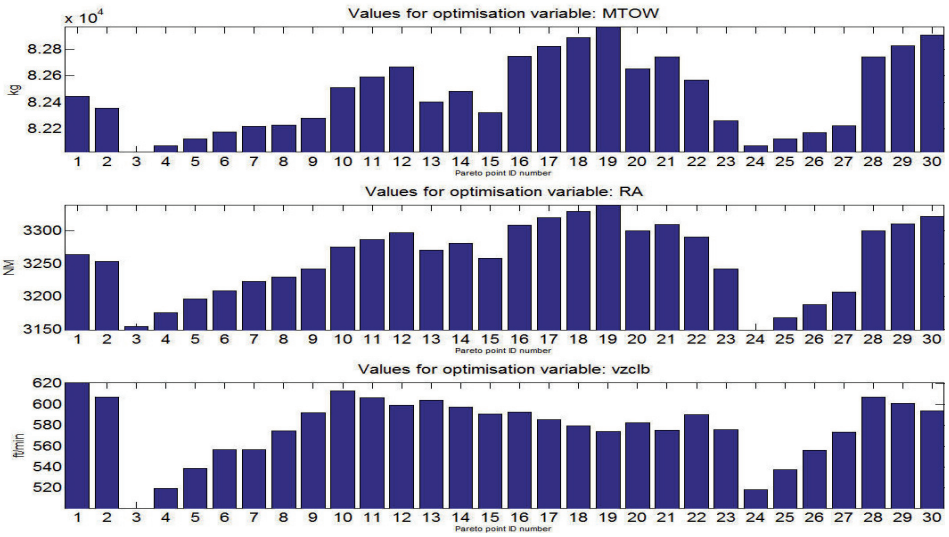


Fig. 12. Pareto frontier – absolute values by point.

Enterprise)- integrated project AIP3 CT-2003-502917, partly sponsored by the Sixth Framework Programme of the European Community under priority 4, "Aeronautics and Space".

References

1. D.V. Steward, *Systems Analysis and Management: Structure, Strategy and Design* (Petrocelli Books, Inc., New York, 1981)
2. M.J. Buckley, K.W. Fertig, D.E. Smith, Design sheet: an environment facilitating flexible trade studies during conceptual design, in *1992 Aerospace Design Conference*, 3–6 February, 1992 (Irvine, Ca), AIAA 92-1191
3. D. Tang, L. Zheng, Z. Li, Computers and Industrial Engineering **38**, 479 (2000)
4. M.D. Guenov, T.D. Libish, H. Lockett, Computational design process modelling, in *25th International Congress of the Aeronautical Sciences conferences (ICAS)*, 3–8 September, 2006 (Hamburg, Germany), ISBN 0-9533991-7-6
5. P. Fantini, L.K. Balachandran (Libish), M.D. Guenov, Computational Intelligence in Multi Disciplinary Optimization at Feasibility Design Stage, in *First International Conference on Multidisciplinary Design Optimization and Applications*, 17–20 April, 2007, Besancon, France (EDP Sciences), ISBN 978-2-7598-0023-0
6. I. Das, J.E. Dennis, SIAM J. Optimiz. **8**, 631 (1998)
7. I. Das, An Improved Technique for Choosing Parameters for Pareto Surface Generation Using Normal-Boundary Intersection, in *Third World Congress of Structural and Multidisciplinary Optimization (WCSMO-3),* edited by C.L. Bloebaum, et al., 17–21 May, 1999 (Buffalo, NY, University of Buffalo), Vol. 2, pp. 411–413
8. A. Messac, A. Ismail-Yahaya, C.A. Mattson, Structural and Multidisciplinary Optimization, Journal of the International Society of Structural and Multidisciplinary Optimization (ISSMO) **25**, 86 (2003)
9. A. Messac, C. Mattson, AIAA J. **42**, 2101 (2004)
10. S.V. Utyuzhnikov, P. Fantini, M.D. Guenov, Numerical Method for Generating the Entire Pareto Frontier in Multiobjective Optimization, in *EUROGEN 2005*, 12–14 September, 2005 (Munich, Germany) ISBN: 3-00-017534-2
11. C.A. Mattison, A.A. Mullur, A. Messac, Minimal Representation of Multiobjective Design Space Using Smart Pareto Filter, in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 4–6 September, 2002. Atlanta, GA, AIAA-2002-5458
12. K. Deb, *Multi-objective genetic algorithms: problem difficulties and construction of test problems*, Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, 1998