Rule Based Architecture for Collaborative Multidisciplinary Aircraft Design Optimisation

Nickolay Jelev, Andy Keane, Carren Holden, András Sóbester

Abstract**—**In aircraft design, the jump from the conceptual to preliminary design stage introduces a level of complexity which cannot be realistically handled by a single optimiser, be that a human (chief engineer) or an algorithm. The design process is often partitioned along disciplinary lines, with each discipline given a level of autonomy. This introduces a number of challenges including, but not limited to: coupling of design variables; coordinating disciplinary teams; handling of large amounts of analysis data; reaching an acceptable design within time constraints. A number of classical Multidisciplinary Design Optimisation (MDO) architectures exist in academia specifically designed to address these challenges. Their limited use in the industrial aircraft design process has inspired the authors of this paper to develop an alternative strategy based on well established ideas from Decision Support Systems. The proposed rule based architecture sacrifices possibly elusive guarantees of convergence for an attractive return in simplicity. The method is demonstrated on analytical and aircraft design test cases and its performance is compared to a number of classical distributed MDO architectures.

Keywords**—**Multidisciplinary design optimisation, rule based architecture, aircraft design, decision support system.

NOMENCLATURE

J_i	= Domain level objective function for Collaborative Optimisation and Analytical Target Cascading
J_{sys}	= Domain level objective function for Enhanced Collaborative Optimisation
$\mathbf{Z_{max}}$	= Vector of maximum magnitude preferred shared design variables
$\mathbf{Z_{min}}$	= Vector of minimum magnitude preferred shared design variables
g	= Additional domain constraints for Enhanced Collaborative Optimisation
c_i	= Domain specific constraints
f_0	= Global Optimisation Function
$\mathbf{lb}$	= Current vector of shared design variables lower bounds
lb_i	= Current lower bound for given shared variable
lb_{imd}	= Current lower bound for given shared variable after downward move
lb_{imu}	= Current lower bound for given shared variable after upward move
$\mathbf{lb_{init}}$	= Initial vector of shared design variables lower bounds
s	= Constraint slack variables for Enhanced Collaborative Optimisation
$\mathbf{ub}$	= Current vector of shared design variables upper bounds
ub_i	= Current upper bound for given shared variable
ub_{imd}	= Current upper bound for given shared variable after downward move
ub_{imu}	= Current upper bound for given shared variable after upward move
$\mathbf{ub_{init}}$	= Initial vector of shared design variables upper bounds
$\mathbf{x}$	= Vector of shared design variables
y	= Output from an analysis routine
$\mathbf{y}$	= Vector of state variables (variables that are output from one domain and input in one or more domains)
$\mathbf{z}$	= Vector of target variables
ϵ_x	= Slack variables for shared design vector in Analytical Target Cascading
ϵ_y	= Slack variables for state vector in Analytical Target Cascading
ϵ_c	= Convergence factor for Rule Based Architecture
ϵ_{mb}	= Bound movement factor for Rule Based Architecture
ϵ_{rb1}	= Primary bound reduction factor for Rule Based Architecture
ϵ_{rb2}	= Secondary bound reduction factor for Rule Based Architecture
λ_c	= Compatibility variable in Enhanced Collaborative Optimisation
λ_f	= Feasibility variable in Enhanced Collaborative Optimisation

I. INTRODUCTION

THE aerodynamic and structural design of wings has been the subject of interest for a considerable time as these disciplines typically tightly coupled. Traditionally aerodynamic design precedes structural optimisation because the aerodynamic loads on the wing are needed before the structural design can begin. This sequential approach was used in a number of modern clean sheet airliner designs including the B777 [1], A380 [2] and B787 [3]. In 1933, Prandtl [4] showed that when the aerodynamics and structures analyses were solved concurrently, the global drag for a given wing weight could be reduced beyond that achieved by the sequential method. This observation was further confirmed by other academics [5]–[7].

Numerous strategies (also defined as architectures [8]) capable of tackling such problems have been developed over the years. These are built on Multidisciplinary Design Optimisation (MDO) ideology that aims to exploit the coupling amongst disciplines, combined with numerical optimisation to generate an improved design. These architectures focus on process rather than the outcome in design optimisation, as the global minimum in the problems they aim to solve is often difficult to prove.

Over a dozen different architectures currently exist in academia. The interested reader is directed to the survey by

Andy Keane (Prof.) and András Sóbester (Dr.) are part of Computational Engineering and Design Research Group, within Faculty of Engineering and the Environment at the University of Southampton, UK.

Carren Holden (Dr.) is associated with Airbus Operations Ltd., Pegasus House, Aerospace Avenue, Filton, Bristol BS34 7PA, UK.

Nickolay Jelev is with the Computational Engineering and Design Research Group, within Faculty of Engineering and the Environment at the University of Southampton, UK (e-mail: N.Jelev@soton.ac.uk).

Martins & Lambe [8], which summarises many of the most well known architectures and reiterates several conclusions found in academic papers on the topic. Much of the research completed in this field tackles the problem in one of two contrasting approaches, monolithic and distributed. The former use a single optimiser that combines all involved disciplines together. The optimiser iterates through the discipline analyses until a minimum is reached. The latter approaches perform a distributed optimisation process using a multi-tier system of optimisers. Low level optimisers are combined with analysis software and perform local optimisation. A system level optimiser coordinates the low level domain optimisers in an attempt to bring all disciplines into agreement.

In general monolithic architectures are computationally superior over distributed approaches, in terms of their ability to tackle challenging problems in an acceptable number of analysis evaluations [9], [10]. Although they remain a preferred choice for solving specific engineering problems, to date their application in the early stages of industrial aircraft design has remained largely limited. Their internal strategy requires the merger of all relevant analysis tools. In principle this can be difficult to implement in an organisational structure where black box analysis methods are spread across multiple divisions of the company and require regular tuning from skilled operators.

Distributed architectures allow designers to simultaneously explore and optimise individual domains that are embedded in a tightly coupled system and in isolation from other domains. While this autonomy brings about numerous benefits, it introduces a number problems which are covered in more detail Section II. Out of the need to address many of these commonly shared drawbacks, the proposed rule based architecture was born. To make a meaningful assessment of its performance, it is applied to a number of problems specifically formulated for MDO comparison and evaluation.

II. Selection of Classical Distributed MDO Architectures

One way to satisfy the system consistency is to introduce *target* variables between the system and domain levels. The architectures examined in this section use these variables in their internal formulation. An upper domain controls the use of target variables and communicates them to lower domains that towards achieving these targets in their internal optimisation objective. If these targets are unattainable or consistency problems occur, they are revised and the process is repeated. As a result, all domains work towards a common vector of targets thus ensuring system feasibility at the end of the optimisation process. The main advantages of this family of architectures are their underlying simplicity and similarity to industrial design approaches. More specifically, when designs are driven by contractual requirements rather than optimum performance metrics, the existence of target variables matches the presence of economics driven characteristics.

It is common practice to present the architectures both graphically and mathematically. Here we have combined the mathematical formulation in a graphical format to represent the extent of the coupling between the system and domain level optimisation processes.

A. Collaborative Optimisation

The background, formulation and most notable recent refinements of the Collaborative Optimisation (CO) architecture are reviewed next. CO was conceived in 1994, out of the need to decompose the multidisciplinary problem in a way that would reduce disturbances in the natural divisions of aerospace companies and their proffered method of conducting analysis [11].

The ability to give domains design authority, the reduction in inter-domain communications and the flexibility to allow domains to select their own individual analysis tools were all factors behind the development of this architecture. At the time these requirements were driven by the development of new multi-fidelity computational analysis tools and communication difficulties facing geographically partitioned engineering teams. The CO formulation, shown in Fig. 1, completes the optimisation both at system and domain levels, but only completes the analyses at domain level. By channelling analysis and design information through the system level optimiser, the formulation eliminates direct communications between domains. In short, system-level optimiser aims to minimise the global objective, while the domain level optimisers aim to minimise the disagreement between various disciplines. Since the analyses are solely computed at the domain level and are of equal importance, it is not necessary to extend the formulation beyond the bi-level structure. As a result the CO architecture is particularly suitable for problems that do not have a natural hierarchical ordering, but rather have a collection of equally important domains [12].

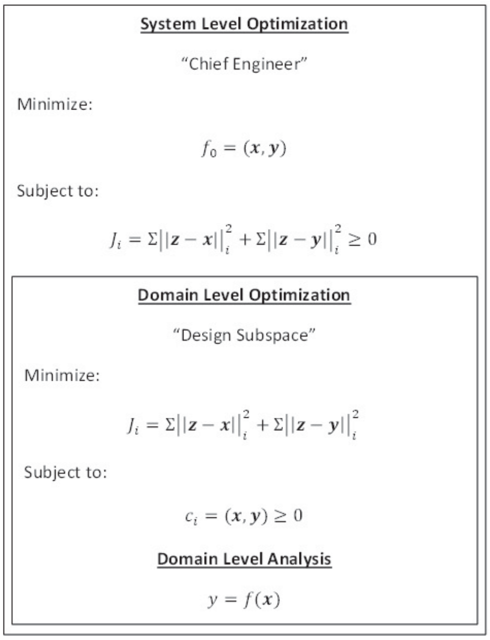

System Level Optimization
"Chief Engineer"
Minimize:
$f_0 = (\boldsymbol{x}, \boldsymbol{y})$
Subject to:
$J_i = \Sigma\left|\left|\boldsymbol{z} - \boldsymbol{x}\right|\right|_i^2 + \Sigma\left|\left|\boldsymbol{z} - \boldsymbol{y}\right|\right|_i^2 \geq 0$
Domain Level Optimization
"Design Subspace"
Minimize:
$J_i = \Sigma\left|\left|\boldsymbol{z} - \boldsymbol{x}\right|\right|_i^2 + \Sigma\left|\left|\boldsymbol{z} - \boldsymbol{y}\right|\right|_i^2$
Subject to:
$c_i = (\boldsymbol{x}, \boldsymbol{y}) \geq 0$
Domain Level Analysis
$y = f(\boldsymbol{x})$

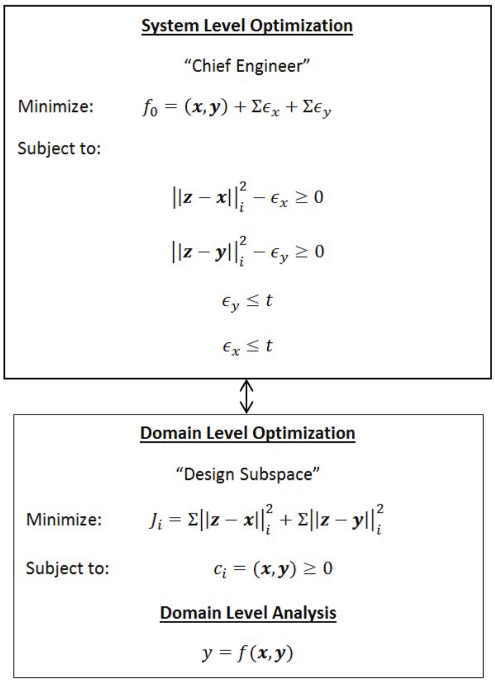
Fig. 1 Collaborative Optimisation

In spite of the organisational advantages, several major shortcomings are observed in the mathematical formulation of this architecture. A number of researchers showed that CO suffers from slow convergence [10], [13], [14], as well as poor robustness [10], [11] when applied to mathematical problems with a high degree of disciplinary cross coupling. Nevertheless this architecture still remains popular amongst academics, often used as a benchmark to the test newly developed architectures.

B. Analytical Target Cascading

In 1999, Michelena developed the architecture termed Analytical Target Cascading (ATC) [15]. It was devised to enable system level performance targets to be cascaded through the organisational hierarchy of design teams in the automotive industry. ATC differs from CO in the assumption that industrial design uses a hierarchical organisational structure. Higher-level domains set performance targets for the multi-tier system of low level domains [12]. Unlike the nested optimisation approach used in CO (Fig. 1), which focuses on discipline integration, ATC focuses on discipline dissolution. In other words, consider a wing design problem with several disciplines. Designers might wish to use the CO architecture to integrate process analysis tools of equally important domains, such as aerodynamics, structures and costing to achieve an acceptable solution. Conversely they might use ATC to organise hierarchical analysis processes, such as stability analysis, controls sizing and control mechanism design, to minimise a series of objectives.

Mathematically, the sub-domain formulation remains unchanged from CO. The major change in the architecture is presented in the system level optimiser constraints which are made up of auxiliary penalty functions and slack variables. In other words the system level constraints control how much disagreement is allowed between domain variables. This is manifested as separation between the system and domain level optimisers as depicted in Fig. 2.



System Level Optimization

"Chief Engineer"

Minimize: $f_0 = (\boldsymbol{x},\boldsymbol{y}) + \Sigma\epsilon_x + \Sigma\epsilon_y$

Subject to:

$$||\boldsymbol{z}-\boldsymbol{x}||_i^2 - \epsilon_x \geq 0$$

$$||\boldsymbol{z}-\boldsymbol{y}||_i^2 - \epsilon_y \geq 0$$

$$\epsilon_y \leq t$$

$$\epsilon_x \leq t$$

Domain Level Optimization

"Design Subspace"

Minimize: $J_i = \Sigma||\boldsymbol{z}-\boldsymbol{x}||_i^2 + \Sigma||\boldsymbol{z}-\boldsymbol{y}||_i^2$

Subject to: $c_i = (\boldsymbol{x},\boldsymbol{y}) \geq 0$

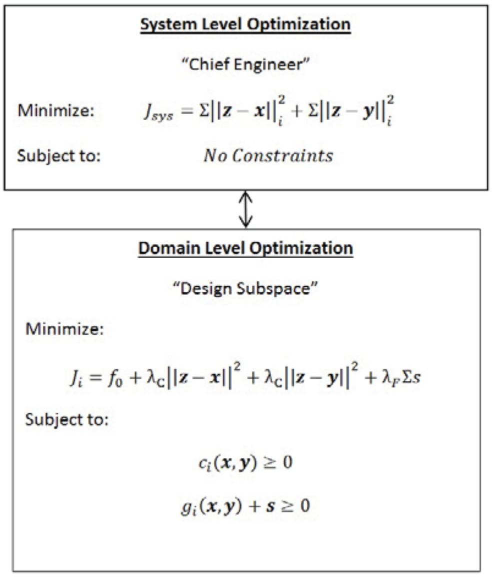
Domain Level Analysis

$$y = f(\boldsymbol{x},\boldsymbol{y})$$

Fig. 2 Analytical Target Cascading

C. Enhanced Collaborative Optimisation

The original formulations of CO and ATC restrict inter-domain communications and channel decisions about target variables solely through higher levels. In 2008, Roth developed a non-hierarchical MDO architecture called Enhanced Collaborative Optimisation (ECO) with the motivation to eliminate the majority of the numerical difficulties associated with CO and increase the influence of domains to better reflect the processes followed in industry [16]. At the core of the architecture is the idea that domains should control the objective function, rather than chasing targets imposed by a system level optimiser. The system level optimiser's goal is to minimise the inconsistencies between the domains, while individual domains minimise a relevant portion of the global objective function. The inter-domain communications occur in the form of constraints preferences, which are communicated across the different departments. Mathematically, the system level optimiser is unconstrained and solely aims to minimise the disagreements between sub-domains. The formulation of the domain objective function is substantially more complicated in comparison to CO and ATC. It consists of a quadratic model of the global objective, a compatibility penalty function to reduce differences between shared and state variables and a set of slack variables to ensure feasibility. Furthermore each domain includes of a set of additional linear constraints functions from other domains as well as domain specific constraints. A formal proof of convergence was demonstrated exists for ECO, unlike the original CO and ATC formulations [16].



System Level Optimization

"Chief Engineer"

Minimize: $J_{sys} = \Sigma||\boldsymbol{z}-\boldsymbol{x}||_i^2 + \Sigma||\boldsymbol{z}-\boldsymbol{y}||_i^2$

Subject to: $No\ Constraints$

Domain Level Optimization

"Design Subspace"

Minimize:

$$J_i = f_0 + \lambda_C||\boldsymbol{z}-\boldsymbol{x}||^2 + \lambda_C||\boldsymbol{z}-\boldsymbol{y}||^2 + \lambda_F\Sigma s$$

Subject to:

$$c_i(\boldsymbol{x},\boldsymbol{y}) \geq 0$$

$$g_i(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{s} \geq 0$$

Fig. 3 Enhanced Collaborative Optimisation

ECO has been shown to outperform previously described versions of CO and ATC in terms of analysis evaluations [17] and it offers considerable computational and organisational advantages over CO. However the most notable drawback is the level of complexity associated with the formulation of the objectives and constraints within each domain. In its most basic form the architecture is shown in Fig. 3.

III. Rule Based Architectures

The ultimate benchmark of a research field's impact is indicated by the utilisation of its theories in industry. Monolithic MDO architectures are used in detail design stage and their applications in industrial design processes is continually expanding [18].

A similar observation cannot be made for classical distributed architectures, which indicates that further work is required to facilitate their application outside of academic test cases. This claim is further supported by evidence that shows MDO practices produce superior results than those of the current sequential optimisation employed by industry. This scepticism can be justified as many of the current distributed methods collectively suffer from poor convergence speeds, poor reliability, complex formulations and require considerable organisational restructuring to enable their integration into current design procedures. In light of these drawbacks, a rule based approach has been proposed to tackle the MDO problems at the preliminary aircraft design stage, with the aim to enable designers to interact with and understand the distributed MDO process. Although, similar blackboard based approaches have received considerable attention in the past [19] and are somewhat in current development [20], their widespread application to MDO problems has remained largely unexplored. The aim here is to combine well established ideas from optimisation, artificial intelligence, data mining and computational steering to develop a competitive distributed architecture that is capable of finding a minimum solution with sufficient user computer interaction to enable its use in industrial aircraft design. It is not the goal of the architecture to find the exact global minimum to machine precision, but rather an acceptable solution that can be further developed in the preliminary aircraft design stage.

A. The Rules

In a recent assessment of MDO methods, Agte et al. [18] noted the applicability of "video game" style rules to solve real world MDO engineering problems. This methodology is the basis of a legacy architecture developed by Price et al. [10]. It consists of a system level rule base, which is tasked with the coordination of the multiple domains to a single feasible optimal result. Multidisciplinary agreement is achieved by moving the bounds on the shared design variables until their scope is deemed to encompass a single solution. The bound movement and reduction directions is based on the predetermined rule set, which is triggered by the outputs from the domains. In this context the output could be the end of an optimisation run, a single analysis evaluation or even a good guess. After each iteration, the rule based strategy triggers a specific bound action based on the preferences from all the domains. Thus, individual domains' optimisation procedures and goals remain largely unchanged from their current sequential approaches, with each domain maintaining a local objective and a number of local optimisation constraints [10].

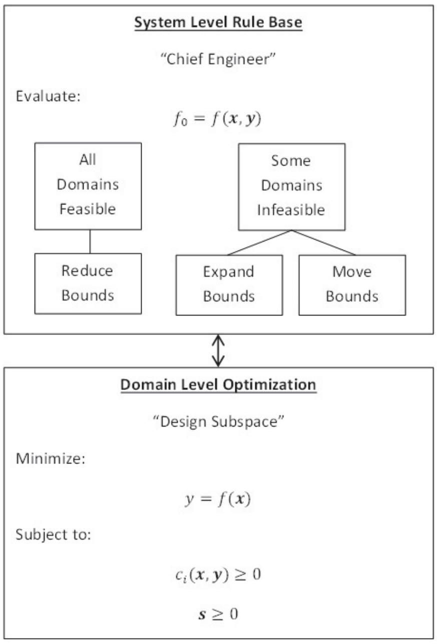


Fig. 4 Legacy Rule Base from Price et al. [10]

Fig. 4 summarises the two main outcomes addressed by the legacy rule set. In the event that all domains post a feasible result, the shared bounds are reduced. Conversely, if one or more domains is unable find a feasible solutions, the bounds are moved or expanded. While these rules were shown to cope with a number of test problems, they suffered from slow convergence in comparison to other distributed methods [10]. Upon further examination by the authors of this paper, the rules failed for problems (such as the one described in Section IV-A) where feasible solutions could be attained by all domains, but further bound reduction would not result in a feasible result. A simple thought experiment, inspired by the logic used in the Hooke and Jeeves [21] algorithm, led to a new set of rules that addressed these problems. These are shown in Table I and Fig. 5.

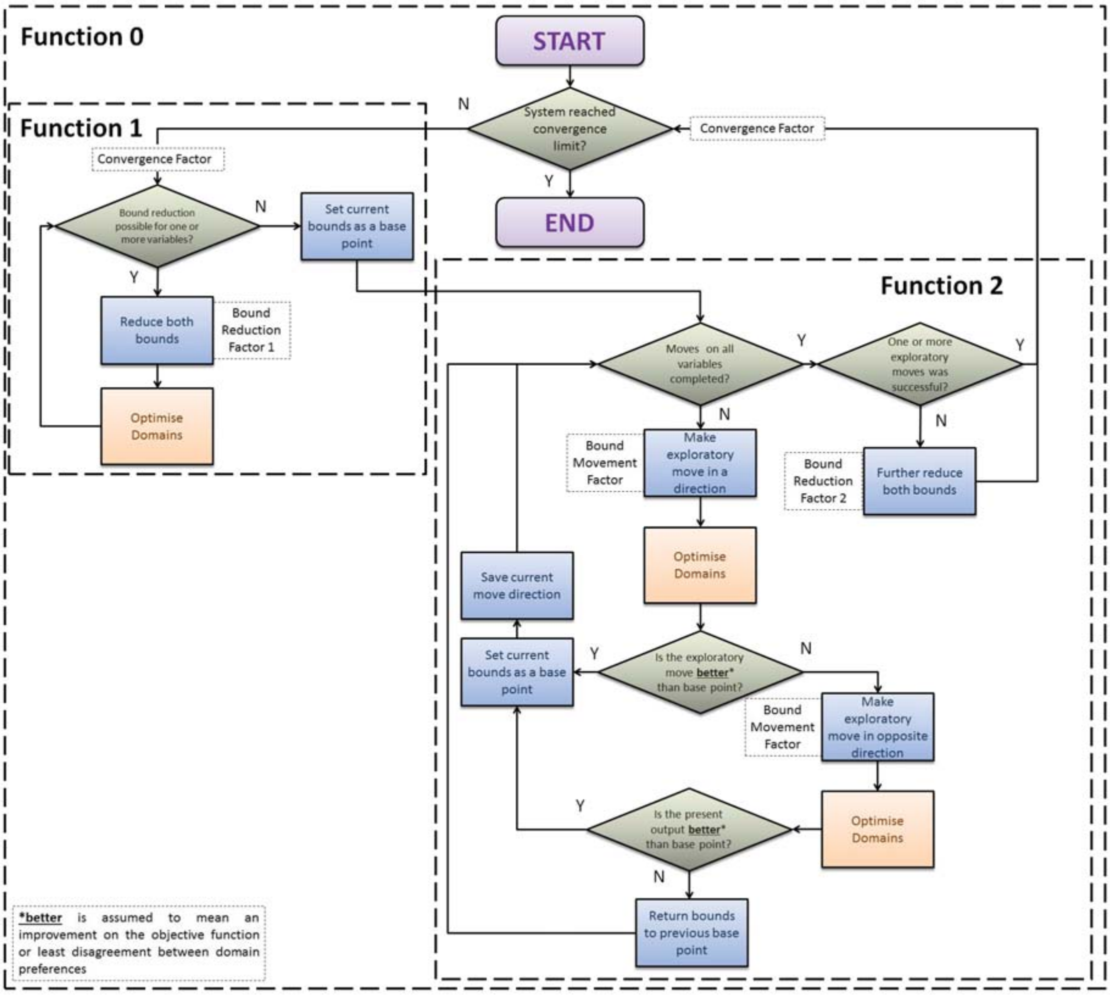

Function 0
START
System reached convergence limit?
N
Y
Convergence Factor
END
Function 1
Convergence Factor
Bound reduction possible for one or more variables?
N
Set current bounds as a base point
Y
Reduce both bounds
Bound Reduction Factor 1
Optimise Domains
Function 2
Moves on all variables completed?
Y
One or more exploratory moves was successful?
Y
N
N
Bound Movement Factor
Make exploratory move in a direction
Bound Reduction Factor 2
Further reduce both bounds
Optimise Domains
Save current move direction
Set current bounds as a base point
Y
Is the exploratory move better* than base point?
N
Bound Movement Factor
Make exploratory move in opposite direction
Optimise Domains
Y
Is the present output better* than base point?
N
Return bounds to previous base point
*better is assumed to mean an improvement on the objective function or least disagreement between domain preferences

Fig. 5 Decision Diagram for the Updated Rule Base

Function 0:
while $(\mathbf{ub} - \mathbf{lb}) < \epsilon_c \times (\mathbf{ub_{init}} - \mathbf{lb_{init}})$

Function 1:
while $(\mathbf{Z_{min}} - \mathbf{lb}) > (\epsilon_c \times \mathbf{Z_{min}})$ and $(\mathbf{ub} - \mathbf{Z_{max}}) > (\epsilon_c \times \mathbf{Z_{max}})$

$$\mathbf{ub} = \mathbf{ub} - \epsilon_{rb1} \times (\mathbf{ub} - \mathbf{Z_{max}}) \quad (1)$$

$$\mathbf{lb} = \mathbf{lb} + \epsilon_{rb1} \times (\mathbf{Z_{min}} - \mathbf{lb}) \quad (2)$$

Optimise Domain Functions, $y_{1,2...k}$
end

Function 2:
for length of vector $\mathbf{ub}$
Perform bound movement:

$$ub_{imu} = ub_i \pm \epsilon_{mb} \times (ub_i - lb_i) \quad (3)$$

$$lb_{imu} = lb_i \pm \epsilon_{mb} \times (ub_i - lb_i) \quad (4)$$

Optimise Domain Functions, $y_{1,2...k}$
Evaluate Objective Function, $f_{01,2...k}$
if outcome is better: keep direction.
else

$$ub_{imd} = ub_i \mp \epsilon_{mb} \times (ub_i - lb_i) \quad (5)$$

$$lb_{imd} = lb_i \pm \epsilon_{mb} \times (ub_i - lb_i) \quad (6)$$

Optimise Domain Functions, $y_{1,2...k}$
Evaluate Objective Function, $f_{01,2...k}$
if outcome is better: keep direction.
end
if bound movements on all variables were unsuccessful:

$$\mathbf{ub} = \mathbf{ub} - \epsilon_{rb2} \times (\mathbf{ub} - \mathbf{Z_{max}}) \quad (7)$$

$$\mathbf{lb} = \mathbf{lb} + \epsilon_{rb2} \times (\mathbf{Z_{min}} - \mathbf{lb}) \quad (8)$$

The rules and mathematical control of the bounds has undergone significant changes from the legacy method developed by Price et al. [10]. The logic can be programmed in three functions that deal with the main logic statements. Function 0 checks if the system has reached convergence. It is defined as the point at which the ratio between the separation

TABLE I
SUMMARY OF THE RULES AND ACTIONS

Rules	Actions
R1 - If the confined hyper-volume enclosed by the shared design variables has been reduced to a value that constitutes a single point.	**A1** - Terminate algorithm.
R2 - If the preferred shared design variables for all domains lie within both bounds.	**A2** - Reduce both lower and upper bounds towards closest domain results.
R3 - If the preferred design variables for all domains lie on both bounds.	**A3** - Move the bounds on one of the shared design variables in a direction.
R3a - If bound move produced a **better* result.	**A3a** - Keep current bound scope and save the movement direction for the next iteration. Move to rule **R1**.
R3b - If bound move did not produced a **better* result.	**A3b** - Move the bounds on the shared design variables in the other direction and recheck rule **R3a**.
R3c - If neither bound move produced a **better* result.	**A3c** - Return bounds to the base point and proceed with rule **R3** for the next shared design variable.
R3d - If bound movements on all shared design variables failed to produce a **better* result.	**A3d** - Force a bound reduction on all shared design variables on both the upper and lower bounds. Move to rule **R1**

**better* indicates whether the generated outcomes resulted in smaller disagreement between the domains (tMDF) or in an improvement in the objective function (tMDO).

of the current bounds ($\mathbf{ub}$ and $\mathbf{lb}$) and initial bounds ($\mathbf{ub_{init}}$ and $\mathbf{lb_{init}}$) reaches the value of the convergence factor ϵ_c. Function 1 deals with the secondary logic rule set, which aims to reduce the scope of the bounds until they become active against the upper and lower constraint boundaries. At the start, each domain shares the preferred design vector with the algorithm. The minimum and maximum preferences for each variable are extracted in the form of the vectors $\mathbf{Z_{min}}$ and $\mathbf{Z_{max}}$. These are used with the bound reduction factor $\epsilon_r b1$ to reduce scope of the bounds on the shared design variables, as given in (1) and (2). Function 2 is subsequently triggered to perform sequential exploratory moves along all variables. Here the bounds for each variable, ub_i and lb_i are moved upwards and/or downwards using the factor ϵ_{mb}. The successful direction of the initial moves is saved and repeated in the following iterations. This is done to reduce the number of objective function evaluations, with the assumptions that bound movements follow a pattern direction. In the event that neither upwards or downwards bound moves on a variable result in an improvement, the process is moved onto the next variable. If all exploratory moves do not result in an improvement, a bound reduction using the factor ϵ_{rb2} is forced on all variables as given by (7) and (8).

We have aimed to reduce the number of user defined factors to four (ϵ_c, ϵ_{rb1}, ϵ_{rb2} and ϵ_{mb}). Much like the factors used by the original Hooke and Jeeves method, these are set by the user and will determine speed, robustness, accuracy and precision of convergence for different problems.

IV. ARCHITECTURE COMPARISON

Two problems are used to assess the performance of the the rules based architectures. The first (an analytical problem) is used to compare the performance against the previously described distributed architectures, while the second demonstrates ability of the proposed method to cope with more sophisticated aircraft design problems. In addition, the problems were also investigated with the monolithic architecture Simultaneous Analysis and Design (SAND) [22] and the subsequent output was used as a benchmark against which all other architectures can be compared.

A. Analytical Test Problem

This popular analytical problem was used by a number of academics to compare the performance of numerous monolithic and distributed architectures [9], [17], [23]. It is suitable as it tries to mimic the behaviour of two conflicting domains by the functions given in (15) and (16), and overall design goals by the objective given by (9). It has a two level composition, thus it can be applied to the hierarchical ATC architecture as well as the bi-level CO and ECO architectures. In addition, it has a known global minimum located at $\mathbf{x} = \{1.9776, 0.0000, 0.0000\}$, $\mathbf{y} = \{3.7553, 3.1834\}$ and $f = 3.1840$. In the following test we have used a deviation tolerance between domain outputs of 0.0001 for convergence of the target-based architectures and for the rule base. The optimiser *fmincon* (SQP) from MATLAB$^{\text{TM}}$ with default settings was used as the domain optimiser and the system level optimiser for CO. The problem was initially started at $\mathbf{x} = \{1, 1, 1\}$ and $\mathbf{y} = \{1, 1\}$, with each subsequent iteration started from the previous solution.

Minimise:

$$f_0 = x_2^2 + x_3 + y_1 + e^{-y_2} \tag{9}$$

Such that:

$$3.16 - y_1 \leq 0 \tag{10}$$

$$y_2 - 24 \leq 0 \tag{11}$$

And:

$$-10 \leq x_1 \leq 10 \tag{12}$$

$$0 \leq x_2 \leq 10 \tag{13}$$

$$0 \leq x_3 \leq 10 \tag{14}$$

Where:

$$y_1 = x_1^2 + x_2 + x_3 - 0.2y_2 \tag{15}$$

$$y_2 = \sqrt{y_1} + x_1 + x_3. \tag{16}$$

Table II shows a summary of the results. The compatibility and feasibility variables in the ECO formulation were set to 0.1 and 15 respectively and the internal factors in the RBA formulation were set to: $\epsilon_c = 0.0001$, $\epsilon_{rb} = 0.65$, $\epsilon_{rb2} = 0.30$ and $\epsilon_{mb} = 0.99$.

B. Application of RB to Conceptual Aircraft Design Problem

The MDO problem considered below is derived from an empirical aircraft design tool [24], used here to evaluate wing performance based on geometry inputs. The problem is decomposed into two separate domains: structures and aerodynamics, with each analysis domain linked to the *fmincon* (SQP) optimiser from Matlab$^{\text{TM}}$. The domains have separate objectives, to reduce weight and drag, while satisfying local constraints for static margin in the aerodynamics domain and wing volume and undercarriage bay length constraints in the structures domain. For this test, the tool is used a black box evaluator of the two domains with no external gradient information. The design problem is constructed with the aim of optimising the wing of a single isle jet aircraft operating at Mach 0.785 with a maximum take off mass of 98,000 kg, a Reynolds number of 7.0 million and a wing area of 130m^2. We derive an objective function comprising of a weighted sum of the wing weight and drag to deliberate in the face of conflicting geometric preferences from the domains. Equation (17) is based on the direct operating cost function used by Price et al. [10], with W_{wg} as the wing weight and D/q as the drag force given in terms of the dynamic pressure.

$$f_0 = 4.7 \times D/q + 1.05 \times 10^{-3} W_{wg} \tag{17}$$

The rule base was configured to optimise parameters sequentially, with certain variables fixed before more detail is added to the design. While no significant advantage in the number of computational evaluations or objective outcome was observed, this sequential approach was primarily selected for two reasons. In the early stages engineers often fix variables that impact vehicle performance the greatest before adding more detail. This guarantees to some extent that the vehicle will meet contractual specifications. Furthermore, the effect that a variable change has on the system becomes increasingly difficult to visualise with a high number of optimisation parameters As a result, the optimisation routine here is limited to three parameters at the time. The order in which variables are optimised was selected based on the results of a Morris and Mitchell factorial screening study [25]. The study outputs a sample mean and standard deviation, which are indicative of the importance that parameter has on the objective function and the non-linearity of that variable respectfully. For this problem, the rule base first determines an optimal aspect ratio and leading edge sweep, while the remaining variables were fixed as parameters. These were are deemed to have the most significant effect on the objective function, given by the results in Fig. 6. Less dominant variables were optimised in the later sequences. This change in the original procedure converts the direct optimisation problem of 10 variables, to a multilevel sequentially optimised problem.

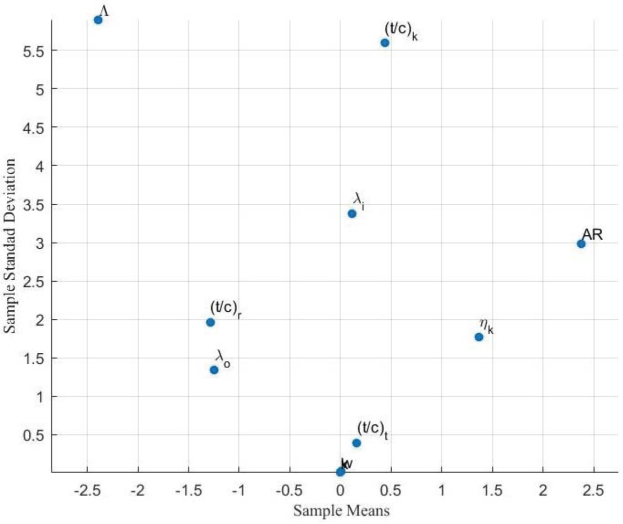


Fig. 6 Estimated means and standard deviation of elementary effect distributions

The ten optimisation variables and the results of the optimisation routines in the current test of the rule base architecture are summarised in Table III.

V. Visualisation

It is necessary for designers to be able to interact with and visualise the process followed by a given architecture. Arguably many of the distributed architectures have failed to entice design engineers because little or no visual feedback is output from the architecture. Here we have developed two visual interfaces that enable designers to monitor the architecture through the sequences of optimisation variables.

Fig. 7 shows the geometric 3D wing preference of each domain. How a sequence of variables affects the objective function is given by the scatter-plot shown for each domain. Each scatter-plot is build from the trace histories of the domains and is further updated as more points become available during successive iterations. The red box shows current scope of the bounds and the red point indicating the current preferred solution by each domain. Fig. 7 shows a sequence of 2 variables, but higher order sequences could be visualised with alternative plotting techniques. The scheme allows engineers to visualise trends across domains in real time and to directly see conflicting variables.

TABLE II
COMPARISON OF ARCHITECTURE PERFORMANCE ON AN ANALYTICAL PROBLEM

	Final Design Variables			Final State Variables		Obj. Fun.	Number of Fun. Calls			
Architecture	x_1	x_2	x_3	y_1	y_2	f_0	Dom. 1	Dom. 2	System Opt.	System Iter.
ATC	1.9963	0.0000	0.0000	3.1500	3.6537	3.1813	3095	2969	14529	30
CO	1.9777	0.0000	0.0000	3.1600	3.7553	3.1840	3295	2704	233	-
ECO	1.9823	0.0001	0.0000	3.1600	3.8489	3.1813	**577***	**577***	-	6
RBA$_{tMDF}$	1.9776	0.0000	0.0000	3.1600	3.7552	3.1840	1880	1584	-	61
SAND	1.9776	0.0000	0.0000	3.1600	3.7553	3.1840	30	30	5	-

*Results in bold signify a prematurely terminated process due to failure to reach the desired convergence.

TABLE III
AIRCRAFT DESIGN PROBLEM VARIABLES AND RESULTS

	Variable	L. Limit	Symbol	U. Limit	Units	Sequence	Starting	RBA$_{tMDO}$	SAND
Find	Aspect Ratio	6.0	AR	12.0	-	1	11.000	10.926	11.157
	Leading Edge Sweep	25	Λ	45.0	deg	1	26.000	26.306	26.595
	Spanwise Kink Position	0.2	η_k	0.45	-	2	0.2000	0.2400	0.2000
	Inner Taper Ratio	0.4	λ_i	0.7	-	2	0.7000	0.7000	0.6515
	Outer Taper Ratio	0.2	λ_o	0.6	-	2	0.2000	0.2008	0.2000
	Root Thickness to Chord Ratio	0.1	$(t/c)_r$	0.18	-	3	0.1000	0.1014	0.1000
	Kink Thickness to Chord Ratio	0.06	$(t/c)_k$	0.14	-	3	0.1100	0.1192	0.1131
	Tip Thickness to Chord Ratio	0.06	$(t/c)_t$	0.14	-	3	0.0600	0.0602	0.0600
	Wing Washout	2.0	w	5.0	deg	4	3.0000	4.9751	3.1243
	Fraction Tip Washout at Kink	0.65	k	0.85	-	4	0.6500	0.6503	0.6501
To Minimise	Drag		D/q		m^2	-	3.1142	3.1565	3.1619
	Wing Weight		W_{wg}		N	-	101201	98333	98431
Subject To	Pitch Up Margin		p	5.4	-	-	4.6417	4.6699	4.8584
	Estimated Wing Volume	23.0	V_{wg}		m^3	-	22.090	23.442	25.065
	Undercarriage Bay Length	2.1	V_{wg}		m	-	2.3774	2.3065	3.1620
Obj. Fun.							25.263	**25.161**	**25.196**
Number of Analysis Calls							Dom Aero.	1232	210
							Dom. Strc.	1478	210
System Level Iterations								37	-

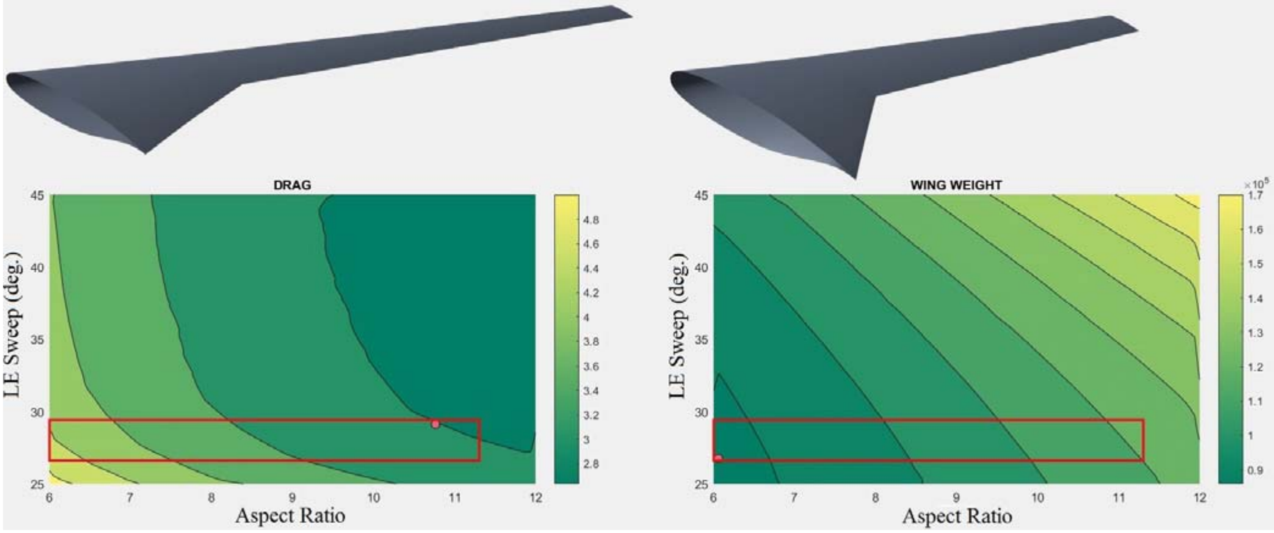


Fig. 7 Graphic Interface of Domain Preferences and Patterns

The second (Fig. 8) interface shows the exploration path followed by the rule base for all optimisation variables. Furthermore it displays the supposed optimum value with respect to the initial bounds.

VI. CONCLUSIONS AND FUTURE WORK

There are numerous real world MDO problems that require a solution strategy that can deal with multiple distributed analysis domains and cannot be realistically solved by any existing monolithic architecture. Hence they require a different, distributed approach. Many of the classical

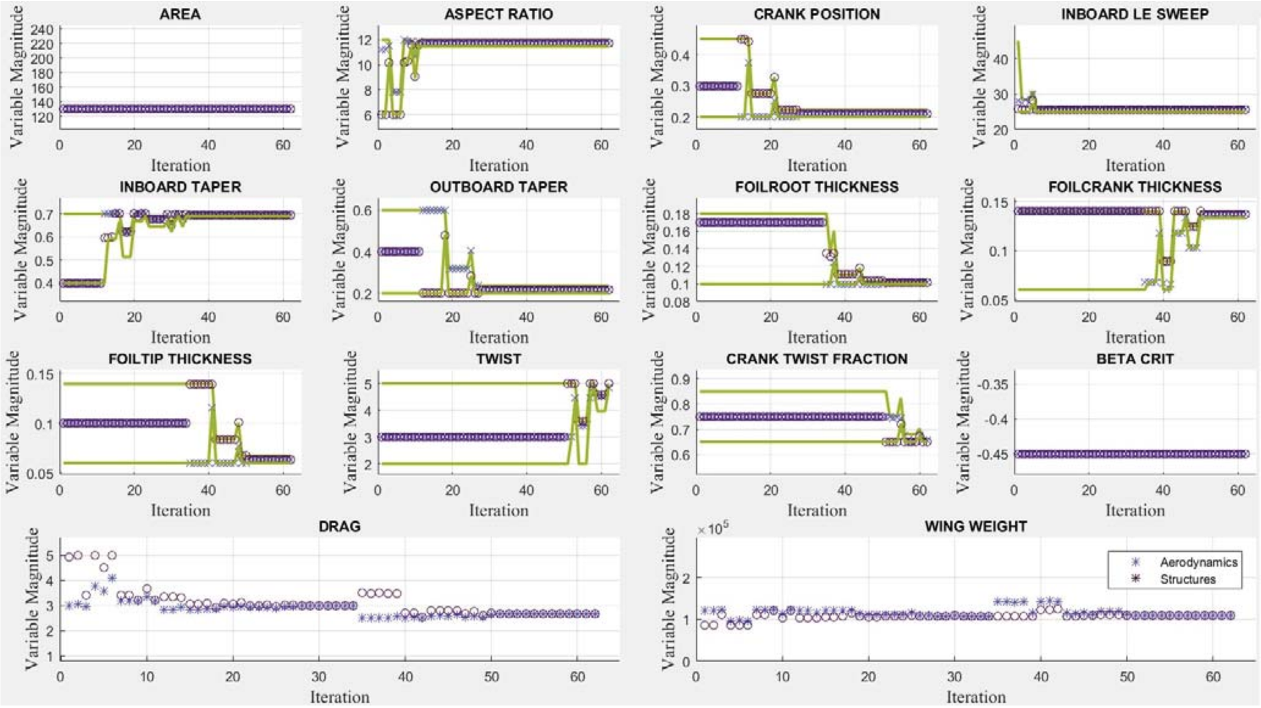


Fig. 8 Graphic Interface of Design Exploration Path Followed by the Architecture

distributed architectures suffer from flaws which make them unsuitable for use in an industrial environment. This has motivated the authors to develop and test an alternative architecture loosely based on the pattern search algorithm proposed by Hooke and Jeeves. The current work shows the merit of an rules based approach which has been demonstrated on two MDO problems. The proposed rule base performed competitively when compared against a number of target-based distributed architectures. It was also shown to tackle an MDO aircraft design problem consisting of 10 optimisation variables and 3 constraints. For this problem the architecture was extended to include two graphical interface to show domain preferences at each iteration.

A number of points have intentionally not been covered in the description of this method, primarily because it remains under development. The most important ones being: What is the best strategy for dealing with state variables in highly coupled problems? What are "good generic relaxation factors for most problems? What further rules would urther reduce convergence times or increase robustness? Inevitably the answers to these come as the architecture is applied to more analytical and design oriented problems. These will be remain focus of the further work in this area.

To expand the study of this architecture, the authors plan to apply the rule base to a real world Unmanned Aerial Vehicle wing design project, with the aim to test its applicability in a team based environment. The main outcomes should show if this method can reach a result in the time constraints of an engineering design project and if the outcomes are significantly better than the current sequential design processes.

Acknowledgement

This work is jointly funded by Airbus UK and the University of Southampton, whose support is gratefully acknowledged.

References

[1] K. Sabbagh, *Twenty First Century Jet: Making and Marketing the Boeing 777*. Scribner Book Company, 1996.

[2] G. Norris and M. Wagner, *Airbus A380: superjumbo of the 21st century*. Zenith Imprint, 2005.

[3] M. H. Sadraey, *Aircraft Design: A Systems Engineering Approach*. John Wiley & Sons, 2012.

[4] L. Prandtl, "Uber tragflugel des kleinsten induzierten widerstandes," *Zeitschrift für Flugtechnik und motorluftschiffahrt*, 1933.

[5] R. T. Jones, "The spanwise distribution of lift for minimum induced drag of wings having a given lift and a given bending moment," 1950.

[6] J. DeYoung, "Minimization theory of induced drag subject to constraint conditions," 1979.

[7] G. K. Kenway and J. R. Martins, "Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration," *Journal of Aircraft*, vol. 51, no. 1, pp. 144–160, 2014.

[8] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary design optimization: A survey of architectures," *AIAA journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[9] N. P. Tedford and J. R. Martins, "Benchmarking multidisciplinary design optimization algorithms," *Optimization and Engineering*, vol. 11, no. 1, pp. 159–183, 2010.

[10] K. A. J. Price, A. R. and C. M. Holden, "On the coordination of multidisciplinary design optimization using expert systems," *AIAA journal*, vol. 49, no. 8, pp. 1778–1794, 2011.

[11] A. S. B. R. G. P. Kroo, I. and I. Sobieski, "Multidisciplinary optimization methods for aircraft preliminary design," *AIAA paper*, vol. 4325, p. 1994, 1994.

[12] J. Allison, M. Kokkolaras, M. Zawislak, and P. Y. Papalambros, "On the use of analytical target cascading and collaborative optimization for complex system design," in *6th world congress on structural and multidisciplinary optimization, Rio de Janeiro, Brazil*, 2005, pp. 1–10.

[13] N. Alexzandrov and R. M. Lewis, "Analytical and computational aspects of collaborative optimization," *NASA/TM-210104-2000*, pp. 1205–1210, 2000.

[14] S. Kodiyalam, “Evaluation of methods for multidisciplinary design optimization (mdo). part 1,” 1998.

[15] N. Michelena, H. M. Kim, and P. Papalambros, “A system partitioning and optimization approach to target cascading,” in *Proceedings of the 12th International Conference on Engineering Design*, vol. 2, 1999, pp. 1109–1112.

[16] B. D. Roth, “Aircraft family design using enhanced collaborative optimization,” Ph.D. dissertation, 2008.

[17] I. Roth, B. & Kroo, “Enhanced collaborative optimization: Application to an analytic test problem and aircraft design, (2008),” in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, BC*, 2008.

[18] J. Agte, O. De Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck, “Mdo: assessment and direction for advancementan opinion of one international group,” *Structural and Multidisciplinary Optimization*, vol. 40, no. 1-6, pp. 17–33, 2010.

[19] H. P. Nii, “The blackboard model of problem solving and the evolution of blackboard architectures,” *AI magazine*, vol. 7, no. 2, p. 38, 1986.

[20] N. Li, R. Tan, Z. Huang, C. Tian, and G. Gong, “Agile decision support system for aircraft design,” *Journal of Aerospace Engineering*, vol. 29, no. 2, p. 04015044, 2015.

[21] R. Hooke and T. A. Jeeves, ““direct search” solution of numerical and statistical problems,” *Journal of the ACM (JACM)*, vol. 8, no. 2, pp. 212–229, 1961.

[22] R. T. Haftka, “Simultaneous analysis and design,” *AIAA journal*, vol. 23, no. 7, pp. 1099–1103, 1985.

[23] R. E. Perez, H. H. Liu, and K. Behdinan, “Evaluation of multidisciplinary optimization approaches for aircraft conceptual design,” in *AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, NY*, 2004.

[24] J. Cousin and M. Metcalf, “The bae (commercial aircraft) ltd transport aircraft synthesis and optimisation program (tasop),” in *Aircraft Design, Systems and Operations Conference*, p. 3295.

[25] M. D. Morris, “Factorial sampling plans for preliminary computational experiments,” *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.