AIAA Journal
Vol. 49, No. 8, August 2011

On the Coordination of Multidisciplinary Design Optimization Using Expert Systems

Andrew R. Price* and Andy J. Keane*
University of Southampton, Southampton, England SO17 1BJ, United Kingdom
and
Carren M. E. Holden
Airbus Operations Ltd., Bristol, England BS99 7AR, United Kingdom

DOI: 10.2514/1.J050928

Multidisciplinary design optimization of complex systems by large enterprises is typically broken down along domain specialist lines with associated expertise, tools, method, and process. This paper investigates an approach to multidisciplinary design optimization that exploits heuristic control of the bounds of the common design variables across the domains of a decomposed problem. By bringing the common design variable vector to a single point in the design space, in the context of internally consistent and multiple discipline feasible state variables (and consistently resolved local design vectors), the multidisciplinary design optimization problem is solved. A system level management approach is presented in which an expert system is used to coordinate the activities of the domain level optimizers. A rule base has been developed to manage the bounds on the common design variable vector, control the exchange and relaxation of state coupling variables, and control the specification of the domain level optimization problems. Through application of the rule base across a range of representative multidisciplinary design optimization problems assembled from the literature, the viability and performance of the method are discussed.

Nomenclature

$\mathbf{A}_j$	=	analysis function of the jth domain
$\mathbf{A}_{jk}$	=	analysis function of the jth domain responsible for generating the coupling state variables input to the kth domain
$g_i^{(j)}$	=	ith element of the inequality constraints vector of the jth domain
$h_i^{(j)}$	=	ith element of the equality constraints vector of the jth domain
N	=	number of design domains/disciplines
NX	=	number of local design variables in the problem
NX_j	=	number of local design variables in the jth domain
NY_j	=	number of state variables evaluated in the jth domain
NY_{jk}	=	number of state coupling variables evaluated in the jth domain intended as input to the kth domain
NZ	=	number of shared design variables
p	=	parameters of a domain optimization problem
$\mathbf{SZ}$	=	vector of complimentary variables for the shared design variables
$\mathbf{SX}_j$	=	complimentary variables for the local design variables of the jth domain
$\mathbf{SY}_j$	=	complimentary variables for the state variables of the jth domain
$\mathbf{SY}_{jk}$	=	complimentary variables for the state variables of the jth domain intended as input to the kth domain
$\mathbf{X}$	=	vector of all local design variables
$\mathbf{X}_j$	=	local design variables vector in the jth domain
$x_i^{(j)}$	=	ith element of the local design variables vector in the jth domain
$\mathbf{Y}_j$	=	state variables vector evaluated in the jth domain
$\mathbf{Y}_{jk}$	=	state variables vector evaluated in the jth domain intended as input to the kth domain
$y_i^{(jj)}$	=	ith element of the noncoupling state variables vector evaluated in the jth domain
$y_i^{(jk)}$	=	ith element of the coupling state variables vector evaluated in the jth domain and intended as input to the kth domain
$\mathbf{Z}$	=	vector of shared design variables
z_i	=	ith element of the shared design variables vector

I. Introduction

In the design of complex engineering systems involving multiple disciplines it is critical that the interactions between the subsystems of the problem are accounted for. Only by considering the fully coupled system can an optimal design emerge. Formal multidisciplinary design optimization (MDO) methods typically fall into two broad categories: 1) monolithic formulations where a single optimizer addresses the whole problem, and 2) multilevel methods where the problem is decomposed along disciplinary lines and optimization takes place at both a system and domain level. The single optimizer approach is conceptually simple to understand but can scale poorly for larger problems and increasing number of disciplines. It may also prove problematic in an industrial setting to bring all of the domain analysis tools under the control of a single optimizer. Multilevel architectures promote discipline autonomy. The system level is responsible for managing interactions between disciplines. Such an approach allows design teams to work with limited interruption based upon targets set at the system level. If MDO methods are to be accepted in an industrial context they must support this form of distributed design optimization for both organizational and computational reasons.

A review of MDO methods is provided by Sobieczszanski–Sobieski and Haftka [1]. A common feature across many of the hierarchical algorithms described is the manipulation of the shared design vector at the system level while maintaining lower level design autonomy. In this work a related approach is proposed; that of replacing the formal system level optimizer with an expert system to reason over information from the domains and make decisions about changes to the common design variables vector or bounds. Such an approach sacrifices, possibly elusive, guarantees of convergence for potentially attractive returns in the enterprise:

Received 5 October 2010; revision received 6 January 2011; accepted for publication 8 January 2011.

*School of Engineering Sciences.

1) It allows systems architects to interact with the design process (whereas a formal optimizer has to run its course and may make bad moves in the process).
2) The path through the design space can be understood by the systems architects.
3) Domains retain autonomy, optimizing the physical problem they specialize in, minimizing disruption to existing organizational/institutional arrangements.
4) The system can intervene if domains are stalled or converging to poor designs and move on to the next iteration.
5) It provides a means to engage people in the process, whereas formal optimizers coordinating domain activities are not necessarily adaptable if the design specification changes in some way as disciplines investigate their local designs.

This paper focuses on methods for mediating between design teams that are performing design optimization in separate domains with a shared subset of common design variables. The probably conflicting preferences for the critical design parameters between domains must be resolved for a viable whole system to emerge. The process of discussion and compromise in the real world is typically achieved through regular gated reviews in which the chief engineers responsible for each domain formally meet to resolve or revise values for the critical parameters of the system design. This work is concerned with exploring methods and tools for enabling design teams to be informed of each others' developing view of the design, to coordinate or influence the direction of optimization and design search within those domains and to improve the efficiency with which an optimal whole system is generated.

An outline of the proposed approach is detailed in Sec. II. The initial rule base that has been developed is discussed and the architecture of the investigative framework is described. To make a meaningful assessment of the performance of the rule based approach a number of standard MDO algorithms have been implemented which are outlined in Sec. III. A number of representative numerical MDO problems have also been collated from the literature, of which two are presented in Sec. IV. The results of running the rule base at the system level on these problems is presented in Sec. V and compared with the results achieved by the standard MDO algorithms. The performance of the expert system approach is discussed in Sec. VI with conclusions and future work presented in Sec. VII.

II. Proposed Approach

A. MDO Formulation

A multidisciplinary design optimization problem is characterized by the presence of multiple domains or disciplines that are each responsible for the detailed design of a subcomponent of the full system. Typically, a MDO problem consists of a set of common design variables $\mathbf{Z}$ that influence some or all of the domains, a set of design variables local to the ith domain $\mathbf{X}_i$ and state coupling variables $\mathbf{Y}_{ji}$ exchanged from domain j to domain i. Each domain analysis evaluates the domain state variables $\mathbf{Y}_{ik,i=1\ldots n}$ for the given input vector ($\mathbf{Z}$, $\mathbf{X}_i$, $\mathbf{Y}_{ji,j=1\ldots n}$). The variables $\mathbf{Y}_{ji}$ are input to the calculation of domain i while variables $\mathbf{Y}_{ik}$ are output from domain i. This exchange of state coupling variables represents a coupled system whose governing equations $\mathbf{Y} = A(\mathbf{Z}, \mathbf{X}, \mathbf{Y})$ must be solved to achieve a viable design (Fig. 1). An optimal system design is achieved when multidisciplinary feasibility is satisfied and the system objective function $f(\mathbf{Z}, \mathbf{X}, \mathbf{Y})$ is minimized subject to all domain and system level constraints. The MDO problem is expressed as:

$$\text{Find } \mathbf{Z}, \mathbf{X} \tag{1}$$

$$\text{to minimize } f(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X})) \tag{2}$$

$$\text{subject to } g(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X})) \leq 0 \tag{3}$$

$$h(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X})) = 0 \tag{4}$$

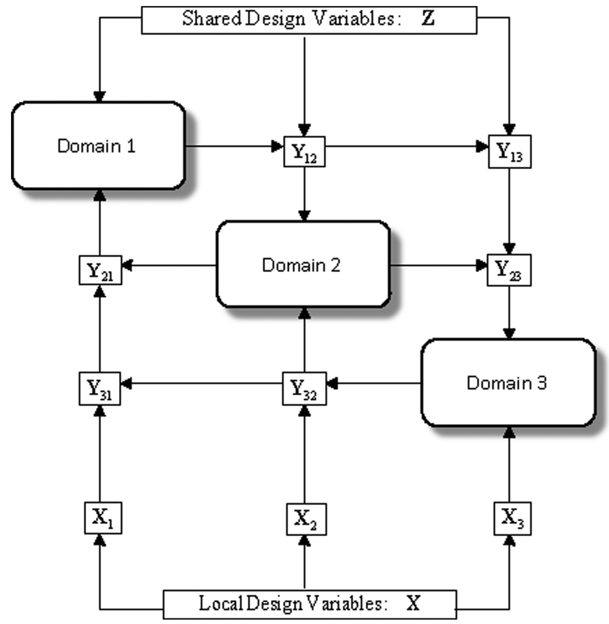

Fig. 1 Data flow required in multidisciplinary analysis.

B. System Architecture

An investigative framework (Fig. 2) has been developed exploiting an expert system as the coordinating process for multidisciplinary design optimization. This system level "master" process has access to a central repository of information (the blackboard [2]) which details both the present state of the design and the history of the MDO search. The expert system employs a rule base to make decisions about how the domain level design optimizations should proceed. The results of the reasoning of the expert system are written into the central database and the domains, acting asynchronously, perform the next local optimization as resource becomes available. The inference engine controls the design process by specifying the bounds and parameters provided as input to the domain optimizers working on their part of the decomposed problem. The design

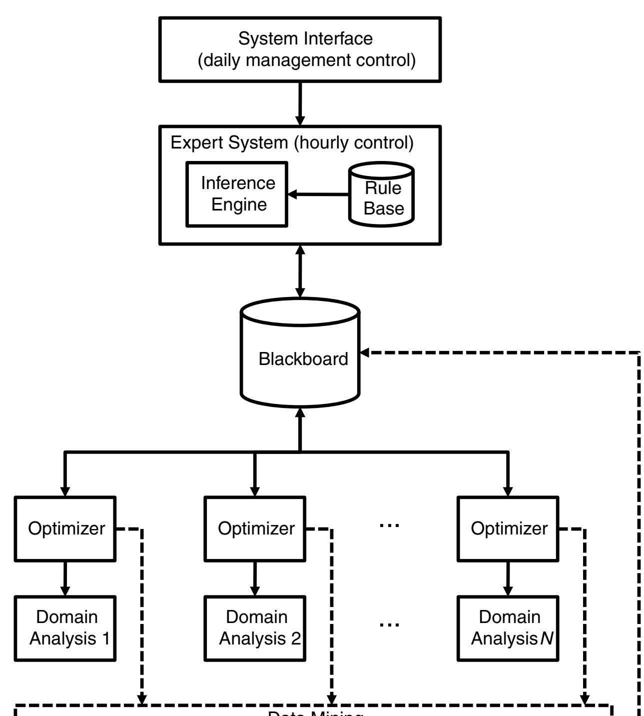

Fig. 2 System architecture deployed to investigate the rule base approach to MDO.

problem is solved by narrowing in on single values for the shared design variables through systematic reduction of their bounds.

The domains operate by first specifying the interface to their analysis model in the system. The blackboard records details of the local design variables and their bounds, the local inequality, equality constraints and parameters. The domains configure their optimization problem using the values specified in the blackboard from the latest expert system reasoning and start an optimization as instructed. In due course, the domains upload their optimizer trace histories as the optimization proceeds so that real time data analysis can be employed and the domain optimization performance assessed. The data analysis aims to provide high-level information about the domain optimization (such as its phase, active and inactive constraints, objective function value, etc.) so that decisions can be made about how to best guide the system design. Upon completion of its optimization each domain uploads its preferred design to the blackboard. The domain writes its optimal values for the local, shared and state variables, the objective function value and the values of the constraint functions to the blackboard. The expert system reasons across the results of all the domains, the summary information about behavior of the optimizers in the domains, and the system objective function to adjust system level variables and specifies new domain optimization/design problems. The process continues until convergence criteria are achieved as determined by the expert system.

C. Rule Base

The investigative MDO framework described here is orchestrated using Python with either a CLIPS[†] or JESS[‡] expert system employed to provide the coordinating master process. A rule base has been developed that provides a set of rules that address input and output (IO), feature recognition [3], and optimization logic (bounds adjustment). The IO rules are responsible for reading and writing data into and out of the expert system (ES). The input rules process the files that comprise the blackboard and assert raw facts in the ES workspace about the optimizer results across the multiple domains. Here the rule base is designed to process this raw data to reach a decision about whether the current bounds need to be updated. The feature recognition aspect of the rule base is a preprocessing stage that is performed outside of the expert system through data analysis of the blackboard data to infer bulk properties of the system. In this particular implementation the rules recognize two situations; all domains have produced a feasible result for the present bounds or some domains are infeasible in the present bounds. This is an "either–or" situation and the subsequent processing proceeds along two separate pathways; the bounds are either reduced in scope or the bounding box is moved back toward feasibility. If the bounds have been modified or the convergence criteria have been achieved then the output rules are responsible for writing this information to the blackboard. The rule base is invoked by the master process at regular intervals, gradually reducing and moving the bounds on the shared design variables until the bounds converge to some given tolerance on a "point" in the design space.

The rule base comprises a number of predicate-action constructs that are designed to manipulate the domain level optimization problems to reach a consensus on the optimal shared design variables vector $\mathbf{Z}$ with an accompanying consistent set of multidiscipline feasible state coupling variables $\mathbf{Y}$. The rules in isolation are notionally easy to understand and the expert system can explain the decisions that are being made at the system level. The rules discussed below comprise the significant logic of the rule base. There are additional rules implemented in the rule base to facilitate the data processing required to provide the appropriate predicate information the main rules fire upon but these are not explicitly outlined here.

The basic logic of the rule base is made up of three groups of rules: the first tests for convergence; the second deals with the situation where feasible solutions are being returned by domains and so there is freedom to converge toward a global optimum; while the final set deals with situations where one or more domains cannot find a feasible solution and then tries to either move or relax restrictions to restore feasibility. These rules are set out next in the order they are tested: once a rule has fired and the blackboard updated the domain searches are again invoked and the process repeats until either convergence is reached or the available computing budget is exhausted.

Ruleset-Z-test-convergence (C1): The optimization process is deemed to have converged once the hyper-volume enclosed by shared design variable bounds has been reduced to a volume that is considered to constitute a single point in the design space. The other rules in the rule base act to ensure that the bounds in each dimension cannot be reduced beyond a minimum separation so that the enclosed hyper-volume should be reasonably uniform across the dimensions. If the convergence criteria are met then an exit flag is set in the blackboard to indicate the process is complete.

Ruleset-Z-reduce (R1): A primary rule in the system is the reduction of the bounds on the shared design variables in the event that all domains have posted a feasible result and the side-constraints of each domain optimization are inactive. From the system perspective the domains have reported their preference in their local design space and stated that their analysis produces a feasible optimum that is not constrained by the current bounds on the shared design variable. This rule therefore reduces the scope of the bounds by moving the upper and lower limit toward the nearest respective domain optimum. In the event that all domains post a feasible result the state coupling variables are moved from their values on input to their optimal values on output. This provides an implicit relaxation scheme as the system optimization proceeds.

Reduce-free-bound (R2): In the event that all domains report a feasible domain optimum but where one of the bounds on a shared design variable is active, progress can be made by reducing the opposite bound. If these predicates are satisfied the free bound is moved toward the nearest domain optimum in that design dimension.

Reduce-least-sensitive-bound (R3): The final extension to the rules for dealing with feasible domain results is to make a decision on which bound to move when both side-constraints are active on a shared design variable. In this instance the sensitivity of the system objective function to perturbation in each side-constraint is evaluated in the domain. The bound for which compromise in its value leads to the least detriment to the system objective function is moved.

Ruleset-Z-move (I1): In the event that one or more domains report that a feasible solution could not be found, the bounds on the shared design variables are moved back toward the last values at which a feasible solution was found. The predicates on this rule will also fire rules that manage state variables so that state coupling variables are moved from their current input values back toward the input values that produced the last feasible result.

Ruleset-Z-expand (I2): In the event that one or more domains cannot find a feasible result and that the previous master reasoning reduced one of the bounds then that previous move is partially reversed.

Ruleset-Z-static-bound (I3): As the system reduces the bounds on the shared variables the side-constraints become increasingly active. The rules listed above are designed to reduce the scope of the bounds. In the event that the previous three reasoning sessions have acted to move one bound toward the opposite bound we identify the opposite bounds as static. If a static bound is detected then its position is refined by moving both bounds in the direction of travel of the moving bound.

The basic logic is thus:

```
if(C1)then
  stop
else
  if(R1)then
    reduce bounds
  elseif(R2)then
    reduce opposite bound
  elseif(R3)then
```

[†]Additional data on CLIPS: A Tool for Building Expert Systems available at http://clipsrules.sourceforge.net/, [accessed December 2008].

[‡]Additional data on JESS: The Rule Engine for the Java Platform available at http://www.jessrules.com/jess, [accessed October 2010].

```
        move least sensitive bound
      else
        if(I1)then
          move bounds back toward feasibility
        elseif(I2)then
          partially reverse previous move
        elseif(I3)then
          move static bound
        endif
      endif
    endif
    repeat until compute limit reached
```

System analysis is not performed during the design process: it is not possible to perform one as there is no consensus on the shared design variables vector until the process converges. As the design proceeds the domains may find different optimal values for the shared design variables. The potential conflict is resolved by reducing and moving the bounds on the shared design variables until the hyper-volume they enclose effectively defines a single design vector. By using rules to manage the exchange of state coupling variables between domains, the final design is multidiscipline feasible once the convergence criteria are achieved.

Each domain is autonomous and in the framework deployed here, periodically polls the database for the latest definition of its optimization problem. A domain will configure and execute the local design problem and upload the results of the optimization. The master process acts asynchronously, interrogating the blackboard at regular intervals, invoking the rule base, and uploading the next set of local optimization problem configurations. The master process is executed with a higher frequency than that with which domains are posting results. The master process can therefore provide a new domain problem at least as frequently as the domain with the shortest period.

The performance of the rule base is explored here using two types of optimizer in the domains. The first approach sets up the domain optimization problems to exploit the Matlab `fmincon` optimizer from its Optimization Toolbox. This provides a means to make meaningful comparison with some standard MDO algorithms from the literature, also implemented in Matlab. In the second approach, a "free-to-air" GA code, obtained from the Kanpur GA Lab, is executed using a 30 member population over 200 generations, resulting in a fixed computational budget of 6000 domain analysis calls per disciplinary optimization. For the investigative framework this is a deliberate restriction that limits the quality of the result obtained at the domain level. For more demanding MDO problems the quality of domain result is likely to decrease. The system level is therefore presented with a more realistic challenge of reasoning over incomplete or partial results/information and making good decisions in guiding the design process. In this regard, we anticipate that more sophisticated data analysis of domain optimizer behavior will be essential to the successful functioning of the rule base approach on difficult problems. An expert system will only be as good as the information over which it reasons.

To assess the performance of the rule based coordination a number of standard MDO algorithms from the literature have been implemented in Matlab using the SQP method `fmincon`. A number of MDO problems have also been assembled from the literature ranging from simple numerical constructs, through relatively simple preliminary aircraft design problems (comprising a page of equations) to a cutdown and decomposed version of a commercial aircraft wing design tool. The problems have been implemented in both the rule base framework and the Matlab MDO framework to enable some comparison of performance in both qualitative and quantitative terms.

III. MDO Methods and Algorithms

In this section we briefly outline a number of the standard MDO algorithms from the literature. The general formulation of MDO problems is described in greater detail in Cramer et al. [4], where details can be found on several of the following methods.

A. Multiple Discipline Feasible

The multiple discipline feasible (MDF) method [4] embodies the traditional approach to multidisciplinary design in which a single optimizer operates by solving the system analysis at each evaluation. The optimizer manipulates the shared and local design variables and performs a full system analysis to achieve multidisciplinary feasibility. The coupled system is solved at each optimization iterate and the optimizer therefore operates over a multidiscipline feasible (MF) system objective function subject to all domain and system constraints

$$\text{Find } \mathbf{Z}, \mathbf{X} \tag{5}$$

$$\text{to minimize } f(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X})) \tag{6}$$

$$\text{subject to } g(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X})) \leq 0 \tag{7}$$

$$\text{where } \mathbf{Y}_i = \mathbf{A}_i(\mathbf{Z}, \mathbf{X}, \mathbf{Y}_{\substack{j=1\ldots N,i \\ i\neq j}}) \tag{8}$$

$$\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_N^T]^T \tag{9}$$

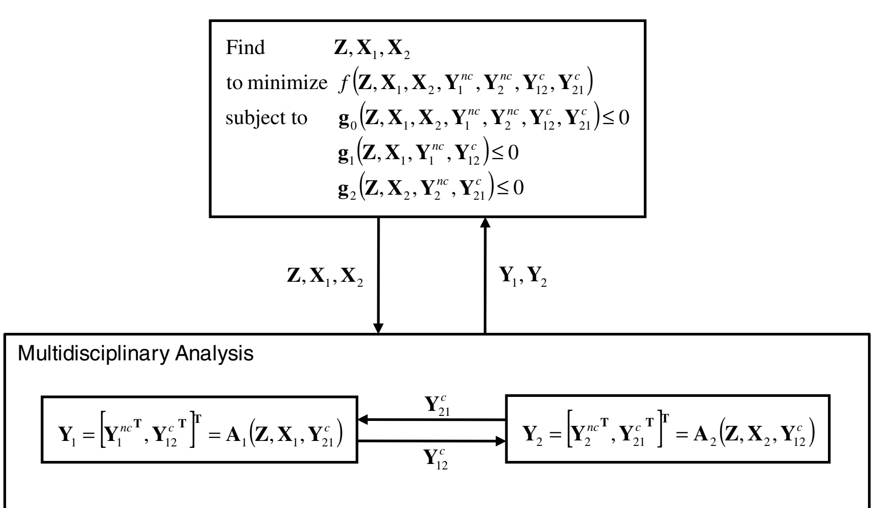

Fig. 3 MDF method.

The construction of the optimization problem for two domains is shown in Fig. 3.

B. Individual Discipline Feasible

The individual discipline feasible (IDF) method [4] introduces surrogate variables for the state coupling data of the problem and gives control of these variables to the optimizer. This eliminates the system analysis from the calculation but at the cost of introducing additional design variables and accompanying equality constraints, employed to ensure that the surrogate variables are consistent with the analysis output when the optimum is found. The method achieves decoupling of the domains (and the attendant burden of an iterative system analysis solver) and each discipline can be analyzed in isolation. The optimizer is responsible for converging to a multidisciplinary feasible state at the optimum by satisfying the added equality constraints:

$$\text{find} \;\; \mathbf{Z}, \mathbf{X}, \mathbf{SY}_{\substack{ji \\ i \neq j}} \tag{10}$$

$$\text{to minimize} \;\; f(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X})) \tag{11}$$

$$\text{subject to} \;\; g(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X}, \mathbf{SY}_{ji})) \leq 0 \tag{12}$$

$$h = \mathbf{Y}_{\substack{ji \\ i \neq j}} - \mathbf{SY}_{\substack{ji \\ i \neq j}} = 0 \tag{13}$$

$$\text{where} \;\; \mathbf{Y}_i = \mathbf{A}_i(\mathbf{Z}, \mathbf{X}, \mathbf{SY}_{\substack{j=1..N,i \\ i \neq j}}) \tag{14}$$

$$\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_N^T]^T \tag{15}$$

The construction of the optimization problem for two domains is shown in Fig. 4.

C. All-At-Once

Similar to IDF, the all-at-once (AAO) method [4] promotes *all* of the state variables of the problem to optimization variables. The method presents the governing equations of each discipline as equality constraints to the optimizer which then seeks to reduce the residuals of the state variables with the surrogate state variables:

$$\text{find} \;\; \mathbf{Z}, \mathbf{X}, \mathbf{SY}_{ji} \tag{16}$$

$$\text{to minimize} \;\; f(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X})) \tag{17}$$

$$\text{subject to} \;\; g(\mathbf{Z}, \mathbf{X}, \mathbf{Y}(\mathbf{Z}, \mathbf{X}, \mathbf{SY}_{ji})) \leq 0 \tag{18}$$

$$h = \mathbf{Y}_{ji} - \mathbf{SY}_{ji} = 0 \tag{19}$$

$$\text{where} \;\; \mathbf{Y}_i = \mathbf{A}_i(\mathbf{Z}, \mathbf{X}, \mathbf{SY}_{\substack{j=1..N,i \\ i \neq j}}) \tag{20}$$

$$\mathbf{Y} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_N^T]^T \tag{21}$$

Again, the system analysis is eliminated from the formulation at the expense of additional design variables and equality constraints. The construction of the optimization problem for two domains is shown in Fig. 5.

D. Collaborative Optimization

Collaborative optimization (CO) [5] is a bilevel method in which complimentary variables are introduced and optimized in the upper level and then passed to the domains as targets for the lower level optimizers. The domains are asked to investigate the local analysis by attempting to match the target design variable values subject to the local constraints. The domains provide the sensitivities of local objective functions formed from the discrepancies between the coupling variables and their surrogates so that the upper level can update the complimentary design variables in such a way that the system objective function can be reduced while maintaining multidisciplinary feasibility. The method provides discipline autonomy and does not require the system analysis to be performed.

There are a number of variations of the method. In this work the upper level optimization problem performs a search for the values of the complimentary design variables $\mathbf{SZ}$, $\mathbf{SY}_{\substack{i,j \\ i \neq j}}$ that minimize the system objective function. A constant local variables design vector $\mathbf{X}^*$, obtained from the domain optimizations, is employed during this system level optimization. The system level optimization is subject to an equality constraint for each domain that ensures that the design variables optimized at the domain level match the target values of the surrogate variables at the optimum. The surrogate design variables are passed to the domains and are used as targets for the domain optimizations. The domain optimizers minimize an objective function composed of the sum of the l_2 norms of the differences between the design variables and their target complimentary values subject to the local constraints of the analysis. At the optimum, the

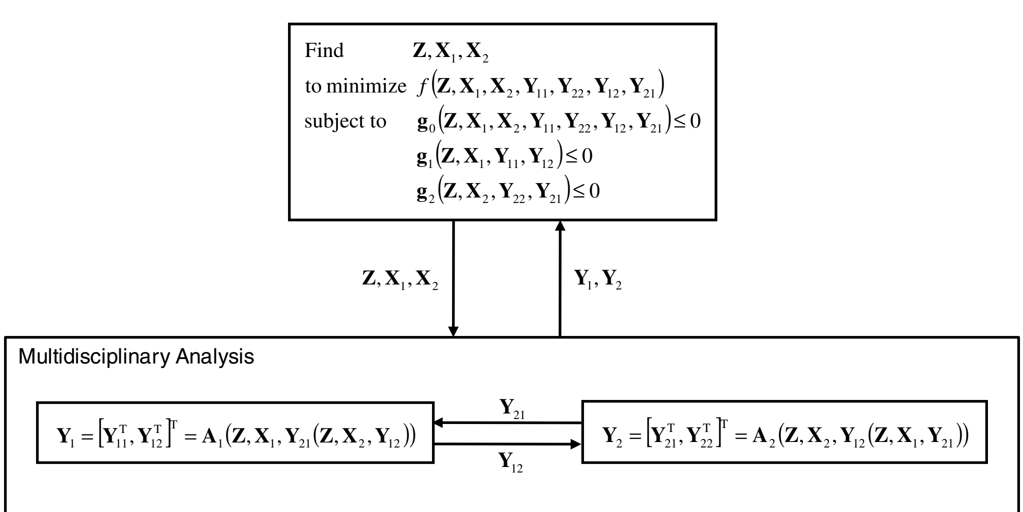

Fig. 4 IDF method.

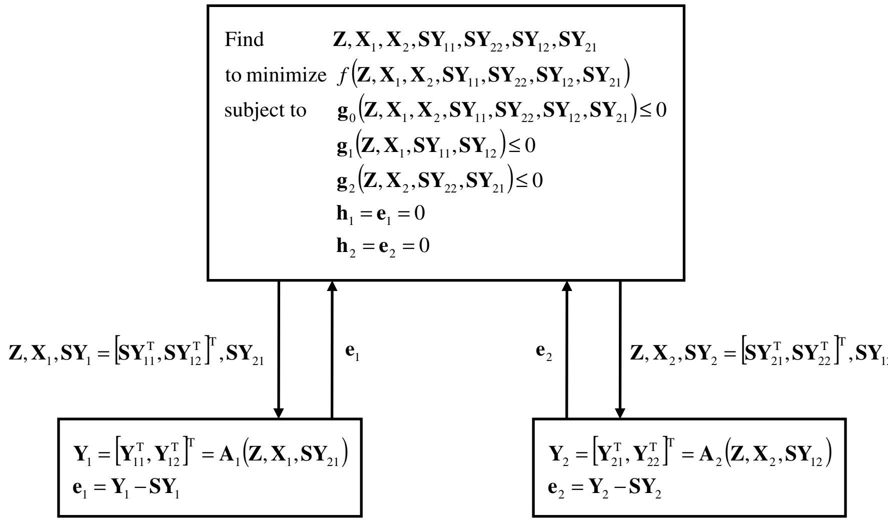

Fig. 5 AAO method.

domains should all be able to match their target variables, subject to local constraints, and the result is multidiscipline feasible.

The system level optimization is:

$$\text{given } \mathbf{X}^* \tag{22}$$

$$\text{find } \mathbf{SZ}, \mathbf{SY}_{\substack{i,j \\ i\neq j}} \tag{23}$$

$$\text{to minimize } f(\mathbf{SZ}, \mathbf{X}^*, \mathbf{Y}(\mathbf{SZ}, \mathbf{X}^*, \mathbf{SY})) \tag{24}$$

$$\text{subject to } J_i^*(\mathbf{Z}, \mathbf{SZ}, \mathbf{Y}, \mathbf{SY}_{\substack{i,j \\ i\neq j}}) = 0 \qquad i = 1\dots N \tag{25}$$

$$\text{where } \mathbf{Y}_i = \mathbf{A}_i(\mathbf{SZ}, \mathbf{X}_i^*, \mathbf{SY}_{\substack{j=1\dots N,i \\ i\neq j}}) \qquad i = 1\dots N \tag{26}$$

The ith domain optimization task is:

$$\text{given } \mathbf{SZ}, \mathbf{SY}_{\substack{j,i \\ j\neq i}}, \mathbf{SY}_{\substack{i,j \\ j\neq i}} \qquad j = 1\dots N \tag{27}$$

$$\text{find } \mathbf{Z}, \mathbf{X}_i, \mathbf{Y}_{\substack{j=1\dots N,i \\ j\neq i}} \tag{28}$$

$$\text{to minimize } J_i = (\mathbf{Z} - \mathbf{SZ})^2 + (\mathbf{Y}_{\substack{i,j=1\dots N \\ j\neq i}} - \mathbf{SY}_{\substack{i,j=1\dots N \\ j\neq i}})^2 + (\mathbf{Y}_{\substack{j=1\dots N,i \\ j\neq i}} - \mathbf{SY}_{\substack{j=1\dots N,i \\ j\neq i}})^2 \tag{29}$$

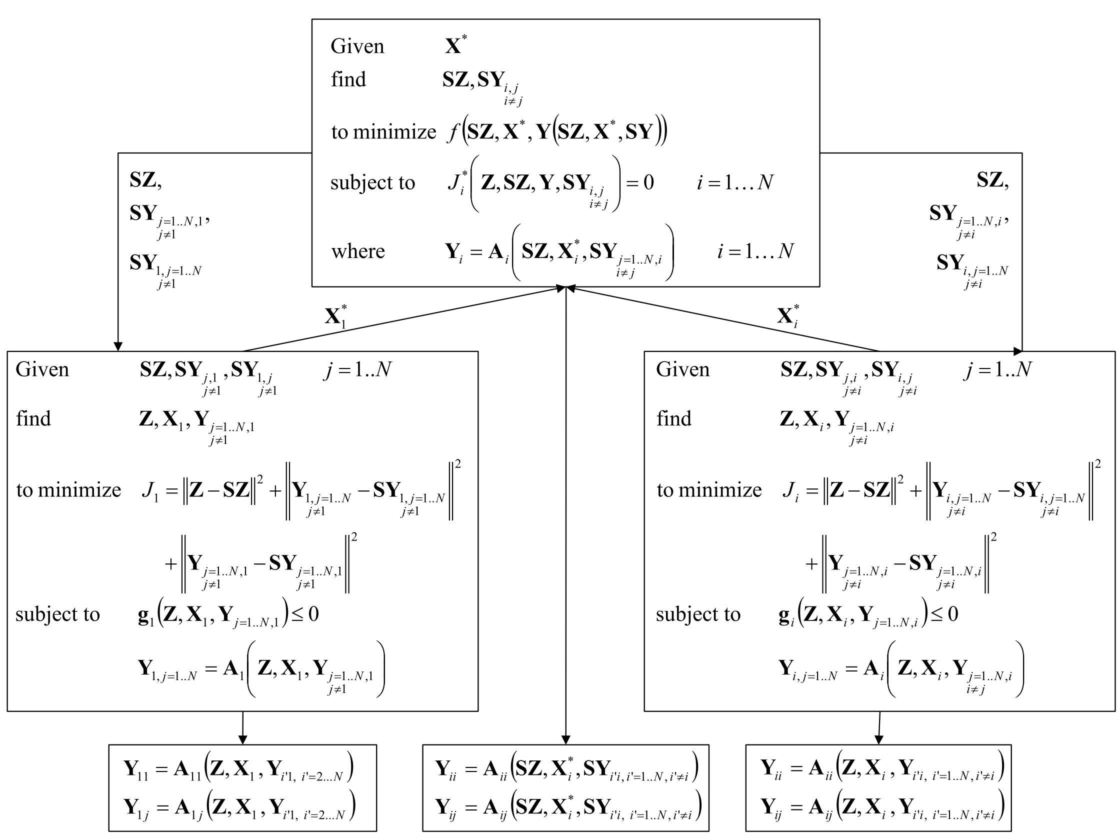

Fig. 6 CO method.

$$\text{subject to } g_i(\mathbf{Z}, \mathbf{X}_i, \mathbf{Y}_{\substack{j=1\ldots N,i \\ i\neq j}}) \leq 0 \tag{30}$$

$$\text{where } \mathbf{Y}_{i,j=1\ldots N} = \mathbf{A}_i(\mathbf{Z}, \mathbf{X}_i, \mathbf{Y}_{\substack{j=1\ldots N,i \\ i\neq j}}) \tag{31}$$

The method is illustrated for two domains in Fig. 6.

E. Bilevel Integrated System Synthesis

The bilevel integrated system synthesis (BLISS) method was introduced by Sobieszczanski–Sobieski et al. [6] and is a decomposition extension of the global sensitivity equations (GSEs). The process flow of the method is shown in Fig. 7. Each cycle of the method starts with a system analysis to establish the state variables $\mathbf{Y}$. The total derivatives of the coupling variables $\mathbf{Y}$ with respect to the local design variables $\mathbf{X}$ are evaluated by solving the GSEs. Each domain then optimizes the local design variables $\mathbf{X}_i$ to find an optimal change in the system objective function, subject to local constraints, while holding the shared design variables vector $\mathbf{Z}$ constant. Optimum sensitivity analysis is then evaluated by solving the GSE/OS equations [7] to find the total derivatives of the system objective function with respect to $\mathbf{Z}$. The shared design variables are then optimized in the system level, holding the local design variables constant. The optimal changes to the local and shared design variables are applied and a new cycle is initiated. At each cycle the method reduces the system objective function or improves the constrained state variables.

The optimization of the ith discipline takes the form:

$$\text{find } \Delta\mathbf{X}_i \tag{32}$$

$$\text{to minimize } d(\mathbf{Y}_{1i}, \mathbf{X}_i) \cdot \Delta\mathbf{X}_i \tag{33}$$

$$\text{subject to } g_i(\mathbf{X}_i) \leq 0 \tag{34}$$

where $d(\mathbf{Y}_{1i}, \mathbf{X}_i)$ is obtained by solving the GSE and represents the first-order predicted change in the system objective function due to changes in the local domain variables $\mathbf{X}_i$.

The system level optimization seeks the change in the shared design variables $\mathbf{Z}$ that reduces the first-order predicted change in the system objective function while holding the local design variables $\mathbf{X}$ constant:

$$\text{find } \Delta\mathbf{Z} \tag{35}$$

$$\text{to minimize } f = f_0 + d(f, Z) \cdot \Delta\mathbf{Z} \tag{36}$$

$$\text{subject to } g_i(\mathbf{Z}, \mathbf{X}_i, \mathbf{Y}(\mathbf{Z}, \mathbf{X}_i)) \leq 0 \tag{37}$$

The BLISS/A method [6] is used in this implementation to evaluate the changes in the objective function with respect to the shared design variables vector $\mathbf{Z}$. A trust region technique [8] is used to ensure that the step sizes are appropriate when using the first-order predicted changes of the system objective function. The BLISS method achieves discipline autonomy and seeks to make optimal changes to the shared and local design vectors at each cycle to reduce the system objective function or to reduce constraint violations. At each cycle the system design can be reviewed by human decision makers to ensure appropriate progress is being made.

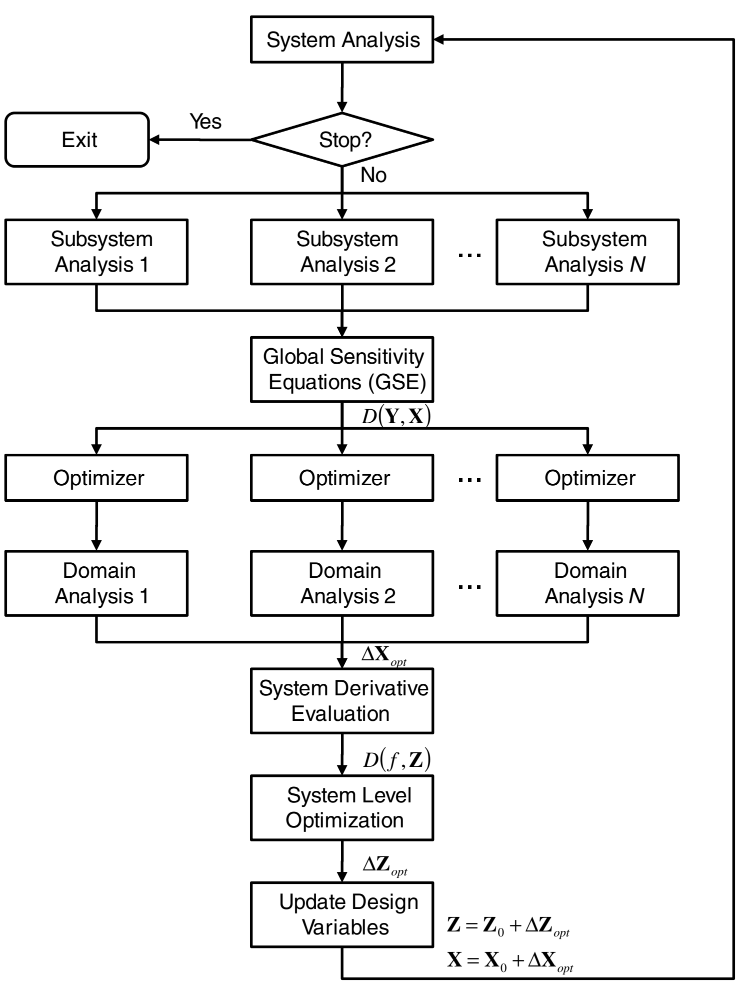

Fig. 7 Process diagram of the BLISS method.

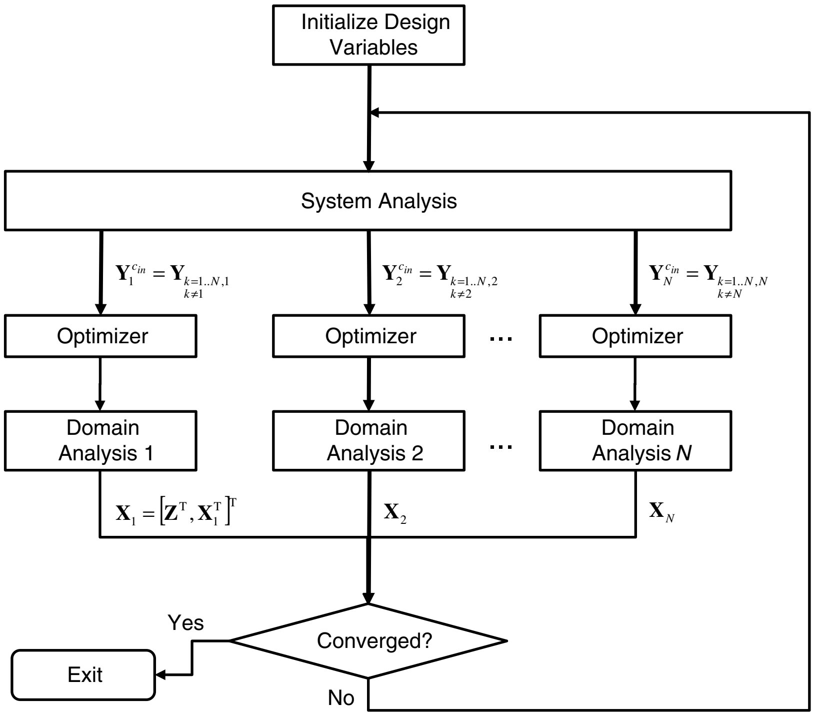

Fig. 8 Process diagram of the MDO method based on independent subspaces.

F. Multiple Disciplinary Optimization of Independent Subspaces

Shin and Park [9] describe the method of multidisciplinary design optimization based on independent subspaces (MDOIS). The method aims to support the optimization of a decomposed system design in which the subsystems complement existing expertise or domain analysis tools. Each domain formulates an optimization problem consisting of local design variables, objective function and constraints. The method first solves the coupled problem by evaluating the system analysis. The resulting state coupling information is passed to the domains and is treated as constant within the disciplinary optimization problems. The domains are termed subspaces and are treated as independent design problems. An iterative process of independent subspace optimizations followed by system analysis is employed until convergence criteria are satisfied. The method can be optionally augmented by solving the global sensitivity equations at each cycle where changes to the state variables are modeled in the subspaces during domain optimization. The process is shown in Fig. 8.

MDOIS requires the separation of the problem into independent optimization subspaces. To fully satisfy the criteria of the method there should not be any shared design variables nor a system objective function or system level constraints. These criteria are not always possible to meet and the method is not appropriate for all MDO problems. In some cases the design team may attempt to formulate the problem such that the shared design variables are optimized in one domain only. For the suite of problems that we present in the next section we highlight where necessary the division of shared design variables across the subspaces.

The MDOIS method is developed to accommodate the design autonomy of domains. The rule based approach that is proposed in this paper is similar in construction to MDOIS but aims to control and optimize the shared design variables of the problem at the system level. The MDOIS method could in principle be established within the expert system by implementing a few simple rules: 1) if all domains have posted a result ⇒ perform system analysis, 2) if system analysis has been performed ⇒ exchange state coupling variables, and 3) if domain inactive ⇒ Configure domain optimization problem to find optimal local design variables vector holding the state coupling variables constant.

IV. Numerical MDO problems

A number of multidisciplinary design optimization problems, ranging from simple numerical examples through to conceptual aircraft design problems, are now introduced. The challenges that each problem provides to the MDO algorithms are discussed. The first simple numerical problems provide analytic solutions against which the precision of the MDO algorithms can be judged. The conceptual aircraft design tools provide a more representative challenge of the class of problem that the rule base approach will encounter in production.

A. Example Problem Shin2, Yi1

This example is the second problem discussed in Shin and Park [9] and also investigated as the first example problem of Yi et al. [10]. The problem was originally developed for testing CSSO in the commercial Isight package. The problem is defined as:

$$\text{find } 0.0 \le b_1 \le 7.0, \qquad 2.0 \le b_2 \le 7.0, \qquad 2.0 \le b_3 \le 7.0 \tag{38}$$

$$\text{to minimize } f(\mathbf{X}) = y_2 + y_5 \tag{39}$$

$$\text{subject to } g_1 = -y_3 \le 0.0 \tag{40}$$

$$g_2 = -y_6 \le 0.0 \tag{41}$$

$$\text{where } \mathbf{X} = [b_1, b_2, b_3]^T \tag{42}$$

$$\mathbf{Y} = [y_1, y_2, y_3, y_4, y_5, y_6]^T \tag{43}$$

$$y_1 = b_1^2 + b_2 + b_3 - 0.2y_4 - 4.0 \tag{44}$$

$$y_2 = y_1 + (b_2 - 2.0)^2 \tag{45}$$

$$y_3 = \frac{y_1}{8.0} - 1.0 \quad (46)$$

$$y_4 = b_1 + b_3 - \sqrt{y_1} - 2.0 \quad (47)$$

$$y_5 = b_3 - 2.0 - e^{-y_4} \quad (48)$$

$$y_6 = 1.0 - \frac{y_4}{10.0} \quad (49)$$

For the purposes of investigation here the problem is decomposed into two disciplines that share a common design variable $\mathbf{Z} = b_1$ with each domain having a single local design variable $\mathbf{X}_1 = b_2, \mathbf{X}_2 = b_3$. To evaluate the analysis function in the first domain the variable b_3 is treated as constant. An extra element in the state coupling vector $Y_{21} = [y_4, b_3]^T$ is introduced which takes the value of b_3 evaluated in domain 2. The subspace optimization problems are expressed as:

Domain 1:

$$\text{given } \mathbf{Y}_{21} = [y_4, b_3]^T \quad (50)$$

$$\text{find } 0.0 \le \mathbf{Z} = [b_1] \le 7.0 \quad (51)$$

$$2.0 \le \mathbf{X}_1 = [b_2] \le 7.0 \quad (52)$$

$$\text{to minimize } f_1 = \mathbf{Y}_{11}^{(1)} \quad (53)$$

$$\text{subject to } g_1 = \mathbf{Y}_{11}^{(2)} \le 0.0 \quad (54)$$

$$\text{where } \mathbf{Y}_{11} = [y_2, -y_3]^T \quad (55)$$

$$\mathbf{Y}_{12} = [y_1] \quad (56)$$

Domain 2:

$$\text{given } \mathbf{Y}_{12} = [y_1] \quad (57)$$

$$\text{find } 0.0 \le \mathbf{Z} = [b_1] \le 7.0 \quad (58)$$

$$2.0 \le \mathbf{X}_2 = [b_3] \le 7.0 \quad (59)$$

$$\text{to minimize } f_2 = \mathbf{Y}_{22}^{(1)} \quad (60)$$

$$\text{subject to } g_2 = \mathbf{Y}_{22}^{(2)} \le 0.0 \quad (61)$$

$$\text{where } \mathbf{Y}_{22} = [y_5, -y_6]^T \quad (62)$$

$$\mathbf{Y}_{21} = [y_4, b_3]^T \quad (63)$$

There are other possible decompositions of the problem and this partitioning is chosen because it is amenable to all of the algorithms to be investigated. The problem has an optimum objective function value of $f = 8.0029$ at the design point $\mathbf{Z} = [3.0284]$ and $\mathbf{X} = [2.0000, 2.0000]$.

B. Yi3

The next numerical problem is taken from the third example study presented in Yi et al. [10] and again involves two disciplines:

$$\text{find } 0.0 \le \mathbf{Z}, \mathbf{X}_1, \mathbf{X}_2 \le 10.0 \quad (64)$$

$$\text{to minimize } f = f_1 + f_2 = (\mathbf{Y}_{11} - 0.5)^2 + (\mathbf{Y}_{22} - 0.5)^2 \quad (65)$$

$$\text{subject to } g_1 = 1.0 - \mathbf{Y}_{11} \le 0.0 \quad (66)$$

$$g_2 = 1.0 - \mathbf{Y}_{22} \le 0.0 \quad (67)$$

$$\text{where } \mathbf{Y}_{12} = (\mathbf{X}_1 - 2.5) + (\mathbf{Z} - 2.0) - 0.4\mathbf{Y}_{21} \quad (68)$$

$$\mathbf{Y}_{11} = (\mathbf{X}_1 - 2.5) + (\mathbf{Z} - 2.0) - 0.5\mathbf{Y}_{21} \quad (69)$$

$$\mathbf{Y}_{21} = (\mathbf{X}_2 - 3.0) + (\mathbf{Z} - 2.0) - 0.6\mathbf{Y}_{12} \quad (70)$$

$$\mathbf{Y}_{22} = (\mathbf{X}_2 - 3.0) + (\mathbf{Z} - 2.0) - 0.7\mathbf{Y}_{12} \quad (71)$$

The problem has an optimum objective function value of $f = 0.5$ but there is no unique global solution; the problem admits multiple solutions.

C. Simple Subsonic Passenger Transport Aircraft Design Problem (LewisC)

A subsonic passenger aircraft design problem is described by Lewis [11] and is investigated in Zhang et al. [12]. The problem is composed of two domains; an aerodynamics model and a weights model. Within the aerodynamics model fundamental flight properties of the design are evaluated including lift-to-drag ratios for takeoff, landing, and cruise conditions. The weights domain calculates the useful payload and evaluates takeoff and landing performance of the aircraft. The problem investigated by Lewis defines l_f, b, and T_i as discrete variables taking integer values between the bounds for l_f and b and specifying a discrete set of permissible values for T_i. In this study we relax these conditions and use continuous variables for these quantities.

The aerodynamics model calculates the cruise velocity V_{br} for the design giving maximum range of the aircraft and evaluates the corresponding drag coefficient C_{d0c}. There is a cyclic dependency between these state variables and fixed point iteration is employed to solve the equations. The system problem can be treated as a two domain multidisciplinary optimization problem or the aerodynamics model could be further decomposed to form a three domain problem. The complexity of a three domain division is not justified so we restrict our investigation to the two domain decomposition and use fixed point iteration within the aerodynamics discipline to resolve the values of V_{br} and C_{d0c}. The domain dependencies are illustrated in Fig. 9.

For the purposes of this investigation the design problem is decomposed into two optimizations. The domains share two of the design variables $\mathbf{Z} = [S, W_{\mathrm{To}}]^T$ with the remaining design variables partitioned as local variables; $\mathbf{X}_1 = [l_f, b]^T$, $\mathbf{X}_2 = [T_i]^T$. The domain optimization problems are:

Aerodynamics Domain Optimization

$$\text{given } \mathbf{Y}_{21} = [R_{\mathrm{fr}}]^T \quad (72)$$

$$\text{find } 1200.0 \le S \le 2500.0 \quad (73)$$

$$140000.0 \le W_{\mathrm{To}} \le 250000.0 \quad (74)$$

$$105.0 \le l_f \le 150.0 \quad (75)$$

$$85.0 \le b \le 140.0 \quad (76)$$

$$\text{to minimize } f_1 = -(L/D)_c/20.0 \quad (77)$$

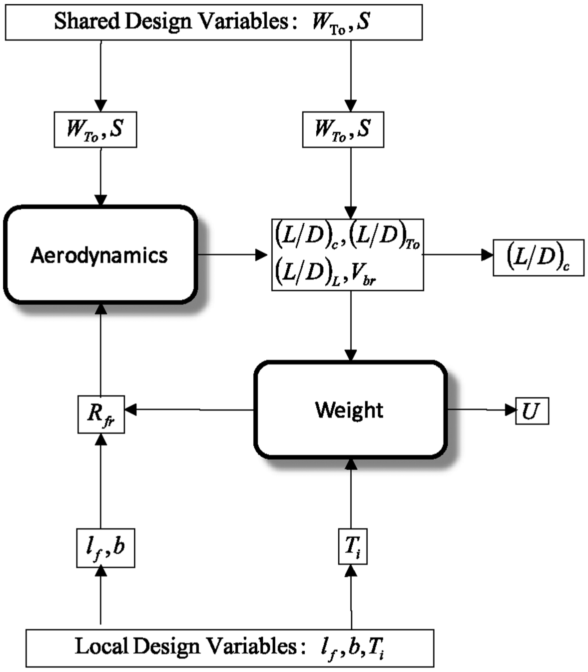

Fig. 9 Data dependencies in the LewisC aircraft design problem.

$$\text{subject to } g_1 = \mathrm{AR} - 10.5 \le 0.0 \tag{78}$$

$$g_2 = C_{d0L} - 0.02 \le 0.0 \tag{79}$$

$$g_3 = C_{d0c} - 0.02 \le 0.0 \tag{80}$$

Weights Domain Optimization

$$\text{given } \mathbf{Y}_{12} = [V_{\mathrm{br}}, (L/D)_c, (L/D)_{\mathrm{To}}, (L/D)_L]^T \tag{81}$$

$$\text{find } 1200.0 \le S \le 2500.0 \tag{82}$$

$$140000.0 \le W_{\mathrm{To}} \le 250000.0 \tag{83}$$

$$27750.0 \le T_i \le 65000.0 \tag{84}$$

$$\text{to minimize } f_2 = -U/0.5 \tag{85}$$

$$\text{subject to } g_1 = 0.3 - U \le 0.0 \tag{86}$$

$$g_2 = 1.0 - R_f \le 0.0 \tag{87}$$

$$g_3 = 0.027 - q_{\mathrm{To}} \le 0.0 \tag{88}$$

$$g_3 = 0.024 - q_L \le 0.0 \tag{89}$$

$$g_4 = D_{\mathrm{To}} - 6500.0 \le 0.0 \tag{90}$$

$$g_5 = D_L - 4500.0 \le 0.0 \tag{91}$$

The domain objectives are weighted such that their values are of order unity. The system objective for the problem, the sum of the domain objectives, is then reasonably balanced between the disciplines.

D. Conceptual Supersonic Business Jet Aircraft Design Problem

In demonstrating the effectiveness of the BLISS method Sobieszczanski–Sobieski et al. [6] presents a conceptual supersonic business jet (SBJ) aircraft design problem. The Matlab source code for the problem is provided in the appendix of a publicly available release of the BLISS manuscript on the NASA Technical Reports Server [13]. The code that comprises the domain analysis functions has been ported to our framework to provide a MDO problem with four domains; structures, aerodynamics, propulsion and range. The objective of the problem is to maximize the Breguet range of the aircraft. The problem is composed of six shared design variables, four local variables and ten principle state variables (see Fig. 10).

The constraints for this problem are not entirely clear and there is some discussion in the literature about what bounds to use [14]. Even within the manuscript from which the source code is obtained there is an inconsistency between the documented problem and the implementation of that problem. The limits implemented in the reference code are the most restrictive and are retained here. For the sake of clarity it is therefore highlighted that the upper limit on wing stress is set at 1.05 as opposed to the value of 1.09 reported in the literature. Similarly the limits on wing twist are set at $0.97 \le \Theta \le 1.03$ as opposed to the range $0.96 \le \Theta \le 1.04$ reported in the original paper. It is noted that Wang et al. reduced the lower bound to $0.9094 \le \Theta \le 1.04$ to admit the optimal solution reported by Sobieszczanski–Sobieski et al. [6] in the original BLISS paper. The implementation here of the analysis codes (copied-and-pasted from the original manuscript) reproduces the data points quoted by Sobieszczanski–Sobieski et al. [6] to reasonable accuracy. Agreement is found with the values reported by Wang et al. [14] in all domains except for the weights analysis. The global optimum design reported in the original paper is reproduced in this implementation and violates the wing twist constraint as reported by Wang et al. [14]. However, we are unable to find constraint limits for which the global optimum of the problem coincides with that reported by Sobieszczanski–Sobieski et al. [6]. Since the differences are minor, the more restrictive wing twist constraint boundary is retained. The optimum range for the investigation of this problem is found to be 3963.77 nm using a number of the MDO methods and is consistent with values reported in the literature [6,13,15–17].

For the rule base framework and the MDOIS method a decomposition of the SBJ problem is used in which each domain optimises its own contribution to the Breguet range equation. The domain optimization problems for these methods are as follows:

Weights Domain Optimization

$$\text{given } \mathbf{Y}_{21} = [L]^T, \mathbf{Y}_{31} = [W_E]^T \tag{92}$$

$$\text{find } t/c, \Lambda, \mathrm{AR}, S_{\mathrm{ref}}, \lambda, x \tag{93}$$

$$\text{to minimize } f_1 = -(W_T/(W_T - W_F)) \tag{94}$$

$$\text{subject to } g_{1-5} = \sigma_{1-5} - 1.05 \le 0.0 \tag{95}$$

$$g_6 = 0.97 - \Theta \le 0.0 \tag{96}$$

$$g_7 = \Theta - 1.03 \le 0.0 \tag{97}$$

Aerodynamics Domain Optimization

$$\text{given } \mathbf{Y}_{12} = [W_T, \Theta]^T, \mathbf{Y}_{32} = [\mathrm{ESF}]^T \tag{98}$$

$$\text{find } t/c, h, M, \Lambda, \mathrm{AR}, S_{\mathrm{ref}}, C_f \tag{99}$$

$$\text{to minimize } f_2 = -(L/D) \tag{100}$$

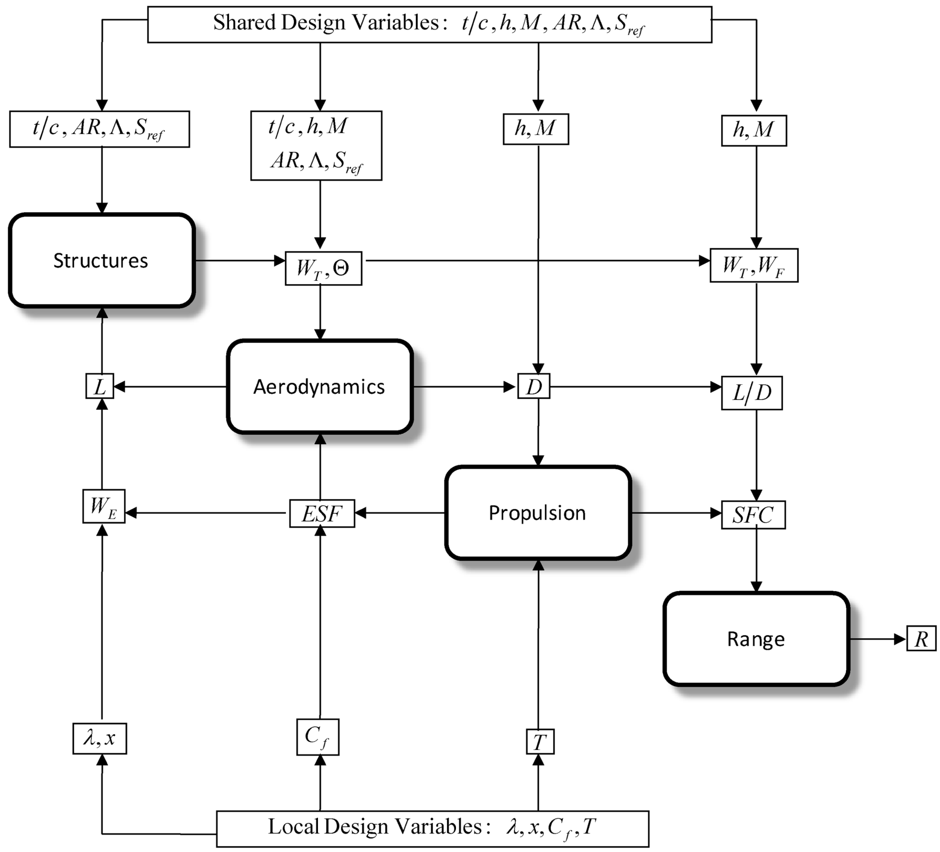

Fig. 10 Data dependencies in the SBJ problem.

$$\text{subject to } g_1 = \mathrm{d}p/\mathrm{d}x - 1.04 \le 0.0 \tag{101}$$

Propulsion Domain Optimization

$$\text{given } \mathbf{Y}_{23} = [D]^T \tag{102}$$

$$\text{find } h, M, T \tag{103}$$

$$\text{to minimize } f_3 = \text{SFC} \tag{104}$$

$$\text{subject to } g_1 = 0.5 - \text{ESF} \le 0.0 \tag{105}$$

$$g_2 = \text{ESF} - 1.5 \le 0.0 \tag{106}$$

$$g_3 = \text{Temp} - 1.02 \le 0.0 \tag{107}$$

$$g_1 = \bar{T} - \bar{T}_{\text{UA}} \le 0.0 \tag{108}$$

Range Domain Optimization

$$\text{given } \mathbf{Y}_{14} = [W_T, W_F]^T, \qquad \mathbf{Y}_{24} = [L/D]^T, \qquad \mathbf{Y}_{34} = [\text{SFC}]^T \tag{109}$$

$$\text{find } h, M \tag{110}$$

$$\text{to minimize } f_2 = -R \tag{111}$$

For the BLISS and CO methods the domain optimization objectives are defined by the methods themselves as discussed above.

E. TADPOLE: Commercial Aircraft Wing Design Tool

The final MDO problem considered here is derived from a commercial in-house tool for preliminary concept design of civil transport aircraft wings. A cutdown version of the empirical design code TADPOLE [18,19] is used to evaluate the performance characteristics of a given wing design. The analysis model is decomposed into three domains that evaluate drag, lift and weight based upon empirical or empirically corrected analytic models. The wing performance calculation is accompanied by a cost evaluation module detailed in [20]. The cost model uses the hierarchical modelling tool Vanguard Studio [21] which derives a cost for the wing given the planform parameters, t/c at key sections and the top and bottom cover weights evaluated in the weights domain. The model incorporates part types (stringers, spars, ribs and skins), manufacturing processes and materials to derive an estimate for the cost of manufacture of the wing. Collectively, the four domain analysis tools provide a representative "real-life" design challenge while remaining sufficiently computationally efficient to allow extensive study of the various MDO methods, domain analysis calls requiring $\mathcal{O}(1s)$ CPU time per evaluation.

The design problem is constructed with the aim of optimizing the wing for operation at Mach 0.785 and a Reynolds number of 7.3 million. The constraints on wing weight, volume, pitch-up margin and root triangle layout are chosen to be representative of a 220-seat wide body airliner. The design problem is detailed in Table 1. The data dependencies of the problem are shown in Fig. 11. The state coupling data of the problem includes two vector quantities exchanged between the aerodynamics and structures domains. The structures domain provides an array of structural cell boundary locations $\boldsymbol{\eta}_s$ defining wing strips for which the aerodynamics analysis model evaluates the corresponding lift coefficients $\mathbf{C}_L$. Given the lift coefficients of the wing strips the structures domain evaluates the

Table 1 TADPOLE design problem

	Variable	Lower limit	Symbol	Upper limit	Units
Find	Wing area	100.0	S	250.0	m^2
	Aspect ratio	6.0	AR	12.0	——
	Normalized spanwise kink position	0.2	η_k	0.45	——
	Inner panel leading-edge sweep	25.0	Λ	45.0	°
	Inner panel taper ratio	0.4	λ_1	0.7	——
	Outer panel taper ratio	0.2	λ_2	0.6	——
	Root thickness to chord ratio	0.1	$(t/c)_r$	0.18	——
	Kink thickness to chord ratio	0.06	$(t/c)_k$	0.14	——
	Tip thickness to chord ratio	0.06	$(t/c)_t$	0.14	——
	Wing washout	4.0	w	5.0	°
	Fraction of tip washout at kink position	0.65	k	0.85	——
To minimize	Drag		D/q		m^2
	Lift (negated)		$-C_L$		
	Aircraft weight at cruise		W_c		N
	Wing cost		Cost		£ m
Subject to	Estimated wing weight	——	$\tilde{W}$	1.35×10^5	N
	Estimated wing volume	40.0	$\tilde{V}$	——	m^3
	Pitch up margin	——	p	5.4	——
	Undercarriage bay length	2.5	l_U	——	m

positioning of the cell boundaries and calculates the weights of the fuel tanks, fuel, structural components and the overall wing weight.

To make comparisons between the domains of the decomposed TADPOLE tool at the system level some comparative measure of performance needs to be made. We employ a direct operating cost (DOC) measure as the system level objective function which is constructed as a linear sum of domain level cost evaluations derived from the physical attributes of the design that they are optimizing. The direct operating cost of the aircraft (wing) is composed of:

1) The cost penalty for drag evaluated as the fuel burn required to maintain thrust equal to drag at cruise (the drag penalty).

2) The cost improvement for lift evaluated as the fuel saving derived from the lift coefficient (the lift factor).

3) The cost penalty of weight evaluated as the fuel burn at cruise derived from the weight of the vehicle (the weight penalty).

4) The annualized capital outlay (the cost of manufacture as derived from the cost model).

We derive weighting factors/penalties for the domain objectives to provide a DOC summation for the system objective function

$$\text{sof} = \text{DOC} = p_D D - p_L L + p_W W + C_{\text{man}} \tag{112}$$

where C_{man} is the cost of manufacture (in £m) as estimated by the cost model. The weight of the aircraft W, the lift force L, and the induced drag D (evaluated in the weight, lift, and drag domains) are converted to cost figures through the weight penalty p_W (in £m/kg), the lift factor p_L (in £m/N) and the drag penalty p_D (in £m/m^2). The weighting penalties are derived using a number of assumptions about economic factors (mission profile, fuel costs, typical fuel cost per passenger seat, depreciation). We follow Kaufmann et al. [22] and estimate the weight, lift, and drag penalties by performing a simple fuel burn calculation.

Following Kaufmann et al. [22], we assume that the aircraft flies for 25 years, 250 days/year and at a range of $2 * 8000$ km/day,

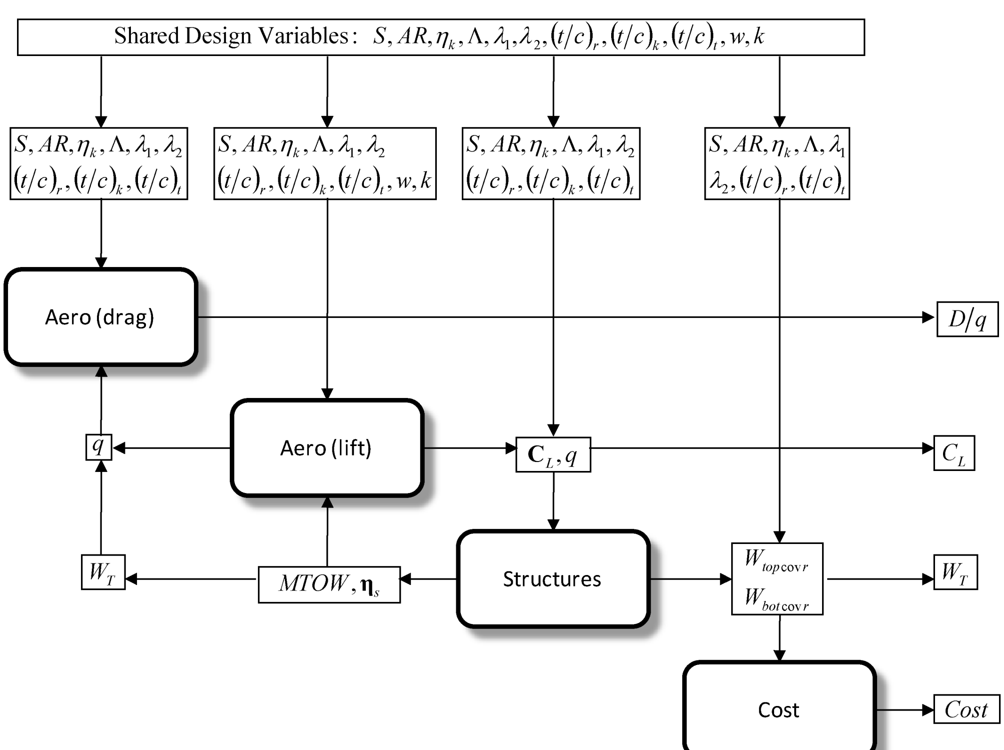

Fig. 11 Data dependencies of the TADPOLE design problem.

Table 2 Results of the MDO methods applied to the problem labeled Shin2

Method	Objective	Maximum constraint ($g <= 0$)	Number of system iterations (system analysis calls)	Number of analysis calls		Shared design variables vector Z	Local design variables vector X	State variables vector Y	
				Dom. 1	Dom. 2				
MDF	8.0029	-2.0475×10^{-9}	4(9)	279	279	$b_1 = 3.0284$	$b_2 = 2.0000$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8569$ $y_5 = 2.8602 \times 10^{-3}$
IDF	8.0029	-2.2204×10^{-16}	8	64	64	$b_1 = 3.0284$	$b_2 = 2.0000$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8569$ $y_5 = 2.8602 \times 10^{-3}$
AAO	8.0029	-2.2204×10^{-16}	9	91	91	$b_1 = 3.0284$	$b_2 = 2.0000$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8569$ $y_5 = 2.8602 \times 10^{-3}$
CO	8.0029	-8.2334×10^{-7}	59	14413	7773	$b_1 = 3.0284$	$b_2 = 2.0001$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8568$ $y_5 = 2.8604 \times 10^{-3}$
BLISS	8.0029	-3.8003×10^{-11}	7(7)	202	202	$b_1 = 3.0284$	$b_2 = 2.0000$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8569$ $y_5 = 2.8602 \times 10^{-3}$
MDOIS	8.0029	-3.8177×10^{-10}	5(5)	209	153	$b_1 = 3.0284$	$b_2 = 2.0000$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8569$ $y_5 = 2.8602 \times 10^{-3}$
Rule base (SQP)	8.0039	-1.4400×10^{-4}	23	990	978	$b_1 = 3.0284$	$b_2 = 2.0000$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8569$ $y_5 = 2.8602 \times 10^{-3}$
Rule base (GA)	8.0029	-1.5942×10^{-8}	22	126672	126672	$b_1 = 3.0284$	$b_2 = 2.0000$ $b_3 = 2.0000$	$y_1 = 8.0000$ $y_2 = 8.0000$	$y_4 = 5.8569$ $y_5 = 2.8602 \times 10^{-3}$

Table 3 Results of the MDO methods applied to the problem labeled Yi3

Method	Objective	Maximum constraint ($g <= 0$)	Number of system iterations (system analysis calls)	Number of analysis calls		Shared design variables vector Z	Local design variables vector X	State variables vector Y	
				Dom. 1	Dom. 2				
MDF	0.5000	0.0000	2(5)	281	281	$b_1 = 2.6112$	$b_2 = 3.4443$ $b_3 = 4.1665$	$y_{11} = 1.0000$ $y_{12} = 1.1111$	$y_{21} = 1.1111$ $y_{22} = 1.0000$
IDF	0.5000	1.1102×10^{-16}	2	19	19	$b_1 = 2.6111$	$b_2 = 3.4444$ $b_3 = 4.1667$	$y_{11} = 1.0000$ $y_{12} = 1.1111$	$y_{21} = 1.1111$ $y_{22} = 1.0000$
AAO	0.5000	-2.2204×10^{-16}	3	33	33	$b_1 = 2.6111$	$b_2 = 3.4444$ $b_3 = 4.1667$	$y_{11} = 1.0000$ $y_{12} = 1.1111$	$y_{21} = 1.1111$ $y_{22} = 1.0000$
CO	0.4998	2.0609×10^{-4}	148	19530	18376	$b_c = 2.5952$	$b_1 = 3.4602$ $b_2 = 4.1824$	$y_{11} = 0.9998$ $y_{12} = 1.1111$	$y_{21} = 1.1109$ $y_{22} = 1.0000$
BLISS	0.5000	2.6671×10^{-7}	66(66)	4941	4950	$b_c = 0.0000$	$b_1 = 6.0556$ $b_2 = 6.7778$	$y_{11} = 1.0000$ $y_{12} = 1.1111$	$y_{21} = 1.1112$ $y_{22} = 1.0001$
MDOIS	0.5000	-4.2723×10^{-8}	21(21)	1168	1184	$b_c = 3.6365$	$b_1 = 2.4191$ $b_2 = 3.1413$	$y_{11} = 1.0000$ $y_{12} = 1.1111$	$y_{21} = 1.1111$ $y_{22} = 1.0000$
Rule base (SQP)	0.5000	5.5263×10^{-7}	20	269	310	$b_c = 3.5382$	$b_1 = 2.5174$ $b_2 = 3.2396$	$y_{11} = 1.0000$ $y_{12} = 1.1111$	$y_{21} = 1.1111$ $y_{22} = 1.0000$
Rule base (GA)	0.5001	-1.0501×10^{-4}	19	108576	108576	$b_c = 2.5820$	$b_2 = 3.4733$ $b_3 = 4.1959$	$y_{11} = 1.0001$ $y_{12} = 1.1111$	$y_{21} = 1.1112$ $y_{22} = 1.0000$

giving an overall lifetime flight distance of 100 million kilometers. Further, assuming that the A310 class has a fuel efficiency of 20.3 g/(passenger · km)[§] carrying 220 passengers gives an aircraft fuel efficiency of 4.466×10^{-3} kg/m. Taking the price of aviation jet fuel as 643.1 /mt[¶] the aircraft fuel cost rate is 2.872×10^{-3} /m and therefore a fuel cost per 8000 km flight of 2.298×10^4 . Estimating an average passenger plus baggage weight of 100 kg we derive the weight penalty $p_W = 1.044394$ /kg as the aircraft fuel cost per flight per kg. The same factor is applicable to the scaling of the lift and with a conversion of units we have $p_L = 1.065 \times 10^{-1}$ /N.

A similar line of reasoning is applied to convert the drag evaluation into an equivalent fuel cost required to maintain cruise velocity per flight. Taking the typical specific fuel consumption for a jet engine on this class of aircraft as $\mathrm{SFC} = 1.714 \times 10^{-5}$ kg/N · s (CF6-80C2B1F turbofan engine (Boeing 747-400 cruise, Airbus A310), Wikipedia), converting the thrust from Newton's to square meters at a dynamic pressure of $q = 1.2667 \times 10^4$ N/m^2 and assuming the fuel cost above and a flight time of 3.366×10^4 s we obtain a drag penalty of $p_D = 4.699 \times 10^3$ /m^2.

Finally, converting all factors to pound sterling using a rate of 1.5 /£ the penalties enable the system objective function to be evaluated as a DOC including the manufacturing cost model.

V. Results

The MDO algorithms described in Sec. III have been applied to the problem suite outlined in Sec. IV. In this section we summarize the performance of each method. For all methods the system analysis is performed using an iterative Jacobi solver unless otherwise stated.

As discussed above, the rule base approach is employed using both the *fmincon* SQP algorithm and a genetic algorithm in the domains. Our intentions are primarily to assess the performance of the rule base in managing the shared design variable bounds and exchange of state variables to reach a multiple discipline feasible solution.

The results of the methods applied to the Shin2 problem are summarized in Table 2. The problem is solved by all methods. For this simple problem the optimizer-based methods, with the exception of CO, solve the problem using about ten system iterations. The CO

[§]Additional data available at http://www.airlines-inform.com/commercial aircraft/Airbus-A310.html [accessed Jan. 2010].

[¶]Additional data on IATA Jet Fuel Price Monitor available at http://www.iata.org/whatwedo/economics/fuel_monitor/index.htm [accessed Jan. 2010].

Table 4 Results of the MDO methods applied to the problem labeled LewisC

Method	Objective	Maximum constraint ($g < = 0$)	Number of system iterations (system analysis calls)	Number of analysis calls: Aero.	Number of analysis calls: Str.	Shared design variables vector Z	Local design variables vector X	State variables vector Y	
MDF	−2.0676	-5.7335×10^{-11}	13(27)	668	668	$S = 1.8667 \times 10^3$ $W_{\mathrm{To}} = 1.9516 \times 10^5$	$l_f = 1.5000 \times 10^2$ $b = 1.4000 \times 10^2$ $T_i = 2.7750 \times 10^4$	$(L/D)_c = 21.3343$ $(L/D)_{\mathrm{To}} = 14.8474$ $(L/D)_L = 18.4172$	$V_{\mathrm{br}} = 614.09$ $R_{\mathrm{fr}} = 0.2955$ $U = 0.5004$
IDF	−2.0152	0.0	5	67	67	$S = 1.7425 \times 10^3$ $W_{\mathrm{To}} = 2.0581 \times 10^5$	$l_f = 1.5000 \times 10^2$ $b = 1.3526 \times 10^2$ $T_i = 4.2076 \times 10^4$	$(L/D)_c = 21.1413$ $(L/D)_{\mathrm{To}} = 13.5157$ $(L/D)_L = 16.9923$	$V_{\mathrm{br}} = 652.80$ $R_{\mathrm{fr}} = 0.2847$ $U = 0.4791$
AAO	−1.9629	-1.4627×10^{-4}	5	127	127	$S = 1.6832 \times 10^3$ $W_{\mathrm{To}} = 2.0641 \times 10^5$	$l_f = 1.4234 \times 10^2$ $b = 1.2637 \times 10^2$ $T_i = 4.1582 \times 10^4$	$(L/D)_c = 20.0482$ $(L/D)_{\mathrm{To}} = 11.9603$ $(L/D)_L = 15.2775$	$V_{\mathrm{br}} = 682.57$ $R_{\mathrm{fr}} = 0.2864$ $U = 0.4802$
CO	−2.0139	3.1050×10^{-2}	250*	156177	469726	$S = 1.5275 \times 10^3$ $W_{\mathrm{To}} = 1.8190 \times 10^5$	$l_f = 1.2238 \times 10^2$ $b = 1.2469 \times 10^1$ $T_i = 2.9520 \times 10^4$	$(L/D)_c = 20.3095$ $(L/D)_{\mathrm{To}} = 12.9401$ $(L/D)_L = 19.3783$	$V_{\mathrm{br}} = 658.07$ $R_{\mathrm{fr}} = 0.2793$ $U = 0.4992$
BLISS	−1.6035	6.8202×10^{-2}	7(7)	148	148	$S = 2.0752 \times 10^3$ $W_{\mathrm{To}} = 2.5000 \times 10^5$	$l_f = 1.050 \times 10^2$ $b = 8.5000 \times 10^1$ $T_i = 7.1244 \times 10^4$	$(L/D)_c = 12.6990$ $(L/D)_{\mathrm{To}} = 4.6612$ $(L/D)_L = 6.6104$	$V_{\mathrm{br}} = 905.43$ $R_{\mathrm{fr}} = 0.3243$ $U = 0.4843$
MDOIS	−1.9706	-6.9561×10^{-7}	8(8)	297	308	$S = 1.8496 \times 10^3$ $W_{\mathrm{To}} = 2.3080 \times 10^5$	$l_f = 1.3529 \times 10^2$ $b = 1.3936 \times 10^2$ $T_i = 5.6499 \times 10^4$	$(L/D)_c = 21.4148$ $(L/D)_{\mathrm{To}} = 12.9732$ $(L/D)_L = 16.3808$	$V_{\mathrm{br}} = 671.14$ $R_{\mathrm{fr}} = 0.2766$ $U = 0.4500$
Rule base (SQP)	−2.0682	0.0000	39	2529	1594	$S = 1.8670 \times 10^3$ $W_{\mathrm{To}} = 1.9504 \times 10^5$	$l_f = 1.3531 \times 10^2$ $b = 1.4000 \times 10^2$ $T_i = 2.7750 \times 10^4$	$(L/D)_c = 21.3516$ $(L/D)_{\mathrm{To}} = 14.8429$ $(L/D)_L = 18.4153$	$V_{\mathrm{br}} = 614.11$ $R_{\mathrm{fr}} = 0.2953$ $U = 0.5004$
Rule base (GA)	−2.0554	-5.4589×10^{-7}	25	144768	144768	$S = 1.8667 \times 10^3$ $W_{\mathrm{To}} = 1.9664 \times 10^5$	$l_f = 1.5000 \times 10^2$ $b = 1.4000 \times 10^2$ $T_i = 2.8044 \times 10^4$	$(L/D)_c = 21.0881$ $(L/D)_{\mathrm{To}} = 14.6893$ $(L/D)_L = 18.1991$	$V_{\mathrm{br}} = 616.43$ $R_{\mathrm{fr}} = 0.2971$ $U = 0.5005$

[a]For MDOIS the shared variables **Z** are optimized in the weights domain.

method does not converge robustly on the solution and proves considerably more expensive to execute. The rule base approach in this instance is using default convergence criteria that are too strict and consequently spends most of its time trying to refine the global optimum.

The results for the Yi3 problem are presented in Table 3. This problem is solved by all methods except CO and the rule base performs well in this instance. The global optimum value of the system objective function $f = 0.5$ is found exactly. However, it is noted that the problem does not have a unique global optimum and admits a number of solutions with the optimal system objective function value $f = 0.5$. The single optimizer methods all solve the problem using only two or three system level iterations. The bilevel methods need significantly more iterations for this problem. Again, Collaborative Optimization proves expensive and does not converge robustly. To converge on an accurate solution using the BLISS method it is necessary to use a trust region method that reduces the step sizes taken with the linear models. Consequently the BLISS method required 80 system level iterations to converge on the optimum solution and does not always do so, finding the optimum only once in the five random trials attempted here. The performance of the algorithms is broadly comparable with the performance figures reported in Yi et al. [10] with the slightly greater number of function calls required in our framework likely attributable to the apparently higher tolerances used by the optimizers.

A consistent optimum is not found for the LewisC subsonic aircraft design problem across the methods investigated. Table 4 summarizes the results for this problem. The rule based approach and the MDF method find the best results. Again, CO is extremely expensive and does not converge well with a manual termination required. BLISS also does not perform well for this problem and returns a result close to the bounds of many of the design variables. It is possible that the trust region algorithm could be improved here but it demonstrates that BLISS will often take the search to the bounds of the design variables.

The results for the subsonic business jet problem are presented in Table 5. The single level methods are all capable of finding the global optimum but are not guaranteed to do so. CO fails for this problem and BLISS performs well (although one should bear in mind that the method was initially published targeting this problem and performs well when the global optimum lies at the boundaries of the design space as this one does). MDOIS does not perform particularly well here with this most likely attributable to the difficulty in partitioning the shared design variables to the domains. It is not clear that there is a sensible partitioning for this problem. The rule base performs well on this problem.

The full TADPOLE problem comprises 11 shared design variables with over 40 state coupling variables. We find that the IDF and AAO methods are unable to converge a solution. For the IDF and AAO methods the design problem presented to the optimizer has in excess of 50 design variables (of which most are unconstrained) and a similar number of equality constraints. The SQP optimizer is not able to provide a satisfactory solution. Again, CO does not work well for this problem. The tightly coupled nature of this problem also makes the assignment of the shared design variables to domain level a difficulty. MDOIS is unable to find a feasible solution from the starting points tested. The rule base outperforms BLISS and is comparable with MDF. The results are summarised in Table 6.

VI. Discussion

As discussed in [10,23] the ability to reproduce results of applying MDO methods to problems from the literature can prove difficult. Implementations of MDO methods often use different optimizer software, start points may be different, configuration of the methods may be tuned to the problem being studied and there may be further subtle differences in the implementation. The results presented here are no exception in that the behavior of the methods is occasionally noticeably different to results reported by other authors for the same

Table 5 Results of the MDO methods applied to the problem labeled SBJ

Method	Objective	Maximum constraint ($g <= 0$)	Number of system iterations (system analysis calls)	Number of analysis calls				Shared design variables vector Z	Local design variables vector X	State variables vector Y	
				Str.	Aero.	Prop.	Rng.				
MDF	3963.77 nm	0.0	7(15)	1066	1066	1066	1066	$t/c = 0.06$ $h = 60000.0$ $M = 1.4$ $\mathrm{AR} = 2.5$ $\Lambda = 70.0$ $S_{\mathrm{ref}} = 1500.0$	$\lambda = 0.3800$ $x = 0.75$ $C_f = 0.75$ $T = 0.1562$	$W_T = 44760.0$ $W_F = 19350.6$ $\Theta = 0.9700$ $L = 44760.0$ $D = 5552.9$	$L/D = 8.0607$ $\mathrm{SFC} = 0.9239$ $W_E = 9435.6$ $\mathrm{ESF} = 0.7327$ $R = 3963.77$
IDF	3963.77 nm	0.0	97	2075	2075	2075	12075	$t/c = 0.06$ $h = 60000.0$ $M = 1.4$ $\mathrm{AR} = 2.5$ $\Lambda = 70.0$ $S_{\mathrm{ref}} = 1500.0$	$\lambda = 0.3800$ $x = 0.75$ $C_f = 0.75$ $T = 0.1562$	$W_T = 44760.0$ $W_F = 19350.6$ $\Theta = 0.9700$ $L = 44760.0$ $D = 5552.9$	$L/D = 8.0607$ $\mathrm{SFC} = 0.9239$ $W_E = 9435.6$ $\mathrm{ESF} = 0.7327$ $R = 3963.77$
AAO	3963.77 nm	0.0	54	2152	2152	2152	2152	$t/c = 0.06$ $h = 60000.0$ $M = 1.4$ $\mathrm{AR} = 2.5$ $\Lambda = 70.0$ $S_{\mathrm{ref}} = 1500.0$	$\lambda = 0.3800$ $x = 0.75$ $C_f = 0.75$ $T = 0.1562$	$W_T = 44760.0$ $W_F = 19350.6$ $\Theta = 0.9700$ $L = 44760.0$ $D = 5552.9$	$L/D = 8.0607$ $\mathrm{SFC} = 0.9239$ $W_E = 9435.6$ $\mathrm{ESF} = 0.7327$ $R = 3963.77$
CO	Fail										
BLISS	3963.77 nm	0.0	15(15)	1060	1060	1060	1060	$t/c = 0.06$ $h = 60000.0$ $M = 1.4$ $\mathrm{AR} = 2.5$ $\Lambda = 70.0$ $S_{\mathrm{ref}} = 1500.0$	$\lambda = 0.3800$ $x = 0.75$ $C_f = 0.75$ $T = 0.1562$	$W_T = 44760.0$ $W_F = 19350.6$ $\Theta = 0.9700$ $L = 44760.0$ $D = 5552.9$	$L/D = 8.0607$ $\mathrm{SFC} = 0.9239$ $W_E = 9435.6$ $\mathrm{ESF} = 0.7327$ $R = 3963.77$
MDOIS	2691.56 nm	9.6154×10^{-3}	10(10)	25725	766	763	748	$t/c = 0.06517$ $h = 60000.0$ $M = 1.4$ $\mathrm{AR} = 2.5$ $\Lambda = 55.0$ $S_{\mathrm{ref}} = 1500.0$	$\lambda = 0.1$ $x = 0.75$ $C_f = 0.75$ $T = 0.1562$	$W_T = 54835.8$ $W_F = 20845.6$ $\Theta = 0.9700$ $L = 54835.8$ $D = 8462.4$	$L/D = 6.4799$ $\mathrm{SFC} = 0.9240$ $W_E = 14690.2$ $\mathrm{ESF} = 1.1169$ $R = 2691.56$
Rule base (SQP)	4160.26 nm	1.7646×10^{-3}	31	217184	217184	217152	217152	$t/c = 0.0605$ $h = 59999.4$ $M = 1.4015$ $\mathrm{AR} = 2.5690$ $\Lambda = 69.9979$ $S_{\mathrm{ref}} = 1499.97$	$\lambda = 0.3528$ $x = 0.75$ $C_f = 0.7636$ $T = 0.1563$	$W_T = 45307.2$ $W_F = 19913.2$ $\Theta = 0.9700$ $L = 45358.0$ $D = 5594.5$	$L/D = 8.1077$ $\mathrm{SFC} = 0.9246$ $W_E = 9384.3$ $\mathrm{ESF} = 0.7289$ $R = 4160.26$
Rule base (GA)	3986.00 nm	-1.1316×10^{-5}	35	217184	217184	217152	217152	$t/c = 0.0602$ $h = 60000.0$ $M = 1.4009$ $\mathrm{AR} = 2.5905$ $\Lambda = 70.0$ $S_{\mathrm{ref}} = 1500.0$	$\lambda = 0.3760$ $x = 0.75$ $C_f = 0.75$ $T = 0.1564$	$W_T = 44817.2$ $W_F = 19415.9$ $\Theta = 0.9700$ $L = 44829.6$ $D = 5555.2$	$L/D = 8.0699$ $\mathrm{SFC} = 0.9246$ $W_E = 9430.5$ $\mathrm{ESF} = 0.7323$ $R = 3986.00$

Table 6 Results of the MDO methods applied to the problem labeled TADPOLE

Method	Objective	Maximum constraint ($g <= 0$)	Number of system iterations (system analysis calls)	Number of analysis calls			
				Drag	Lift	Weight	Cost
MDF	8.7439×10^{-2}	-1.0890×10^{-2}	21(45)	3194	3194	3194	3194
IDF	Fail						
AAO	Fail						
CO	Fail						
BLISS	1.0689×10^{-1}	-1.5408×10^{-2}	83(83)	5987	5987	5987	5987
MDOIS	Fail						
Rule base (SQP)	9.1842×10^{-2}	-6.2000×10^{-2}	100	10484	8066	7659	15437
Rule base (GA)	8.8634×10^{-2}	0.0000	37	217152	217152	217152	217152

[a]IDF and AAO equality constraints not satisfied. Appropriate assignment of shared design variables to domain in the MDOIS method is difficult to achieve due to tightly coupled nature of the problem.

methods on the same problems. Where this is the case possible reasons for the differences are highlighted.

The rule base used here is simple in construction and is only concerned with enforcing compromise upon the design domains, exchanging state information and responding to constraint violations if they occur. The rule base approach has the advantage that it is relatively straight forward to integrate into existing organizational infrastructure. However, further work is required to assess whether the pragmatic rule base approach, that sacrifices formal guarantees of convergence, will be truly competitive across a large range of MDO problem. The relative ease with which a rule based system level control process can be implemented and managed is a significant advantage over methods like BLISS, which can prove difficult to set up. Both BLISS and CO require the domain experts to optimize constructs of the process rather than directly investigate the physics of the underlying problem. Arguably, the BLISS system gradients can be difficult to evaluate and the domain design task can be changed from a physical problem to one of a line-search subject to local constraints. On a practical level, this risks losing the physical insight that the domain experts can bring to the design process and the physics of the system.

CO exhibits poor performance across the problem suite. There are alternative variations of CO in which the equality constraints evaluate the targets using different metrics but all are subject to some inherent weaknesses [24,25]. For the variant of CO employed here the satisfaction of the l_2 norm in the system level constraints causes divergence in the gradients at the optimum. This hinders the convergence of the method. As discussed in [25] the problem is also nonsmooth and the Jacobian of the master problem constraints is singular at the global optimum. There may also be multiple solutions to the domain level problem causing difficulties for the system level optimizer. Alexandrov and Lewis [24] demonstrates that for some problems the method can fail even when started close to an optimum solution. Alternative variations of the method address some of these issues (see for example ECO [26,27], MCO [25]). MDOIS, as expected, does not perform well on systems where there are many shared design variables.

The rule base shows promise and a key consideration in improving it will be the improvement of the data analysis of both the domain optimizer behaviors and the progression of the state variables. In particular the rule base will require refinement in its consideration of objective function value and analysis of behavior under changes in the state coupling variables. Further data analysis will be introduced in areas such as numerical optimization and management process control (detecting stalls, convergence, constraint revision/fidelity).

Further development of the rule base, informed by appropriate data analysis, will be guided with the following ideals in mind:

1) A single rule set is unlikely to be appropriate throughout the entire design life cycle. There is likely to be the need for multiple rule sets from which the design engineer or manager can select the particular set that should be active at any given time.

2) The rule base will need to report its reasoning and provide a view on the overall design process to keep the design team informed.

3) Domain analysis will involve multifidelity models which the rule base will need to account for. The rule base could potentially request analysis at different levels of fidelity to help inform its reasoning or as a result of previously formed rules. Finally, it is worth reiterating that by adopting a rule based approach it is very hard to carry out any formal convergence analysis of the resulting system: all that can be said is that the authors experimented with a series of rules inspired by previous pattern searches such as the well-known Hooke and Jeeves search and that there has been some progress over the years on formal proofs for pattern searches [28,29]. The aim here has rather been to provide an intuitive mechanism for control of domains that might appeal to those without a specialist MDO background.

VII. Conclusions

The design of complex integrated systems in engineering requires a multidisciplinary design optimization strategy. There is no current consensus on a single method applicable to all multidisciplinary problems. As discussed by Perez et al. [16] the standard algorithms from the literature vary in simplicity, transparency, portability, computational efficiency, and accuracy. Monolithic, single optimizer-based formulations are easy to architect but can suffer from additional complexity and attendant problems with integrating domain analysis tools into a single computing platform. Hierarchical methods can prove more complicated to implement but do provide disciplinary autonomy and the opportunity for concurrent analysis. The multilevel optimization formulations rely on an optimizer at the system level to coordinate the activities of the domain optimizers. In this work we investigate the viability of a "blackboard-based" approach to MDO. In such a system, each domain acts autonomously and investigates the local design asynchronously. All domains post their results and retrieve the state coupling data they require from a shared central repository. An ES has been employed at the system level to apply a rule base across the information stored in the blackboard to manage the system level design process.

Initial studies have involved a number of MDO problems ranging from simple numerical schemes, through basic aircraft sizing studies to a cutdown commercial in-house design tool. The rule base works by managing the bounds of the shared design variable vector until the enclosed hyper-volume converges to a specified tolerance and all domains are feasible. The performance of the rule base is found to be good across a range of problems, and it provides an overtly intelligible approach to multidisciplinary optimization and complements typical existing organizational structure in the enterprise. Future work will extend the framework to include data analysis of domain optimizers for improved feature recognition and development of a more sophisticated rule base to improve performance across the range of problems assembled.

Acknowledgment

This work was funded by Airbus UK, whose support is gratefully acknowledged.

References

[1] Sobieszczanski–Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments,"

Structural and Multidisciplinary Optimization, Vol. 14, No. 1, 1997, pp. 1–23.
doi:10.1007/BF01197554

[2] Keane, A. J., and Nair, P. B., *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, Wiley, New York, 2005, p. 616.

[3] Jain, A. K., Duin, R. P. W., and Mao, J., "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, 2000, pp. 4–37.
doi:10.1109/34.824819

[4] Cramer, E. J., Dennis, J. E., Frank, Paul D., Lewis, R. M., and Shubin, G. R., "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.
doi:10.1137/0804044

[5] Braun, R., *Collaborative Optimization: An Architecture for Large-Scale Distributed Design*, Stanford Univ., Palo Alto, CA, 1996.

[6] Sobieszczanski–Sobieski, J., Agte, J. S., and Sandusky, R. R., "Bilevel Integrated System Synthesis," *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 164–172.
doi:10.2514/2.937

[7] Sobieszczanski–Sobieski, J., Barthelemy, J.-F., and Riley, K. M., "Sensitivity of Optimum Solutions of Problem Parameters," *AIAA Journal*, Vol. 20, No. 9, 1982, pp. 1291–1299.
doi:10.2514/3.51191

[8] Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust Region Framework for Managing the use of Approximation Models in Optimization," Vol. 15, No. 1, 1997, pp. 16–23.

[9] Shin, M.-K., and Park, G.-J., "Multidisciplinary Design Optimization Based on Independent Subspaces," *International Journal for Numerical Methods in Engineering*, Vol. 64, No. 5, 2005, pp. 599–617.
doi:10.1002/nme.1380

[10] Yi, S., Shin, J., and Park, G., "Comparison of MDO Methods with Mathematical Examples," *Structural and Multidisciplinary Optimization*, Vol. 35, No. 5, 2007, pp. 391–402.
doi:10.1007/s00158-007-0150-2

[11] Lewis, K., *An Algorithm for Integrated Subsystem Embodiment and System Synthesis, in Mechanical Engineering*, Georgia Inst. of Technology, Atlanta, 1997, p. 532.

[12] Zhang, K. S., Han, Z. H, Li, W. J., and Song, W. P., "Bilevel Adaptive Weighted Sum Method for Multidisciplinary Multi-Objective Optimization," *AIAA Journal*, Vol. 46, No. 10, 2008, pp. 2611–2622.
doi:10.2514/1.36853

[13] Sobieszczanski–Sobieski, J., Agte, J. S., and Sandusky, R. R., *Bi-Level Integrated System Synthesis (BLISS)*, NASA Langley Research Center, NASA, Hampton, VA, 1998.

[14] Wang, D., Wang, G. G., and Naterer, G. F., "Extended Collaboration Pursuing Method for Solving Larger Multidisciplinary Design Optimization Problems," *AIAA Journal*, Vol. 45, No. 6, 2007, pp. 1208–1221.
doi:10.2514/1.21167

[15] Masmoudi, M., and Parte, Y. S., "Disciplinary Interaction Variable Elimination (DIVE) Approach for MDO," *European Conference on Computational Fluid Dynamics ECCOMAS CFD*, ECCOMAS, Barcelona, 2006.

[16] Perez, R. E., Liu, H. H. T., and Behdinan, K., "Evaluation of Multidisciplinary Optimization Approaches for Aircraft Conceptual Design," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2004-4537, 2004.

[17] Arroyo, S. F., Cramer, E. J., Dennis, J. E., Frank, P. D., "Comparing Problem Formulations for Coupled Sets of Components," *Optimization and Engineering*, Vol. 10, No. 4, 2009, pp. 557–573.

[18] Keane, A. J., and Scanlan, J. P., "Design Search and Optimization in Aerospace Engineering," *Philosophical Transactions of the Royal Society Series A: Physical Sciences and Engineering*, Vol. 365, No. 1859, 2007, pp. 2501–2529.
doi:10.1098/rsta.2007.2019

[19] Cousin, J., and Metcalfe, M., "The BAe Ltd. Transport Aircraft Synthesis and Optimisation Program," *AIAA/AHS/ASEE Aircraft Design, Systems and Operations Conference*, AIAA Paper 90-3295, 1990.

[20] Scanlan, J., Rao, A., Bru, C., Hale, P., and Marsh, R., "The DATUM Project: A Cost Estimating Environment for the Support of Aerospace Design Decision Making," *Journal of Aircraft*, Vol. 43, No. 4, 2006, pp. 1022–1029.
doi:10.2514/1.17362

[21] "Vanguard Studio. 2007," Vanguard Software Corp., Cary, NC.

[22] Kaufmann, M., Zenkert, D., and Wennhage, P., "Integrated Cost/Weight Optimization of Aircraft Structures," *Structural and Multidisciplinary Optimization*, Vol. 41, No. 2, 2010, pp. 325–334.
doi:10.1007/s00158-009-0413-1

[23] Martins, J. R. R. A., and Marriage, C., "An Object-Oriented Framework for Multidisciplinary Design Optimization," *Third AIAA Multidisciplinary Optimization Specialist Conference*, AIAA Paper 2007-1906, 2007.

[24] Alexandrov, N. M., and Lewis, R. M., "Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design," *AIAA Journal*, Vol. 40, No. 2, 2002, pp. 301–309.
doi:10.2514/2.1646

[25] DeMiguel, A.-V., and Murray, W., "An Analysis of Collaborative Optimization Methods," *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 2000-4720, 2000.

[26] Roth, B. D., and Kroo, I. M., "Enhanced Collaborative Optimization: A Decomposition-Based Method for Multidisciplinary Design," *International Design Engineering Technical Conferences & Computers and Information in Engineering IDETC/CIE*, ASME, Fairfield, NJ, 2008.

[27] Roth, B. D., and Kroo, I. M., "Enhanced Collaborative Optimization: Application to an Analytic Test Problem and Aircraft Design," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2008-5841, 2008.

[28] Nocedal, J., "Theory of Algorithms for Unconstrained Optimization," *Acta Numerica*, Vol. 1, 2008, pp. 199–242.
doi:10.1017/S0962492900002270

[29] Powell, M. J. D., "Direct Search Algorithms for Optimization Calculations," *Acta Numerica*, Vol. 7, 2008, pp. 287–336.
doi:10.1017/S0962492900002841

T. Zang
Associate Editor