AIAA JOURNAL
Vol. 41, No. 10, October 2003

Bilevel Integrated System Synthesis for Concurrent and Distributed Processing

Jaroslaw Sobieszczanski-Sobieski*
NASA Langley Research Center, Hampton, Virginia 23681
and
Troy D. Altus,† Matthew Phillips,† and Robert Sandusky‡
George Washington University and NASA Langley Research Center, Hampton, Virginia 23681

A new version is introduced of the bilevel integrated system synthesis method intended for optimization of engineering systems conducted by distributed specialty groups working concurrently in a multiprocessor computing environment. The method decomposes the overall optimization task into subtasks associated with disciplines or subsystems, where the local design variables are numerous and a single, system-level optimization whose design variables are relatively few. The subtasks are fully autonomous as to their inner operations and decision making. Their purpose is to eliminate the local design variables and generate a wide spectrum of feasible designs whose behavior is represented by response surfaces to be accessed by a system-level optimization. It is shown that, if the problem is convex, the solution of the decomposed problem is the same as that obtained without decomposition. A simplified example of an aircraft design shows the method working as intended. A discussion of the method merits and demerits as well as recommendations for further research is included.

Nomenclature

AR = aspect ratio
c = coupling equality constraints
g = behavior constraints local to a black box (BB) (another term for a module)
h = equality constraints tantamount to solution of analysis
h = cruise altitude (in Table 1 only)
L/D = lift to drag ratio
Q = system level design variables, $\{\boldsymbol{X}_{\text{sh}}|\boldsymbol{Y}^*|w\}$, a subset of $\boldsymbol{Z}$
R = flight range
S_{ref} = wing reference area
T = throttle setting
t/c = airfoil depth
ts = cross-sectional dimensions
$\boldsymbol{U}$ = local design variables, $\{\boldsymbol{X}_{\text{loc}}|\boldsymbol{Y}\hat{}\}$, a subset of $\boldsymbol{Z}$
U, L = upper and lower bounds
L_{ht} = horizontal tail location coordinate
M = Mach number
w = weighting factor in suboptimization objective function
W = weight
W_T = total weight, takeoff gross weight (TOGW)
$\boldsymbol{X}_{\text{loc}}$ = design variables local to a BB
$\boldsymbol{X}_{\text{sh}}$ = shared design variables affecting directly two or more BBs (modules)
$\boldsymbol{Y}^*$ = behavior variables input into a BB from other BBs
$\boldsymbol{Y}\hat{}$ = behavior variables output from a BB, some elements of $\boldsymbol{Y}\hat{}$ designated to be $\boldsymbol{Y}^*$
$Y_s\hat{}$ = particular data item selected in a particular BB output to be the system objective
$\boldsymbol{Z}$ = $\{\boldsymbol{X}_{\text{sh}}|\boldsymbol{X}_{\text{loc}}|\boldsymbol{Y}^*|\boldsymbol{Y}\hat{}\}$ a vector of variables in a not decomposed, combined analysis and optimization problem (Examples in aircraft design include wing aspect ratio and sweep angle $\boldsymbol{X}_{\text{sh}}$, wing cover panel thickness and a composite ply orientation angle $\boldsymbol{X}_{\text{loc}}$, elastic deformation that alters the wing aerodynamic shape $\boldsymbol{Y}^*$ and $\boldsymbol{Y}\hat{}$, and aerodynamic loads that cause the wing deformation.)
Λ = sweep angle
λ = taper ratio
Θ = effective wing area change due to twist

Subscripts

E = engine
F = fuel
HT = horizontal tail
s = system objective
T = TOGW
w = wing
0 = optimal

Superscripts

a = approximate
$\hat{}$ = output from a BB
$*$ = input to a BB from another BB or generated by the system optimizer

Received 4 September 2002; revision received 2 March 2003; accepted for publication 17 March 2003.

*Senior Research Scientist, Analytical and Computational Methods Branch, Mail Stop 240, Structures and Materials Competency. AIAA Fellow.
†Research Assistant, Mail Stop 335. Member AIAA.
‡Professor, Mail Stop 335. Fellow AIAA.

I. Introduction

THE relentless drive of computer technology toward ever higher computing speed (floating point operations per second) has enabled solution of large computational problems encountered in design of complex engineering systems, for example, an aircraft, in a small fraction of time that was required just a few years ago. Nevertheless, it is still not practical to solve such problems in one monolithic calculation because the number of design variables is large, the analysis is high fidelity, and it consists of a system of coupled codes. Such a monolithic calculation, perhaps, may never become a routine practice regardless of the computing speed available because design of complex engineering systems requires that all sorts of information ranging from experimental to computational

be synthesized by human judgment. Consequently, it is a common practice to conduct a design process by collaboration of autonomous groups of specialists that retain control over their domains of expertise and that work concurrently to compress the project elapsed time.

Motivated by this, several methods have emerged to enable decomposition of the system optimization into a set of smaller tasks aligned with disciplines or physical subsystems. Optimization by decomposition rests on rich literature whose roots may be traced to Ref. 1, inspired by problems in large organization management, and Ref. 2, concerned with large linear programming applications. The early contributions that followed included Mesarovic et al.[3] addressing the control of hierarchical systems and development of more general mathematical foundations for nonhierarchical systems in Refs. 4 and 5. Based on the preceding references, applications to structures was initiated in Ref. 6 and has been further developed to include other engineering disciplines that couple with structural mechanics in design of flight vehicles and other engineering systems. These developments coalesced in multidisciplinary design optimization[7–9] and resulted in a body of literature much too large to be cited here in full. A few representative references are Refs. 10–22. (References 21 and 22 are surveys.) The decomposition methods emerging from these developments share as common threads the use of suboptimizations at a lower level, all coordinated by solution of a higher, system-level optimization, and a natural ability to engage large numbers of concurrently operating processors in the technology of massively concurrent distributed processing (MCDP) that is now developing a computational infrastructure to support engineering design at an unprecedented level of effectiveness. Their diversity lies in the means that link the two levels, in the use of various approximations, and in the definition of the objective function(s) at both levels. As to the linking, the typical approaches are the sensitivity of optimum to parameters, the system behavior sensitivity, control over the contributions a suboptimization makes to the satisfaction of constraints in another suboptimization, and imposition of targets on the suboptimization results.

The approximations commonly used in constructing surrogate linear or nonlinear models are often based on the disciplinary and system behavior derivatives. Recently, domain-spanning approximations such as the response surfaces (RS) or neural nets (NN) constructed with the aid of design of experiments (DOE) techniques became popular because they lead to repetitive but independent calculations that can be performed simultaneously to exploit the MCDP technology, for example, Refs. 13, 16, and 23.

The choice of the objective function(s) is fundamentally important. It is also difficult because in a suboptimization the objective ought to reflect local information combined with the influence of that information on the system performance and vice versa at the system level, while preserving the suboptimization autonomy.

A structural design of the wing is an example that illustrates considerations that underlie the choice of the objective in a subsystem optimization. Its output includes structural weight and elastic deformation. Conventionally, the weight would be the objective. However, the ultimate objective is a measure of the aircraft system performance. That performance, in general, benefits from reduction of weight and drag. Because increased structural stiffness lowers the elastic wing drag, it follows that, for the system benefit, the wing structure ought to be designed for an optimal combination of low weight and high stiffness. (The common transport aircraft design practice of using a "jig shape" to reduce the deformable wing drag was deliberately discarded in this study because there is only one jig shape available to compensate for the wing deformation at one point in the mission profile, usually the midcruise point. That single-point compensation may be inadequate in design of a supersonic transport whose mission may include long subsonic segments. Hence, for the sake of generality, the presentation retains a two-way coupling between aerodynamics and structures.) Furthermore, the wing structural weight may be lightened by tailoring the wing deformation to redistribute the aerodynamic load to reduce the root bending moment. The resulting load distribution is, in general, different from the one that minimizes the drag.

The original version of bilevel integrated system synthesis (BLISS)[15,24,25] addressed the enunciated considerations by computing the system sensitivity derivatives of the system objective function with respect to the subsystem design variables and, then, defining a subsystem objective function as a sum of the subsystem design variables weighted by these derivatives. The subsequent versions included the use of RS[23] and a variant specialized for structural optimization.[26]

The key new concept in BLISS 2000 reported herein is a new formulation for the objective function in the subsystem optimization and the use of that optimization to control the subsystem output for the benefit of the system performance. In the new formulation, the subsystem optimization objective is a sum of the subsystem outputs, each output weighted by a coefficient that is treated as a design variable in the system-level optimization. This formulation eliminates the costly system sensitivity and optimum sensitivity analyses. It also enables representation of the subsystem optima at the system-level by the RS prepared "offline" by autonomous, concurrent, and potentially distributed operations. It is expected that the specialty groups in charge of these operations work in a MCDP environment, being free to draw on any sources of information and to apply any tools they choose.

The remainder of the paper defines the BLISS 2000 algorithm first. It follows with a discussion and interpretation of the algorithm in qualitative terms and offers a theoretical justification of the approach by demonstrating that the solution obtained by decomposition solves the same problem, at least when it is convex, formulated without decomposition. Remarks about the computational cost conclude the method description.

A simplified, conceptual-level design problem of a supersonic business jet illustrates the method and contributes an empirical basis to a concluding assessment of the method limitations, advantages, and recommended future developments.

II. BLISS 2000 Algorithm

The object of optimization in BLISS is a generic engineering system whose representative example is an aircraft (Fig. 1). The diagram in Fig. 2 shows a few major disciplines that form a coupled system in a generic aircraft mathematical model. The model comprises modules, also called black boxes (BBs), coupled by data exchanges.

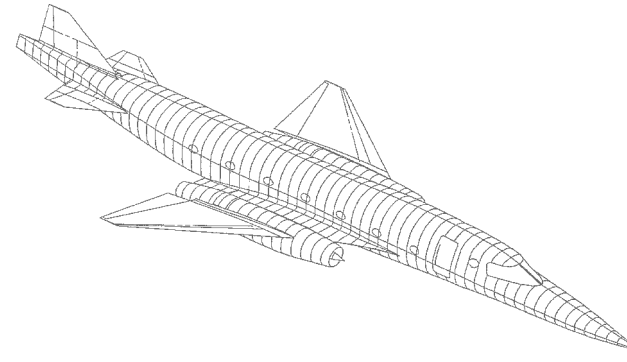

Fig. 1 Numerical example: generic Supersonic Business Jet.

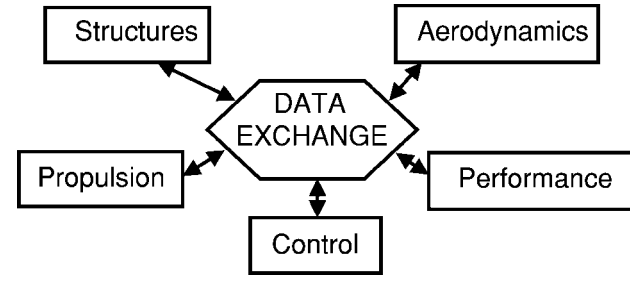

Fig. 2 Example of a few principal disciplines coupled in aircraft treated as a system.

A. Original Problem

For any system optimization-by-decomposition scheme to be valid, it should address the same system optimization problem formulated without decomposition. Therefore, introduction of the BLISS algorithm begins with an all-in-one (also known as all-at-once) formulation that combines analysis and optimization of a generic system of which Fig. 2 is an example. It reads as follows.

Find:

$$\boldsymbol{Z} \tag{1}$$

Minimize:

$$F(\boldsymbol{Z}) = Y_s\hat{}(\boldsymbol{Z}) \tag{1a}$$

Satisfy:

$$g(\boldsymbol{Z}) <= 0, \qquad \text{for each BB} \tag{1b}$$

$$h(\boldsymbol{Z}) = 0, \qquad \text{for each BB} \tag{1c}$$

$$c(\boldsymbol{Z}) = \boldsymbol{Y}^* - \boldsymbol{Y}\hat{} = 0 \tag{1d}$$

$$\boldsymbol{Z}L <= \boldsymbol{Z} <= \boldsymbol{Z}\boldsymbol{U}, \qquad \text{side constraints} \tag{1e}$$

Output:

$$Z_0, \qquad F_0 \tag{1f}$$

where the inequalities g represent the behavior constraints local to a BB, for example, structural strength and limit on the aerodynamic pressure gradient along the wing airfoil chord, and the equalities h correspond to the solution of the BB governing equations (the BB inner analysis, for example, equations of the finite element analysis). The output–input equalities c describe the system intermodular couplings, for example, the aerodynamic loads that deform a wing should be computed for the wing deformed due to those loads. In Eq. (1) they are present for two reasons:

1) The input $\boldsymbol{Y}^*$ to a particular BB is supplied by the optimizer instead of being received directly as $\boldsymbol{Y}\hat{}$ from another BB, where it would originate in a conventional analysis of a coupled system. Therefore, the equalities c are needed to ensure that upon convergence the optimizer-generated $\boldsymbol{Y}^*$ is the same as the corresponding $\boldsymbol{Y}\hat{}$.

2) The system analysis needs to be solved. Equating the corresponding $\boldsymbol{Y}^*$ and $\boldsymbol{Y}\hat{}$, in pairs, is tantamount to obtaining a solution.

The formalism of Eq. (1) is known as the simultaneous analysis and design (SAND). It brings the h and c equalities under a single optimization algorithm in contrast to the more common optimization techniques that eliminate $\boldsymbol{Y}\hat{}$ and $\boldsymbol{Y}^*$ from $\boldsymbol{Z}$ by solving the h an c equalities for each set of the trial values of $\boldsymbol{X}_{\text{sh}}$ and $\boldsymbol{X}_{\text{loc}}$ sent out by the optimizer.

Whereas Eq. (1) constitutes a starting point for deriving the BLISS algorithm and its theoretical justification, its numerical solution is not practical as a tool for optimization of systems of any significant complexity for a number of reasons. The primary reason is the need to exploit existing (legacy) codes in the BBs. Such codes are notorious for producing noisy outputs that render useless the gradients needed to guide the search algorithm. The remedies of automated differentiation or recoding to build-in an analytical sensitivity analysis are expensive to implement and would impede the capability of accommodating existing codes, the chief advantage of BLISS. The other remedy of using gradientless search, for example, a genetic algorithm, would be prohibitively costly due to the large number of system analyses required even in a relatively small problem (such as the numerical example herein), let alone for the number of variables typical for industrial-strength cases. Finally, for the SAND results to be a valid benchmark, one would have to construct an artificial, provably convex test case. In absence of proven convexity, the result depends on the choice of the solution algorithm and its details and on the particular initialization. Hence, the global optimum cannot be guaranteed, and the focus of the method, then, shifts to demonstration of an ability to improve the initial design.

B. BLISS 2000 Optimization by Decomposition

The BLISS 2000 algorithm separates the BB suboptimizations from the system optimization. To introduce the algorithm, this section provides a formal statement for the optimization at the BB level, followed by definition of the RS that link the preceeding optimization to the system-level optimization, and concludes with a formal statement for the latter. The section that follows elaborates on a rationale for the two-level procedure, its theoretical underpinnings, and the computational cost considerations.

The design variables $\boldsymbol{Z}$ are divided into the system level variables $\{\boldsymbol{X}_{\text{sh}}|\boldsymbol{Y}^*\}$ and the subsystem level variables $\{\boldsymbol{X}_{\text{loc}}|\boldsymbol{Y}\hat{}\}$ and the following optimization problem is solved for each BB.

Given:

$$Q = \{\boldsymbol{X}_{\text{sh}}|\boldsymbol{Y}^*|w\} \tag{2}$$

Find:

$$\boldsymbol{U} = \{\boldsymbol{X}_{\text{loc}}|\boldsymbol{Y}\hat{}\} \tag{2a}$$

Minimize:

$$f(\boldsymbol{U}) = \sum w_i Y_i\hat{} \tag{2b}$$

Satisfy:

$$g(\boldsymbol{U}) <= 0, \qquad \text{for each BB} \tag{2c}$$

$$h(\boldsymbol{U}) = 0, \qquad \text{for each BB} \tag{2d}$$

$$\boldsymbol{U}L <= \boldsymbol{U} <= \boldsymbol{U}\boldsymbol{U} \tag{2e}$$

Output:

$$Y_0\hat{}, \qquad \boldsymbol{X}_{\text{loc}_0} \tag{2f}$$

The weighting coefficients w_i in f, one coefficient per element of the output vector $\boldsymbol{Y}\hat{}$, are appended to the system-level variables to link the BB suboptimization to the system optimization. The w coefficients are allowed to vary within adjustable intervals $\boldsymbol{U}L$ and $\boldsymbol{U}\boldsymbol{U}$ that include both positive and negative values. The negative value of w_i signifies maximization with respect to the corresponding $Y_i\hat{}$. In Ref. 27, it is shown that for the weighted objective function to be effective in this application, one should allow the w range to include both positive and negative values. For reasons to be explained later, the constrained minimum of f is not passed to the system optimization; only the $Y_0\hat{}$ values are so transferred.

The Eq. (2) optimization is executed at a number of points dispersed in the Q space that belongs to the BB. The formal DOE techniques may aid in forming the dispersal pattern to achieve a reasonable coverage of the domain. From the system perspective the solution method choice is immaterial and that choice need not be the same for all of the BBs. The specialists in charge of a BB are free to use any suitable method, even experiments, or a guess.

C. Approximate Model for Optimized Subsystems

Next, an RS is fitted (or a neural net may be trained) to each of the elements of $Y_0\hat{}$. When each RS is regarded as a leaf in a sheaf, the resulting database is a sheaf of RS (SRS). It constitutes an approximate model in the Q space of the BB optimized using the $\boldsymbol{U}$ coordinates, so that

$$Y_0\hat{}^a = Y_0\hat{}^a[\text{SRS}(Q)], \qquad \boldsymbol{U}_0^a = \boldsymbol{U}_0^a[\text{SRS}(Q)]$$

$$\text{for} \quad QL <= Q <= Q\boldsymbol{U} \tag{3}$$

In Eq.(3), the superscript a denotes approximate values and the notation such as $Y_0\hat{}^a = Y_0\hat{}^a$ [SRS(Q)] means that the approximate values are retrieved from precomputed SRS. The bounds on Q, QL, and $Q\boldsymbol{U}$, are the best estimates, accounting for any side constraints and, also, incorporating the move limits necessary for the ensuing interlevel iteration. The SRS may be called a domain approximation because it covers the entire Q space within the given bounds.

In principle, a similar RS approximation could be constructed for each element of $\boldsymbol{X}_{\mathrm{loc}\,0}$. However, the resulting volume of data to be stored may be so large as to make this impractical. If so, $\boldsymbol{X}_{\mathrm{loc}\,0}$ can be regenerated in a manner discussed in Sec. II.E.

The computational cost of an RS is proportional to the number of points to be evaluated, which grows with the number of variables. It is desirable, therefore, to reduce the number of variables in Q by a condensation technique. Such condensation is imperative for those variables in Q that represent field quantities, for example, the field of the pressure loads distributed over the wing and the corresponding field of displacements. A condensation of a field data may be accomplished by defining the field variable $P(\{v\})$ in the space of $\{v\}$ as

$$P(v) = P^{a}[p_i f_i(v)] \tag{4}$$

where the number of the parameters p_i and the basis functions $f_i(\{v\})$ is made as small as possible, for example, displacements P normal to a wing may be represented by a function in the chordwise and spanwise coordinates $\{v\}$. Let the function f_i be the product of polynomials whose coefficients are the parameters p. Assuming the chordwise polynomial to be linear and the spanwise one to be cubic reduces the number of parameters p to only 5. The p parameters enter the procedures as if they were $\boldsymbol{Y}^{*}$ and $\boldsymbol{Y}\hat{}$, each p being represented by its own RS.

When all the BBs are optimized, the functions in Eq. (3) for each BB are available to the system optimization that executes next.

D. BLISS System Optimization

The system-level optimization problem is as follows.

Given:

$$\text{a set of SRS, one for each BB} \tag{5}$$

Find:

$$\{Q\} = \{\boldsymbol{X}_{\mathrm{sh}} | \boldsymbol{Y}^{*} | w\} \tag{5a}$$

Minimize:

$$F(Q) = Y\hat{}^{a}_{0s} \tag{5b}$$

Satisfy:

$$c = \boldsymbol{Y}^{*} - Y\hat{}_{0}(Q) = 0 \tag{5c}$$

$$QL <= Q <= QU \tag{5d}$$

Output:

$$Q_0, \qquad F_0 \tag{5e}$$

Equation (5) may be solved by any optimization technique, the data for $F(Q) = Y\hat{}^{a}_{0s}$ and $c(Q)$ for any Q being retrieved, as per Eq. (3), from the SRS database. The efficiency of the technique is not critical because that data retrieval is nearly instantaneous. As in Eq. (1), $c = 0$ in Eq. (5) is tantamount to a system analysis; thus, the BLISS algorithm implements, in effect, the SAND method at the system level, but not necessarily at the BB level, where the choice of methods for solving Eq. (2) is autonomous.

E. BLISS Iterative Procedure

The retrieval of data from SRS [Eq. (3)] in a nonlinear system is burdened with an error, $\varepsilon = Y\hat{}_{0s} - Y\hat{}^{a}_{0s}$, whose control requires iteration between Eqs. (2) and (5) involving move limits incorporated in QL and QU in Eq. (5d). These move limits are being adjusted in each iteration, and occasionally, they may push against the SRS bounds. That may require the addition of new points at which to solve Eq. (2) and refitting of the SRS.

The iterative procedure may also include at its conclusion a retrieval of the optimal $\boldsymbol{X}_{\mathrm{loc}}$ whose elements might have been stored in form of SRS. If the volume of such storage is prohibitively large, one may choose to regenerate the $\boldsymbol{X}_{\mathrm{loc}}$ instead by repeating Eq. (2) while substituting the latest optimal $\{\boldsymbol{X}_{\mathrm{sh}} | \boldsymbol{Y}^{*} | \boldsymbol{Y}\hat{}\,| w\}$. However, such regeneration is equivalent to a partial execution of one additional iteration of the entire BLISS procedure; therefore, some of the consistency between Eqs. (2) and (5) is lost, resulting in an additional error.

The overall, step-by-step recipe for BLISS 2000 optimization algorithm is now written:

0) Start.

1) Initialize all variables, system and local: $\boldsymbol{X}_{\mathrm{sh}}$, $\boldsymbol{X}_{\mathrm{loc}}$, $\boldsymbol{Y}^{*}$, and w, and their $\boldsymbol{U}$ and L bounds.

2) Initial system analysis by solving the equations a) $h = 0$, in all BBs, that is, perform analyses in the individual BBs, to obtain $\boldsymbol{Y}\hat{}$, b) $c = 0$, to obtain new $\boldsymbol{Y}^{*}$, and c) iterate parts a and b until convergence. This step improves the starting point for the remainder of the procedure by facilitating establishment of the RS bounds and $\boldsymbol{Y}^{*}$ values. It is optional.

3) Approximate model development for each BB. This step may be done simultaneously for all of the BBs.

a) Reduce dimensionality of the Q space per Eq. (4).

b) Disperse by a suitable DOE technique a minimum number of points required to define an RS approximation in the BB Q space, bounded by QL and QU.

c) Solve Eq. (2) in subspace $\boldsymbol{U}$ at the preceding points in space Q. This may be done simultaneously for all the points within Q.

d) Fit an SRS to the results of steps 3b and 3c.

e) Verify quality SRS by random sampling. If needed, add new points and discard old points, and use least-square fit (or an equivalent technique) to improve the SRS quality.

f) After each system optimization (step 4), shift, extend, or shrink the intervals QL and QU to avoid excursions beyond the SRS bounds and to maintain the approximation quality.

4) Solve system optimization Eq. (5) in space Q, accessing the SRS data per Eq. (3).

5) Check the termination criteria: Exit, or repeat from step 3 using SRS already available or updated per steps 3e and 3f.

6) Retrieve the optimal $\boldsymbol{X}_{\mathrm{loc}}$.

7) Stop.

One execution of the procedure from step 3 to step 5 is called the major iteration or the cycle.

III. BLISS Rationale, Theoretical Justification, and Computational Effort

With the BLISS 2000 algorithm defined, attention now turns to its underlying rationale, theoretical justification, and the computational cost of the method.

A. Qualitative Discussion of Rationale

The system objective F is computed as an element of $\boldsymbol{Y}\hat{}$ in one of the BBs. The $\boldsymbol{Y}\hat{}$ from any BB influences F because of the couplings c. However, one does not initially know the strength or even the sign of these influences. The suboptimization task for a BB is, then, to develop a wide choice of the BB designs, each having a different set of outputs $\boldsymbol{Y}\hat{}$ and each being feasible with respect to the local constraints h and g.

That task is accomplished by using the BB optimization in Eq. (2) merely as a tool to control the BB output $\boldsymbol{Y}\hat{}$, the weighting coefficients w in Eq. (2b) acting as parameters of that control. Replacement of the set of the original vector of the system-level design variables $\{\boldsymbol{X}_{\mathrm{sh}} | \boldsymbol{Y}^{*}\}$ with the vector $\{\boldsymbol{X}_{\mathrm{sh}} | \boldsymbol{Y}^{*} | w\}$, that includes w as additional design variables, results in a range of the BB designs $\{\boldsymbol{X}_{\mathrm{loc}} | \boldsymbol{Y}\hat{}\}$, at each point in the $\{\boldsymbol{X}_{\mathrm{sh}} | \boldsymbol{Y}^{*}\}$ space, all of the BB designs being feasible. Availability of a wide choice of these designs in form of SRS enables the system-level optimization in Eq. (5) to find a set of the BB designs that are compatible with respect to $c = 0$ and generate outputs $\boldsymbol{Y}\hat{}$ whose collective influence minimizes the objective equation (5b).

The described approach to BB suboptimization rests on the concept of optimizing a component to attain a desired response. Before explaining why that particular approach to the multiobjective optimization in Eq. (2) was chosen, one should acknowledge that a similar control of the outputs might be accomplished by other

means. For example, the collaborative optimization (CO) method, for example, Refs. 14 and 16, achieves that control by imposing targets Y^T on the output variables $\boldsymbol{Y}^{\wedge}$ and minimizing the discrepancy $f = \Sigma(Y^T - \boldsymbol{Y}^{\wedge})^2$ between Y^T and $\boldsymbol{Y}^{\wedge}$ as the objective function in Eq. (2) to generate the optimal $\boldsymbol{Y}^{\wedge}$. Unlike BLISS 2000, CO includes the $\boldsymbol{X}_{\text{sh}}$ inputs as design variables in $\boldsymbol{U}$ in Eq. (2) together with $\boldsymbol{X}_{\text{loc}}$, and imposes the targets on $\boldsymbol{X}_{\text{sh}}$ to generate optimal $\boldsymbol{X}_{\text{sh}}$ in addition to optimal $\boldsymbol{X}_{\text{loc}}$. The w variables are absent in the CO formulation because their role is assigned, in effect, to the target variables. In both the CO version reported in Ref. 16 and in BLISS 2000 herein, the optimal results are stored for the use in the system-level optimization in a form of RS.

The formulation with targets Y^T in place of w could also be an alternative in BLISS. It could be combined with stating the objective function in Eq. (2b) in a goal programing style[28] as

$$f = \Sigma(d^+ + d^-)$$

where d^+ and d^- measure the overachievement or underachievement of the targets.

Given a gamut of techniques for optimizing a component to attain a desired response, the selection of the simplest technique of the weighted sum as in Eq. (2b) (the composite objective function) stems from the rigorous geometrical argument given in Ref. 27 that shows it to be a good choice. It is so because, at least for convex problems, it always returns designs that are feasible, located on the constraint boundary or in the feasible space interior, and includes both Pareto and non Pareto design points. In Ref. 27 the method performance is illustrated with numerical examples, including a fairly large case of a builtup wing structure, and examines the method merits against the alternatives. Even though the evidence presented in Ref. 27 justifies selection of the composite objective function technique for solution of the BB optimization [Eq. (2)], in BLISS the BB optimizations are autonomous so that users may implement other techniques according to their preferences.

B. Theoretical Justification

The following chain reasoning shows that a solution of the set of Eqs. (2) and (5) satisfies the original problem described by Eq. (1):

1) Solution of Eq. (2) satisfies constraints $h = 0$ and $g <= 0$ present in Eq. (1).

2) Solution of Eq. (5) satisfies constraints $c = 0$ present in Eq. (1).

3) Solution of Eq. (2) and the subsequent generation of SRS [Eq. (3)] renders $\boldsymbol{Y}^{\wedge} = Y^{\wedge a}_0(\{\boldsymbol{X}_{\text{sh}}|\boldsymbol{Y}^*|w\})$ and $\boldsymbol{X}_{\text{loc}} = \boldsymbol{X}_{\text{loc}^a_0}$ $(\{\boldsymbol{X}_{\text{sh}}|\boldsymbol{Y}^*|w\})$. If the problem is convex, the preceding relations are unique for each vector element (single-valued functions).

4) Because of the preceding relations, the space of $(\boldsymbol{X}_{\text{sh}}|\boldsymbol{Y}^*|w\}$, searched in Eq. (5), maps uniquely (in convex problems) into the original space of $\{\boldsymbol{X}_{\text{sh}}|\boldsymbol{X}_{\text{loc}}|\boldsymbol{Y}^*|\boldsymbol{Y}^{\wedge}\}$, searched in Eq. (1), because $\boldsymbol{X}_{\text{sh}}$ and $\boldsymbol{Y}^*$ define both spaces, and $\boldsymbol{X}_{\text{loc}}$ and $\boldsymbol{Y}^{\wedge}$ are turned into functions of $\{\boldsymbol{X}_{\text{sh}}|\boldsymbol{Y}^*|w\}$ per reasoning 3.

5) The existence of the mapping of the space of Eq. (5) into the space of Eq. (1) together with satisfaction of the constraints g, h, and c assures that the solution of the sequence Eqs. (2), (3), and (5) also satisfies Eq. (1), subject to the caveats that follow.

1) If the problem of Eq. (1) has local minima (a nonconvex problem), then the solutions of Eqs. (2), (3), and (5) may not arrive at the solution of Eq. (1) even when starting from the same initial point because the algorithmic differences may result in different search path being traced. This is the reason why no universal search method has as yet been devised to be effective for all types of nonconvexities in their limitless variety.

2) If the SRS were error free and wide enough to contain the solution to Eq. (5), a single execution of the sequence Eqs. (2–5) would suffice. Otherwise, the sequence has to include the SRS updates and be iterated to reduce the error $\varepsilon = Y^{\wedge}_0 - Y^{\wedge a}_0$ and to extend the RS-covered space.

Comparison of the BLISS algorithm with the classic algorithm given in Ref. 2 reveals the former to be a generalization of the latter. The generalization includes nonlinear formulations at the BB and system levels, autonomous BBs, and the use of RS to decouple the two levels and to create opportunities for concurrent processing.

C. Computational Effort

In BLISS 2000, the SRS data may be generated simultaneously and independently at distributed sites. Furthermore, this data generation does not require any modifications to the existing codes. Each code involved in Eq. (2) may simply be replicated over many processors and executed with different inputs Q. It is a classic case of the so-called coarse-grain parallelism in computing. Thus, there is no additional up-front reprogramming cost, although there is some cost penalty in communication and bookkeeping.

The BLISS computational cost as measured by the elapsed time is primarily in repetitive execution of Eq. (2) that for a large-scale problem overshadows the cost of Eq. (5). For a quadratic polynomial RS in the Q space of N dimensions, the minimum number of points is $M \cong \mathcal{O}(N^2/2)$. However, one may choose to increase the number of points to $M_2 = M_1 M$, $M_1 > 1$, to improve the RS accuracy, in which case the RS may be fitted by means of the method of least squares (or an equivalent) with the added benefit of smoothing that lessens the probability of entrapment in a local minimum. Furthermore, the procedure of solving the sequence Eqs. (2–5) requires a problem-dependent number of cycles, N_c, so the total number of points increases to $\mathcal{O}(N_c N^2/2)$. Consequently, the quadratic RS may be impractical beyond $N = 12$–20 in applications where the elapsed computing time per point is substantial and all calculations are sequential. Therefore, the use of BLISS 2000 in large-scale applications depends on the technology of MCDP for compressing all M operations into the elapsed time of one. Such compression not only extends radically the RS limits of practicality now, but also provides for their further relaxation with the progress of the MCDP technology.

Another way to circumvent the "curse of dimensionality" is to settle for a linear form of RS that makes $M \cong O(N)$. Then, the iterative process of solving the sequence of Eqs. (2–5) becomes similar to the sequential linear programming (or its more refined version of the sequential approximate programming[29]). Many years of experience with the described practice with diverse applications indicates that the number of times N_A the approximations must be refreshed in these sequential processes depends on the class of application but stays independent of N and reasonably small within that class. For instance,[29,30] in structural optimization dominated by membrane stresses, typically, $N_A = 5$–10, and it rises to 20–30 where bending is prevalent.[30] The total number of points to be evaluated in such a process is of $\mathcal{O}(N_A N)$. Recent numerical evidence attesting to effectiveness of the linear RS in a sequential approximate optimization is provided in Refs. 31 and 32, where the linear RS is used in an MCDP environment, and in Ref. 16, where it is an enhancement in CO.

Consequently, substituting N_A and N_c expected for a particular BB and checking whether $\mathcal{O}(N_A N) < \mathcal{O}(N_c N^2/2)$ is true constitute an important factor in the choice between the linear or quadratic RS for that particular BB.

Another factor is an organizational one. The RS refreshing operations [Eq. (2)] need to be completed at the outset of each cycle so that a degree of coordination among the groups performing these operations is necessary. The cost of that coordination adds to the cost of the entire procedure. On that score, the higher-order RS are superior because N_c is likely to be smaller than N_A; hence, the coordination is needed less often.

IV. Numerical Example

A supersonic business jet (SBJ) shown in Fig. 1 is a numerical example for the BLISS 2000 algorithm. It is the same case that was used in Refs. 15, 24, and 25 in its overall layout, but sufficiently different in detail to make the results incomparable. The present case and its relation to the earlier one are documented completely in Ref. 33. The model of the aircraft system is reduced to four BBs shown in Fig. 3, which shows also their data exchanges in a standard format of the data dependence matrix, also known as the n-square format. That format depicts the modules strung along the diagonal. Each module accepts input vertically from above or below, and outputs horizontally, left or right.

Table 1 Variables and dimensionality data for RS

BB	Output	Input variables X_{sh}	Input variables Y^*	Number of coefficients w	Number of inputs	Number of points NS
Structures	W_T, W_F, Θ	t/c, AR_W, Λ_W, S_{ref}, S_{HT}, AR_{HT}, λ	L, W_E	(3) 2	11	78
Aerodynamics	L, D, L/D	t/c, h, M, AR_W, Λ_W, S_{ref}, S_{HT}, AR_{HT}, λ	W_T, Θ, ESF[a]	(3) 2	14	120
Propulsion	SF, C, W_E, ESF	h, M	D	(3) 2	5	21
Performance	Range	h, M	W_T, W_F, L/D, SFC[b]	N/A	6	28

[a]Engine scale factor. [b]Specific fuel consumption.

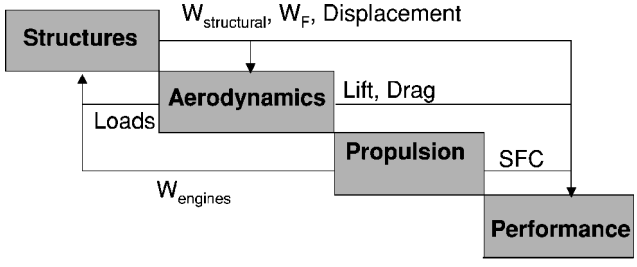

Fig. 3 Numerical example: disciplines (BBs) and a few instances of coupling data inscribed by the arrows that represent data links.

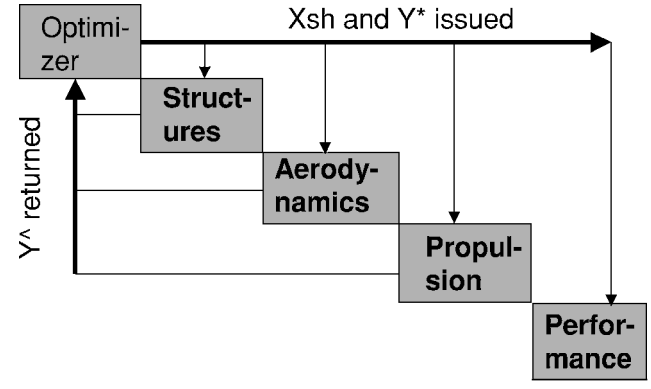

Fig. 4 System optimizer issues all inputs X_{sh} and Y^* and receives approximate $\hat{Y}$ from RS surrogates of the BBs (Table 1 defines X_{sh}, Y^*, and $\hat{Y}$).

The BB Structures comprise a plate representation of the wingbox, connected to a rigid-beam model of the fuselage to which a horizontal tail is attached. The wingbox strength (stress and local buckling constraints) and stiffness account for the wing sandwich covers. The wingbox volume is the fuel tank volume. The BB Aerodynamics calculates lift distribution sensitive to the wing trapezoidal geometry changes and to the local variations of the angle of attack generated by an aerodynamic twist and by the elastic deformation. This BB includes also computation of the drag accounting for the wave drag and accounts for the trim constraints by performing a trim analysis involving the tail volume. The BB Propulsion simulates the propulsion by interpolation of a lookup table that contains data on the thrust and specific fuel consumption as functions of the Mach number and altitude. Finally, the BB Performance computes the flight range by the Breguet equation.

The system objective is the flight range under fixed takeoff gross weight (TOGW). The design variables are chosen to correspond to the system level and to the individual BBs, with the number of variables in each category kept to a minimum judged adequate for the numerical example purposes. The system design variables govern the geometry and include the Mach number and altitude. The local variables are in Structures, the cross-sectional dimensions of the sandwich wing covers at several locations over the wing; in Aerodynamics, the horizontal tail sweep angle, wing, and horizontal tail location coordinates; and in Propulsion, the throttle setting. Table 1 provides more information for the input, output, and the RS statistics for the BBs. The field variables of aerodynamic loads and the wing twist were condensed to their distribution parameters. As to w, their number in a BB equals, in general, the number of the $\hat{Y}$ variables indicated in parentheses in the w column. However, if a particular $\hat{Y}$ in a BB may be identified such that its maximization would never bring any improvement to the system, for example, the structural weight, the corresponding w may be held always positive and the w vector may be normalized by that w to reduce the dimensionality of the RS. Table 1 indicates the lengths of the w vectors thus reduced by the numbers shown without parentheses in the w column.

The n-square diagram in Fig. 4 corresponds to the one in Fig. 3. It shows the system-level optimizer as the sole input source for all BBs. Each BB in Fig. 4 is now a SRS representation of that BB suboptimization. The c constraints equate the coupling variables issued by the optimizer Y^* to those actually computed $\hat{Y}$.

The design points for construction of RS are placed using the D-optimal technique from the DOE methodology. The RS are quadratic polynomials. As the procedure progresses, they are being periodically updated to keep them centered in the Q space around the latest optimal solution, to prevent the optimal solution from remaining lodged against the RS boundary, and to reduce the RS span as the procedure homes on the system optimum.

The BB analysis codes are in FORTRAN, and MATLAB® Optimization Toolbox is the tool for optimization at the BB level and the system level. The main program that executes the entire procedure is also written in MATLAB.

In this example, BLISS 2000 manipulates the system and local design variables adjusting the wing structural weight, engine weight, aerodynamic drag, fuel volume, and fuel consumption to maximize the range.

The results are very voluminous,[33] and so only a small but representative sample is displayed herein in Table 2, which shows how the objective and a few local and system variables were changing over the BLISS cycles. In this case, the result variations diminish to the level commensurate with the accuracy of the analysis typically after 6 or 8 cycles; hence, the number of cycles is set to 10. The data are shown normalized by their values obtained in iteration 10; the latter are also presented in the rightmost column. The BB internal optimization histograms are not shown because the local optimization performance has no bearing on the overall convergence of the method owing to the BB autonomy.

Table 2 includes the objective function of the flight range. At a cycle number, Table 2 shows the range predicted by RS, R_{rs}, used in that cycle, and the range actually computed, R_{an}, at the outset of the next cycle. The two values converge into a complete agreement in five cycles. The convergence is oscillatory partially due to the initial differences of the RS-predicted and actual values and partially due to the search of a compromise among the multitude of tradeoffs at two levels accounting for constraints.

The primary significance of the Table 2 results is in demonstration of the improvement of the design and of the convergence of the data output by analysis and those obtained from the RS approximations. There was no attempt to solve the test case by the SAND (all-in-one) approach for a number of reasons discussed in conjunction with Eq. (1). Table 2 shows the method transform the initially infeasible design to one where all the constraints are satisfied and the objective is improved by 47%. That particular value of improvement is

Table 2 Sample of data changing over 10 BLISS cycles

Normalized[a] parameter	Cycle 0	1	2	3	4	5	6	7	8	9	10	Value from cycle 10
M	1.08	1.01	1.07	1.10	1.05	1.03	1.00	0.99	1.00	0.99	1.00	1.663
TOGW	1.06	1.11	1.19	1.12	1.05	1.02	1.02	1.01	1.01	1.00	1.00	14,246 kg
S_{ref}	0.69	0.35	0.77	0.83	0.77	0.72	0.85	0.86	0.93	0.98	1.00	53.7 m^2
ESF	1.94	1.22	0.90	1.05	1.10	1.03	0.97	1.00	1.03	1.01	1.00	0.517
t/c	2.16	2.89	2.89	1.60	1.69	1.55	1.31	1.26	1.12	1.02	1.00	0.035
$ts1$[b]	1.04	1.03	1.05	1.07	1.04	1.00	0.98	1.00	0.99	0.99	1.00	9.75 cm
T, %	1.70	1.03	1.52	1.22	1.10	1.07	1.01	0.98	1.00	1.00	1.00	20.6
L_{ht}, % mean aerodynamic chord	0.71	0.61	1.00	1.00	0.60	0.76	1.00	1.00	1.00	1.00	1.00	350
w_1	0.00	0.00	0.48	1.19	1.90	1.67	2.14	1.43	1.10	1.00	1.00	0.210
w_3	0.00	1.79	1.61	1.02	1.16	0.89	1.25	1.02	1.00	1.02	1.00	0.560
R_{rs}	——	0.77	0.93	1.20	0.76	0.91	0.97	0.99	0.98	0.99	1.00	5247
R_{an}	——	0.68	0.91	1.07	0.77	0.91	0.97	0.99	0.98	0.99	1.00	9717 km

[a]Normalized by value from cycle 10. [b]Sandwich caliper thickness of the wing upper cover inboard segment.

significant merely as a metric of performance of the method in this particular test, but the range final value must not be interpreted as realistic. The analysis scope and fidelity and the completeness of the set of variables and constraints would have to be much higher for such an interpretation.

V. Conclusions

The BLISS 2000 method is intended specifically for support of design projects that require collaboration of autonomous, distributed specialty groups and exploitation of the Massively Concurrent Distributed Processing (MCDP) technology.

The method decomposes the overall system optimization into subotimizations associated with the specialty groups. Response surfaces (RS) approximate the suboptimization optimal results in the coordinating, system-level optimization that, provably in convex problems, leads to the same solution, subject to the approximation errors, that would have been obtained without decomposition. An example of a simplified, conceptual design of an aircraft illustrates the method.

The proof-of-the-concept numerical example reported herein showed BLISS performing as intended, that is, transforming the initially infeasible aircraft system design into a feasible one, converging the approximate analysis estimates with the full analysis results, and utilizing existing codes unchanged as modules. Preliminary assessment of the method merits and demerits follows.

The BLISS 2000 performance depends on the quality of approximation of the subsystem optimization results as functions of the system-level design variables and coupling variables. The number of times these approximation need to be updated and the number of new design points in these updates are problem dependent. Assessment of the approximation errors is a factor in choosing between the linear or higher-order approximations for the individual modules and in deciding on the frequency of the approximation updates.

BLISS 2000 tends to generate a large amount of computing and voluminous intermediate data. On the other hand, it simplifies the entire procedure conceptually to reduce the human effort and time required for learning a new method. This appears to be a cost-effective tradeoff, considering that computing gets less expensive all of the time, while the labor costs grow. The elapsed time consumed by the BLISS-generated computation is compressed by the MCDP technology. That technology has established itself as the future of computing in science and engineering, and BLISS being intrinsically coarse-grain parallel is posed to exploit that technology and grow with it.

The BLISS method performs best in applications that decompose into subsystems with numerous local design variables and constraints leaving a relatively small number of design variables at the system level. A nearly complete autonomy of the operations in the subsystems is the key BLISS advantage that enables full utilization of the local knowledge and control over the budget and task time.

The system level-optimization benefits from nearly instantaneous response of the approximations; therefore, it does not call for any particularly high sophistication or efficiency of the search technique.

Further research is recommended to determine how much BLISS could reduce the elapsed project time and whether it would benefit from the alternative formulations of the module-level optimization, use of the linear instead of quadratic form of the RSs, and approximations other than RSs. It is through generation of experience in large-scale practical applications that guidelines for further gains in efficiency and accuracy will develop.

References

[1]Bellman, R. E., and Dreyfus, S. E., *Applied Dynamic Programming*, Princeton Univ. Press, Princeton, NJ, 1962.

[2]Dantzig, G. B., and Wolfe, P., "The Decomposition Algorithm for Linear Programing," *Operations Research*, Vol. 8, 1960.

[3]Mesarovic, M. D., Macko, D., and Takahara, Y., *Theory of Hierarchical, Multilevel Systems*, Academic Press, New York, 1970.

[4]Lasdon, L., *Optimization Theory for Large Scale Systems*, Academic Press, New York, 1970.

[5]Wismer, D. A. (ed.), *Optimization Methods for Large Scale Systems*, McGraw–Hill, New York, 1970.

[6]Schmit, L. A., and Ramanathan, R. K., "Multilevel Approach to Minimum Weight Design Including Buckling Constraints," *AIAA Journal*, Vol. 16, No. 2, 1978, pp. 97–104.

[7]Sobieszczanski-Sobieski, J., "Multidisciplinary Design Optimization; an Emerging New Engineering Discipline, *Advances in Structural Optimization*, edited by J. Herskovits, Kluwer Academic, Norwell, MA, 1995, pp. 483–496.

[8]Cramer, E. J., Dennis. J., Frank, P. D., Lewis, R. M., and Shubin, G. R., "Problem Formulation for Multidisciplinary Design Optimization, *SIAM Journal on Control and Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.

[9]Alexandrov, N., and Hussaini, M. Y. (eds.), *Multidisciplinary Design Optimization—State of the Art, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[10]Sobieszczanski-Sobieski, J., "A Linear Decomposition Method for Large Optimization Problems," NASA TM-83248, Feb. 1982.

[11]Sobieszczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems," *Proceedings of NASA/USAF Symposium on Multidisciplinary Optimization*, NASA CP-3031, Vol. 1, Sept. 1988, pp. 51–78.

[12]Renaud, J., and Gabriele, G., "Improved Coordination in Nonherarchic System Optimization," *AIAA Journal*, Vol. 31, No. 12, 1993, pp. 2267–2273.

[13]Wujek, B. A., Renaud, J. E., Batill, S. M., and Brockman, J. B., "Design Flow Management and Multidisciplinary Design Optimization in Application to Aircraft Concept Sizing," AIAA Paper 96-0713, Jan. 1996.

[14]Braun, R. D., and Kroo, I. M., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *Multidisciplinary Design Optimization: State of the Art, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, edited by N. Alexandrov and M. Y. Hussaini, Society for Industrial and Applied Mathematics, Philadelphia, 1997, pp. 98–116.

[15]Sobieszczanski-Sobieski, J., Agte, J. S., and Sandusky R. R., Jr., "Bilevel Integrated System Synthesis," *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 164–172.

[16]Sobieski, I. P., and Kroo, I., "Collaborative Optimization Using Response Surface Estimation," *AIAA Journal*, Vol. 38, No. 10, 2000, pp. 1931–1938.

[17]Papalambros, P., and Michelena, N., "Model-based Partitioning in Optimal Design of Large Engineering Systems," *Multidisciplinary Design Optimization: State of the Art, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, edited by N. Alexandrov and M. Y. Hussaini, Society for Industrial and Applied Mathematics, Philadelphia, 1997, pp. 98–116.

[18]Wagner, T., and Papalambros, P., "A General Framework for Decomposition Analysis in Optimal Design," *Advances in Design Automation*, edited by B. J. Gilmore, Vol. 2, American Society of Mathematical Engineers, New York, 1993, pp. 315–325.

[19]Kodiyalam, S., and Sobieszczanski-Sobieski, J., "Multidisciplinary Design Optimization—Some Formal Methods, Framework Requirements, and Application to Vehicle Design," *International Journal for Vehicle Design*, Vol. 25, No. 1 and 2 (Special Issue), 2000, pp. 3–22.

[20]Sobieszczanski-Sobieski, J., "Optimization by Decomposition in Structural and Multidisciplinary Optimization," *Optimization of Large Structural Systems*, edited by G. I. N. Rozvany, Vol. 1, NATO ASI Series, Series E: Applied Sciences, Vol. 1, Kluwer, 1993, Sec. 1, pp. 197–233.

[21]Balling, R. J., and Sobieszczanski-Sobieski, J., "Optimization of Coupled Systems: A Critical Overview of Approaches," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 6–17.

[22]Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Design Optimization: Survey of Recent Developments," *Structural Optimization*, Vol. 14, No. 1, Jan. 1997, pp. 1–23.

[23]Kodiyalam, S., and Sobieszczanski-Sobieski, J., "Bilevel Integrated System Synthesis with Response Surfaces," *AIAA Journal*, Vol. 38, No. 8, 2000, pp. 1479–1485.

[24]Agte, J. S., Sobieszczanski-Sobieski, J., and Sandusky, R., "Supersonic Business Jet Design Through Bi-Level Integrated System Synthesis," Society of Automotive Engineers, Paper SAE 1999-01-5622, Oct. 1999.

[25]Sobieszczanski-Sobieski, J., Emiley, M. S., Agte, J., and Sandusky, R., Jr., "Advancement of Bi-level Integrated System Synthesis (BLISS)," AIAA Paper 2000-0421, Jan. 2000; also NASA TM 2000-210305, 2000.

[26]Sobieszczanski-Sobieski, J., and Kodiyalam, S., "BLISS/S: A New Method for Two-Level Structural Optimization," *Structural and Multidisciplinary Optimization Journal*, Vol. 21, No. 1, 1999, pp. 1–11; also AIAA Paper 99-1345, 1999.

[27]Sobieszczanski-Sobieski, J., and Venter, G., "Imparting Desired Attributes by Optimization in Structural Design," AIAA Paper 2003-1546, April 2003.

[28]Rao, S. S., *Engineering Optimization—Theory and Practice*, Wiley, New York, 1996, Chap. 12.3.6.

[29]Haftka, R. T., and Gurdal, Z., *Elements of Structural Optimization*, Kluwer Academic, Norwell, MA, 1992, Chap. 6.4. and example p. 194.

[30]Schmit, L. A., "Structural Synthesis: Genesis and Development," *AIAA Journal*, Vol. 19, No. 10, 1981, pp. 1249–1263.

[31]Yang, R.-J., Gu, L., Tho, C., and Sobieszczanski-Sobieski, J., "Multidisciplinary Design Optimization of a Full Vehicle with High Performance Computing," AIAA Paper 2001-1273, April 2001.

[32]Sobieszczanski-Sobieski, J., Kodiyalam, S., and Yang, R.-J., "Optimization of a Car Body Under Constraints of Noise, Vibration, and Harshness (NVH), and Crash," *Structural Optimization*, Vol. 22, No. 4, 2001, pp. 295–306; also AIAA Paper 2000-1521, 2000.

[33]Altus, T. D., "A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS)," NASA CR-2002-211652, May 2002.

A. Messac
Associate Editor