



AIAA 98-4916

Bi-Level Integrated System Synthesis

J. Sobieszczanski-Sobieski, J. Agte, and R. Sandusky, Jr.

NASA Langley Research Center

Hampton, VA

**7th AIAA/USAF/NASA/ISSMO Symposium
on Multidisciplinary Analysis and Optimization**

September 2-4, 1998 / St. Louis, MO

BI-LEVEL INTEGRATED SYSTEM SYNTHESIS (BLISS)

*Jaroslav Sobieszczanski-Sobieski**, *Jeremy S. Agte†*,
and *Robert R. Sandusky, Jr.‡*

Abstract

BLISS is a method for optimization of engineering systems by decomposition. It separates the system level optimization, having a relatively small number of design variables, from the potentially numerous subsystem optimizations that may each have a large number of local design variables. The subsystem optimizations are autonomous and may be conducted concurrently. Subsystem and system optimizations alternate, linked by sensitivity data, producing a design improvement in each iteration. Starting from a best guess initial design, the method improves that design in iterative cycles, each cycle comprised of two steps. In step one, the system level variables are frozen and the improvement is achieved by separate, concurrent, and autonomous optimizations in the local variable subdomains. In step two, further improvement is sought in the space of the system level variables. Optimum sensitivity data link the second step to the first. The method prototype was implemented using MATLAB and iSIGHT programming software and tested on a simplified, conceptual level supersonic business jet design, and a detailed design of an electronic device. Satisfactory convergence and favorable agreement with

the benchmark results were observed. Modularity of the method is intended to fit the human organization and map well on the computing technology of concurrent processing.

0. Introduction

Optimization of complicated engineering systems by decomposition is motivated by the obvious need to distribute the work over many people and computers to enable simultaneous, multidisciplinary optimization. It is important to partition the large undertaking into subtasks, each small enough to be easily understood and controlled by people responsible for it. This implies granting people in charge of a subtask a measure of authority and autonomy in the subtask execution, and allowing human intervention in the entire optimization process.

Reconciliation of the need for subtask autonomy with the system level challenge of "everything influences everything else" is difficult. Each of the leading MDO methods that have evolved to date (survey papers: Balling and Sobieszczanski-Sobieski, 1996; and Sobieszczanski-Sobieski, J., and Haftka, R. T., 1997) tries to address that difficulty in a different way. In the system optimization based on the Global Sensitivity Equations (GSE) (Sobieszczanski-Sobieski, J. 1990, Olds, J. 1992, Olds, J. 1994), the partitioning applies only in the sensitivity analysis while optimization involves all the design variables simultaneously. The Concurrent SubSpace Optimization method provides for separate optimizations within the modules (Sobieszczanski-Sobieski, J. 1988, Renaud and Gabriele, 1991, 1993, and 1994; Stelmack and S. Batill, 1998) but handles all the design variables simultaneously in the coordination problem. The Collaborative Optimization method (Braun and Kroo, 1996; Sobieski and Kroo, 1998) also enables separate optimizations within the modules, each performed to minimize a difference between the state and design variables and their target values set in a coordination problem. This problem combines the

* Manager, Computational AeroSciences, and Multi-disciplinary Research Coordinator, NASA Langley Research Center, MS 139 Hampton VA 23681, AIAA Fellow

† Graduate Research Scholar Assistant, George Washington University, Joint Institute for Advancement of Flight Sciences, LT, United States Air Force

‡ Professor, George Washington University, Joint Institute for Advancement of Flight Sciences, AIAA Fellow
Copyright © 1998 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental Purposes. All other rights are reserved by the copyright owner.

system optimization with the system analysis, therefore its dimensionality may be quite large.

Most of the above method implementations had to overcome difficulties with integration of dissimilar codes. This has stimulated use of Neural Nets and Response Surfaces as means by which subdomains in the design space may be explored off-line and still be represented to the entire system. Unfortunately, effectiveness of this approach is limited to approximately 12 to 20 independent variables, hence, it is best suited for the early design phase. Consequently, a clear need remains for a method applicable in later design phases when the number of design variables is much larger. Methods that build a path in design space fit that requirement. Ultimately, one needs both domain-exploring methods and path-building methods, enhanced with seamless 'gear-shifting' between the two.

Motivated by the above state of affairs, BLISS attacks the problem by performing an explicit system behavior and sensitivity analysis using the GSE, autonomous optimizations within the subsystems performed to minimize each module contribution to the system objective under the local constraints, and a coordination problem that engages only a relatively small number of the design variables that are shared by the modules. Solution of the coordination problem is guided by the derivatives of the behavior and local design variables with respect to the shared design variables. These derivatives may be computed in two different ways, giving rise to two versions of BLISS.

In either version, BLISS builds a gradient-guided path, alternating between the set of disjointed, modular design subspaces and the common system-level design space. Each segment of that path results in an improved design so that if one starts from a feasible state, the feasibility in each modular design subspace is preserved while the system objective is reduced. In case of an infeasible start, the constraint violations are reduced while the increase of the objective is minimized. Because the system analysis is performed at the outset of each segment of the path, the process can be terminated at any time, if the budget and time limitations so require, with the useful information validated by the last system analysis. In addition to enabling complete human control in the subspace optimization, BLISS allows the engineering team to exercise judgment, at any point in the procedure, to intervene before committing to the next successive pass.

BLISS has been developed in a prototype form and has been successfully demonstrated on the small-scale test

cases reported herein.

1. Notation

- BB - black box, a module, in the mathematical model of a system.
- BBA($Y_r, (Z, X_r)$) - analysis of BB_r to compute Y_r for given Z and X_r
- BBOF $_r$ - BB Objective Function computed in BB_r
- BBOPT $_r(X_r, \phi_r, G_r)$ - optimization in BB_r defined by eq.(2.1/9)
- BBOSA $_r(X_{r,opt}, Z, Y_{r,s})$ - analysis of BB optimum for sensitivity to parameters
- BBSA($D(Y_r, (Z, X_r, Y_{r,s}))$) - sensitivity analysis of BB_r to compute its output derivatives w.r.t. Z , X_r , and $Y_{r,s}$
- $D(V1, V2)$ - total derivative $dV1/dV2$
- $d(V1, V2)$ - partial derivative $\partial V1/\partial V2$;
 $D()$, and $d()$ dimensionality depends on the dimensionalities of $V1$ and $V2$:
 $V1$ and $V2$ - are both scalars, then D and d are scalars
 $V1$ vector, $V2$ scalar, then D and d are vectors
 $V1$ scalar, $V2$ vector, then D and d are vectors
 $V1$ vector, $V2$ vector, then D and R are matrices
- G_o - vector of constraints active at the constrained minimum, length NG_o
- G_r - vector of the constraint functions, $g_{r,t}$ local to BB_r , $g_{r,t} \leq 0$ is a satisfied constraint
- G_{yz} - constraints in a BB that have a stronger dependence on Y and Z , than on X
- GSE - Global Sensitivity Equations (Sobieszcanski-Sobieski, 1990); GSE/OS - GSE/Optimized Subsystems.
- I - identity matrix.
- L - vector of the Lagrange multipliers corresponding to G_o , length NG_o
- LP - Linear Programming
- NB - the number of BBs in the system
- NLP - NonLinear Programming
- opt - subscript denotes optimized quantity
- P - vector of parameters, p_i , kept constant in the process of finding the constrained minimum, length NP.
- SA($(P, Z, X), Y$) - system analysis; a computation that outputs Y for a system defined by P , Z , and X
- SOF - System Objective Function computed in one of the BBs
- SOPT(Z, Φ) - system objective optimization defined by eq. (2.2.3/1)
- SSA($D(Y, (Z, X))$) - system sensitivity analysis to compute sensitivity of the system response Y w.r.t. Z and X
- TOGW - take-off gross weight
- T - superscript denotes transposition.

X_r - vector of the design variables $x_{r,j}$, length NX_r , these variables are local to BB_r ; X without subscript - a vector of all concatenated X_r , length NX
 XL, XU - lower and upper bounds on X , side-constraints.
 Y_r - vector of behavior (state) variables output from BB_r , these are the coupling variables; an element of Y_r is denoted $y_{r,i}$; some of $y_{r,i}$ are routed as inputs to other BBs, and may also be routed as output to the outside; the Y_r length is NY_r ; Y without subscripts - a vector of all concatenated Y_r , length NY
 $Y_{r,s}$ - vector of variables input to BB_r from BB_s , these are the coupling variables; an element of $Y_{r,s}$ is denoted $y_{r,s,i}$; note that by this definition $Y_{r,s}$ is a subset of Y_r , vector length $NY_{r,s}$
 Z - vector of the design variables z_k that are shared by two or more BBs, these are the system-level variables; length NZ
 0 - subscript denotes the present state from which to extrapolate, or the optimal state.
 ZL, ZU - lower and upper bounds on Z , side-constraints
 Δ - increment
 $\Delta ZL, \Delta ZU$ - move limits
 ϕ_r - the local objective function in BB_r
 Φ - the system objective function equated to one, particular $y_{1,i}$

2. The Algorithm

In this section, the symbols defined in Notation are used in a shorthand manner without repeating their definitions.

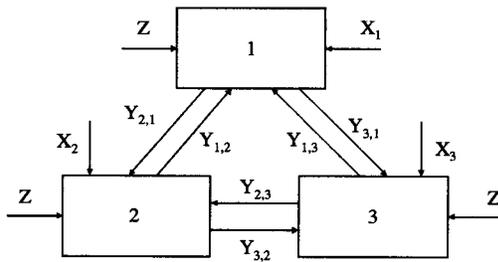


Figure 1: System of Coupled BBs

The algorithm is introduced using an example of a generic system of three BBs, as shown in Figure 1. Three is a number small enough for easy conceptual grasp and compact mathematics, yet large enough to unfold patterns that readily generalize to larger NB. Even though the system in Figure 1 is generic, it may be useful to bear a specific example in mind. Let it be an aircraft so that:

$BB1$ - performance analysis
 $BB2$ - aerodynamics
 $BB3$ - structures
 Φ - maximum range for given mission characteristics
 $Y_{1,2}$ - includes the aerodynamic drag; $Y_{1,3}$ - includes the structural weight; $Y_{2,1}$ - includes Mach number; $Y_{3,1}$ - includes TOGW; $Y_{2,3}$ - includes the structural deformations that alter the aerodynamic shape; $Y_{3,2}$ - includes the aerodynamic loads
 $g_{1,t}$ - a noise abatement constraint on the mission profile; $g_{2,t}$ - limit of the chordwise pressure gradient; $g_{3,t}$ - allowable stress
 $x_{1,j}$ - cruise altitude; $x_{2,j}$ - leading edge radius; $x_{3,j}$ - sheet metal thickness in the wing skin panel No. 138
 z_1 - wing sweep angle; z_2 - wing aspect ratio; z_3 - wing airfoil maximum depth-to-chord ratio; z_4 - location of the engine on the wing

The algorithm depends on the availability of the derivatives of output with respect to input for each BB. That assumes the differentiability of the BB internal relationships to at least the first order. It is immaterial how the derivatives are computed, finite differencing may always be used, but it is expected that in most cases one will utilize one of the more efficient analytical techniques (Adelman and Haftka, 1993).

The algorithm comprises the system analysis and sensitivity analysis, local optimizations inside of the BBs (that includes the BB-internal analyses), and the system optimization. We will not elaborate on SA beyond pointing out that it is highly problem-dependent, and likely to be iterative if there are any non-linearities in the BB analyses. Each pass through the BLISS procedure improves the design in two steps: first by concurrent optimizations of the BBs using the design variables X and holding Z constant; and next, by means of a system-level optimization that utilizes variables Z . We begin with the BB-level optimization.

2.1. BB-level (discipline or subsystem) optimizations.

The basis of the algorithm is the formulation of an objective function unique for each BB such that minimization of that function in each BB results in the minimization of the system objective function. To introduce that formulation let us begin with the system objective function (SOF). The SOF is computed as a single output item in one of the BBs; without loss of generality we assume that it is BB_1 so that

$$\Phi = y_{1,i} \quad (1)$$

is one of the elements of Y_1 .

Total derivatives of Y w.r.t. $x_{r,j}$, $D(Y, x_{r,j})$, are computed according to Sobieszczanski-Sobieski, 1990, by solving a set of simultaneous, algebraic equations known as Global Sensitivity Equations, GSE, (see Appendix, Section 1, for details) for a particular $x_{r,j}$

$$[A] \{D(Y, x_{r,j})\} = \{d(Y, x_{r,j})\} \quad (2)$$

where A is a square matrix, NY_xNY_r , composed of submatrices forming this pattern

$$\begin{bmatrix} I & A_{1,2} & A_{1,3} \\ A_{2,1} & I & A_{2,3} \\ A_{3,1} & A_{3,2} & I \end{bmatrix} \quad (3)$$

where I stands for identity matrix, $NY_r x NY_r$, and $A_{r,s}$ are matrices of the derivatives that capture sensitivity of the BB_r output to input. For example

$$\begin{aligned} A_{2,3} &= -[d(Y_2, Y_3)], NY_2 x NY_3 \\ A_{3,2} &= -[d(Y_3, Y_2)], NY_3 x NY_2 \end{aligned} \quad (4)$$

One should note that eq. 2 can be efficiently solved for many different $x_{r,j}$ using techniques available for linear equations with many right-hand sides.

Having $D(Y, x_{r,j})$ computed from eq. 2 for all $x_{r,j}$, we can express Φ as a function of X by the linear part of the Taylor series

$$\Phi = y_{1,i} = (y_{1,i})_0 + D(y_{1,i}, X_1)^T \Delta X_1 + D(y_{1,i}, X_2)^T \Delta X_2 + D(y_{1,i}, X_3)^T \Delta X_3 \quad (5)$$

where D -terms are vectors of length NX_r .

We see from eq. 5 that

$$\Delta \Phi = D(y_{1,i}, X_1)^T \Delta X_1 + D(y_{1,i}, X_2)^T \Delta X_2 + D(y_{1,i}, X_3)^T \Delta X_3 \quad (6)$$

the three terms showing explicitly the contributions to of the local design variables from each of the three BB_s .

It is apparent that to minimize $\Delta \Phi$ we need to charge each BB with the task of minimizing its own objective. Using BB_2 as an example, objective ϕ_2 is

$$\phi_2 = D(y_{1,i}, X_2)^T \Delta X_{2,j}, j = 1 \rightarrow NX_2 \quad (7)$$

The above equations state mathematically the fundamentally important concept that in a system optimization the contributing disciplines should not optimize themselves for a traditional, discipline-specific objective such as the minimum aerodynamic drag or minimum structural weight. They should optimize themselves for a “synthetic” objective function that measures the influence of the BB_r design variables X_r on the entire system objective function.

Another way to look at it is to observe that, in long-hand

$$\begin{aligned} \phi_2 &= D(y_{1,i}, x_{2,1})^T \Delta X_{2,1} + D(y_{1,i}, x_{2,2})^T \Delta X_{2,2} + \dots \\ &+ D(y_{1,i}, x_{2,j})^T \Delta X_{2,j} + \dots, j = 1 \rightarrow NX_2 \end{aligned} \quad (8)$$

so it may be regarded as a composite objective function commonly used in multiobjective optimization. One may say, therefore, that in a coupled system the local disciplinary or subsystem optimizations should be multiobjective with a composite objective function. The composite objective should be a sum of the local design variables weighted by their influence on the single objective of the whole system. It should be emphasized that this is true also in that particular BB_r where Φ is being computed. In the aircraft example it is $\Phi = y_{1,i}$ in BB_1 according to eq. 1. However, the BB_1 optimization objective is not $\phi_1 = y_{1,i}$. Instead, it is ϕ_1 from an equation analogous to eq.8.

The local optimization problem may be stated formally for BB_2

$$\text{Given: } X_2, Z, \text{ and } Y_{2,1}, Y_{2,3} \quad (9)$$

$$\text{Find: } X_2; \text{ length } NX_2$$

$$\text{Minimize: } \phi_2 = D(y_{1,i}, X_2)^T \Delta X_2$$

$$\text{Satisfy: } G_2 \leq 0, \text{ including side-constraints}$$

Incidentally, we adhere to the convention which calls for minimization of the objective function. If the application requires that function be maximized, as it does in the example of aircraft range, we convert the objective, e.g., $\Phi = -(\text{range})$.

The optimization problem for BB_1 , and BB_3 are analogous. All three problems being independent of each other may be solved concurrently. This is an opportunity for concurrent engineering and parallel processing.

By solving eq. 9 for all three BB_s , we have improved

the system because, according to eq. 5 and 6, we have reduced Φ by $\Delta\Phi$, while satisfying constraints in each BB.

2.2. System-level optimization.

So far we have improved the system by manipulating X in the presence of a constant Z. We can score further improvement by exploiting Zs as variables. To do so we need to know how Z influences $\Phi = y_{1,i}$. That is, we need $D(y_{1,i}, Z)$.

At this point, the BLISS algorithm forks into two alternatives, termed BLISS/A and BLISS/B.

2.2.1. BLISS/A

This version of BLISS computes the derivatives of Y with respect to Z by modified GSE, eq.(2.1/2) (equations from other sections are cited in (), the section number given before the /). The GSE modification accounts for the fact that optimization of a BB turns its X into a function of Y and Z that enter that particular BB as parameters. The modification leads to a new generalization of GSE that takes the following form

$$[M] \begin{Bmatrix} D(Y, z_k) \\ D(X, z_k) \end{Bmatrix} = \begin{Bmatrix} d(Y, z_k) \\ d(X, z_k) \end{Bmatrix} \quad (1)$$

termed GSE/OS for GSE/Optimized Subsystems. The GSE/OS yields a vector $D(Y, z_k)$ and $D(X, z_k)$, and because Φ is one of the elements of Y, $\Phi = y_{1,i}$, we get the desired derivative $D(\Phi, z_k)$. Derivation and details of the GSE/OS structure, including the definition of the matrix M, are in Section 2 of the Appendix. At this point it will suffice to say that the matrix of coefficients in GSE/OS is populated with $d(Y_r, Y_s)$, $d(Y_r, X_r)$, and $d(X_r, Y_s)$. These terms and the RHS terms of $d(Y, z_k)$ and $d(X, z_k)$ are obtained from the following sources

- $d(Y_r, Y_s)$, $d(Y_r, X_r)$, and $d(Y, z_k)$ ----- from BBSA
- $d(X_r, z_k)$, $d(X_r, Y_s)$ ----- from BBOSA

The terms $d(X, z_k)$ and $d(X, Y_s)$ are the derivatives of optimum with respect to parameters that, in principle, may be obtained by differentiation of the Kuhn-Tucker conditions, e.g., an algorithm described in Sobieszcanski-Sobieski et al, 1982. That approach, however, requires second order derivatives of behavior, too costly in most large-scale applications. Therefore, an approximate algorithm adapted from Vanderplaats and Cai, 1986, is given in Section 3 of the Appendix. In that algorithm, parameters are perturbed by a small

increment, one at a time, and the BB optimization is repeated by Linear Programming (LP) starting from the optimal point. Derivatives of optimal X and Y with respect to parameters are then computed by finite differences.

2.2.2. BLISS/B

This version of BLISS avoids calculation of $d(X_r, z_k)$ and $d(X_r, Y_s)$ altogether by using an algorithm that yields $D(\Phi, P)$, where P includes both Y and Z. The algorithm, described in literature (e.g., Barthelemy and Sobieszcanski-Sobieski, 1983) is based on the well-known notion that the Lagrange multipliers may be interpreted as the prices, stated in the units of Φ , for the constraint changes caused by incrementing p_i . For a general case of the objective $F=F(P)$ and $G_o=G_o(P)$, the algorithm gives the following formula for $D(F, P)_o$

$$D(F, P)_o = d(F, P) + L^T d(G_o, P)$$

To use the above in BLISS, consider that in P we have an independent Z but $Y=Y(Z)$ so that the terms $d()$ require chain-differentiation. Hence, the above general formula transforms to

$$D(y_{1,i}, Z)_o^T = (L^T d(G_o, Z))_1 + (L^T d(G_o, Z))_2 + (L^T d(G_o, Z))_3 + [(L^T d(G_o, Y))_1 + (L^T d(G_o, Y))_2 + (L^T d(G_o, Y))_3](D(Y, Z)) + D(y_{1,i}, Z)^T \quad (1)$$

where L is the vector of Lagrange multipliers and ()₁, ()₂, and ()₃ identify the BBs 1, 2, and 3.

The terms in the above equation originate from the following sources:

- $d(G_o, Z)$ and $d(G_o, Y)$ - BBSA performed on isolated BB,
- L - obtained for BB, at the end of BBOPT
- $D(Y, Z)$ - from GSE in SSA
- $D(y_{1,i}, Z)$ - the column corresponding to $y_{1,i}$ in the above matrix $D(Y, Z)$

BLISS/B is substantially simpler in implementation than BLISS/A and it eliminates the computational cost of one LP per parameter Y and Z. Optimizers that yield L as a by-product of optimization are available for use in BBOPT, or L may be obtained as described in Haftka and Gurdal, 1992.

2.2.3. Optimization in the Z-space.

Once $D(y_{1,i}, Z)$ have been computed from either eq.

(2.2.1/1) or as $D(y_{1,i}, Z)_0$ from eq. (2.2.2/1), we can further improve the system objective by executing the following optimization, using any suitable optimizer

Given: Z and Φ_0 (1)

Find: ΔZ

Minimize: $\Phi = \Phi_0 + D(y_{1,i}, Z)^T \Delta Z$

Satisfy: $ZL \leq Z + \Delta Z \leq ZU; \Delta ZL \leq \Delta Z \leq \Delta ZU$

Where Φ_0 is inherited from the previous cycle SA for X and Z (initialized if it is the first cycle). It is recommended to handle the Z constraints by means of a trust-region technique, e.g., Alexandrov 1996. In the above, term $D(y_{1,i}, Z)$ is a constrained derivative that protects $G_0 = 0$ in all BBs. Therefore, the optimization is unconstrained except of the side-constraints and move limits.

However, some BBs may have constraints that depend on Z and Y more strongly than on X (in the extreme case some constraints may not be functions of X at all, only of Y and Z). Such constraints, denoted G_{yz} , may be difficult (or impossible) to satisfy by manipulating only X in BBOPT. To satisfy them, one must add to the Z-space optimization in eq.1 their extrapolated values

$$G_{yz}^T = G_{yz,0}^T + (d(G_{yz}, Z) + d(G_{yz}, Y)D(Y, Z))^T \Delta Z \leq 0 \quad (2)$$

where $d(G_{yz}, Z)$, and $d(G_{yz}, Y)$ are obtained from BBSA. In this instance, the Z-space optimization becomes a constrained one.

3. Iterative Procedure

The two operations, the local optimizations in the BBs and the system-level optimization, described in Sec. 2, result in a new system, altered because of the increments on X and Z. This means that inputs to and outputs from SA, BBA, BBSA, SSA, BBOPT, BBOA (BLISS/A), and SOPT all need to be updated, and the sequence of these operations repeated with the new values of all quantities involved, including new values of all the derivatives because they would change if there were any nonlinearities in the system (as there usually are).

In a large-scale application where execution of each BLISS cycle may require significant resources and time, the engineering team may wish to review the results before committing to the next cycle. That inter-

vention may entail a problem reformulation, such as overriding the variable values, deleting and adding variables, constraints, and even BBs.

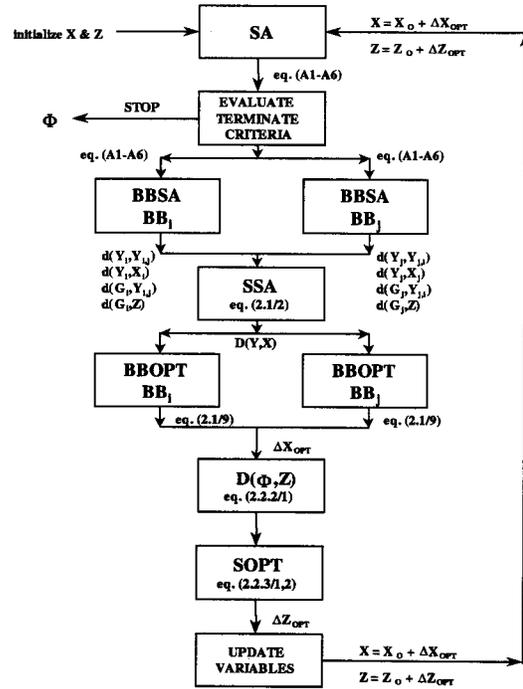


Figure 2: BLISS/B Flowchart

Thus, the following procedure emerges, illustrated also by a flowchart in Figure 2 for BLISS/B with the BLISS/A operations, if different, noted in [].

0. Initialize X & Z

1. SA to get Ys and Gs; this includes BBAs for all BBs.

2. Examine TERMINATION CRITERIA, exercise judgment to override the results, modify the problem formulation, and CONTINUE or STOP.

3. BBSA to obtain $d(Y, X)$, $d(Y_{r,s}, Y_s)$, $d(G, Z)$, and $d(G, Y)$, and SSA, eq. (2.1/2), to get $D(Y, X)$ [and $D(Y, Z)$]; Here is an opportunity for concurrent processing.

4. BBOPT for all BBs, eq. (2.1/9) using ϕ formulated individually for each BB (eq. (2.1/6, 7)), get ϕ_{opt} and ΔX_{opt} ; obtain L for G_0 [skip L]. Here is an opportunity for concurrent processing.

5. Obtain $D(\Phi, Z)$ as in eq. (2.2.2/1). [Execute BBOA to obtain $d(X, Z)$ and $d(X, Y)$, and form and solve GSE/OS (Appendix, Section 3) to generate $D(Y, Z)$]. Here is an opportunity for concurrent processing.

6. SOPT to get ΔZ_{opt} by eq. (2.2.3/1 and 2) herein.

7. Update all quantities, and repeat from 1.

$$X = X_0 + \Delta X_{opt}; Z = Z + \Delta Z_{opt}$$

Note: Termination is placed as #2 after SA to ensure that the full analysis results document the final system design, as opposed to having it documented only by the extrapolated quantities. Also, at this point the engineering team may decide whether to intervene by modifying the variable values, and adding or deleting the design variables and constraints.

When started from a feasible design, the procedure will result in an improved system, while the local constraints are kept satisfied within extrapolation accuracy, even when terminated before convergence.

In case of an infeasible design start, the improvement will be in the sense of reductions in the constraint violations, while the objective may exhibit an increase, at least initially. The procedure achieves the improvement by virtue of optimization alternating between the domain of NB X-spaces (Step #4) and the single Z-space (Step #6).

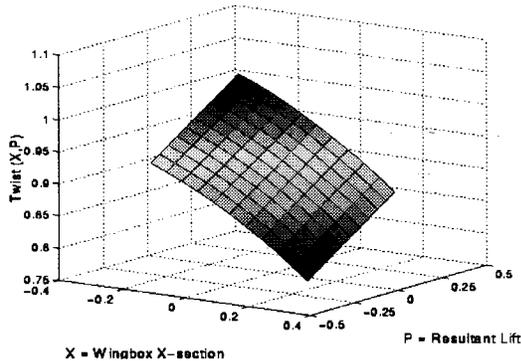


Figure 3: Polynomial Representation of Wing Twist

Caveat: because in BLISS/B the extrapolation of Φ in eq. (2.2.3/1) is based on the Lagrange multipliers in eq. (2.2.2/1), its accuracy depends on the BBOPT yielding a feasible solution, and on the active constraints G_0 remaining active for updated Z . If some constraints leave the active set G_0 , or new constraints enter, a discontinuous change of the extrapolation error may result. For example, consider the wing aspect ratio AR as a Z-variable and suppose that for AR = 3 it is the stress due to the wing bending that is one of the active constraints in the structures BB. If optimization in the Z-space took the design to AR = 4, the next cycle may reveal that the stress constraint is satisfied but a flutter constraint becomes critical. Past experience (Sobieszcanski-Sobieski, 1983) shows that this dis-

continuity is likely to slow, but not to prevent, the process convergence, and may be controlled by adjusting the move limits.

4. Numerical Tests and Examples

BLISS/A was tested on a sample of test problems from Hock and Schittkowski, 1981, and on a design of an electronic package. BLISS/B was exercised on the latter, and also on a very simplified aircraft configuration problem. Both versions of BLISS performed as intended in all of the tests. The sole purpose of these initial numerical experiments was to test and to demonstrate the BLISS procedure logic and data flow, therefore, the BBs were merely surrogates of the numerical processes that need to be used in real applications.

4.1. Aircraft Optimization

The aircraft test was an optimum cruise segment of a supersonic business jet based on the 1995-96 AIAA Student Competition. This problem was selected because of its available data base and the availability of the black boxes written in Visual Basic in form of Excel spreadsheets. The supersonic business jet was modeled as a coupled system of structures (BB₁), aerodynamics (BB₂), propulsion (BB₃), and aircraft range (BB₄). All the disciplines were represented by modules comprising an analysis level typical for an early conceptual design stage.

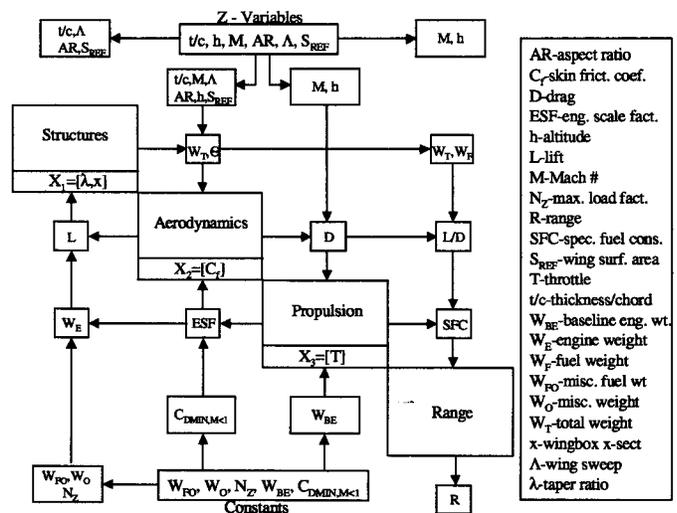


Figure 4: Data Dependencies for Range Optimization

The aircraft optimization was a maximization of the range computed through the Breguet range equation. For testing purposes, additional design and state vari-

ables were introduced in BBs 1 through 3, and functional relationships not present in the original BBs were supplied to reflect what is commonly known about the typical functions involved in design. For example, stress is expected to fall as a reciprocal of the increase of the skin thickness in a wing box. Such relationships were represented by polynomial functions. One plot of such a function is shown in Figure 3, portraying the wing twist as a function of the wing box cross-sectional dimensions scale factor and the wing lift.

var \ cycle*	1	2	3	4	5
Range (SSA)	535.79	1581.67	3425.35	3961.41	3963.98
Extpl. Error	-535.79	-536.67	-431.63	-56.26	-3.43
BB1 Extpl.	17.17	-0.16	-3.26	-0.86	0.00
BB2 Extpl.	16.85	0.00	0.00	0.00	0.00
BB3 Extpl.	26.00	110.92	-76.84	0.00	0.00
X Extpl.	60.02	110.75	-80.10	-0.86	0.00
Z Extpl.	449.19	1301.30	559.90	0.00	0.00
Range (Extpl.)	1045.00	2993.72	3905.15	3960.55	3963.98
λ	0.25	0.14951	0.17476	0.25775	0.38757
x	1	0.75	0.75	0.75	0.75
C_f	1	0.75	0.75	0.75	0.75
T	0.5	0.1676	0.20703	0.15624	0.15624
t/c	0.05	0.06	0.06	0.06	0.06
$h(ft)$	45000	54000	60000	60000	60000
M	1.6	1.4	1.4	1.4	1.4
AR	5.5	4.4	3.3	2.5	2.5
$\Lambda(^{\circ})$	55	66	70	70	70
$S_{ref}(ft^2)$	1000	1200	1400	1500	1500

*One cycle is one pass through the BLISS procedure

Table 1: A/C Results for 20% Move Limit

Section 4 of the Appendix defines the BBs in this example by their input and output variables, and by the functions that link output to input. Table A1 also identifies local constraints and side constraints. Note that BB₂ contains a constraint that does not depend on its X or Y input, thus the Z-space optimization is a constrained one, per eq. (2.2.3/1 & 2). Side constraints on Z were judiciously selected to guard against conditions not accounted for in the BBAs. For example, the lower bound of 2.5 on aspect ratio stemmed from the subsonic performance considerations.

The BBs are coupled by the output-to-input data transfers (design structure matrix) depicted in Figure 4. Note that BB₄ is an analysis-only module and does not feedback any data to other BBs.

This test was conducted entirely using MATLAB 5 and its Optimization Toolbox. To establish a benchmark, the system was first optimized using an all-in-one approach in which the MATLAB optimizer was coupled directly to SA and saw no distinction between the X and Z variables. Next, the test case was executed using BLISS/B, starting at different infeasible initial points

chosen by varying the six design variables that are not arguments in the polynomial functions. The choice of initial values for variables that are arguments of the polynomial functions was limited due to the nature of

num \ den	λ	x	C_f	T	t/c
W_T	0.01146	1.71536	0.01981	-0.15744	0.12714
W_F	0	0	0	0	0.72626
Θ	-0.03342	0.19971	3.31E-15	-1.73E-14	-2.10E-14
L	0.01146	1.71536	0.01981	-0.15744	0.12714
D	-4.19E-05	0.00581	0.12457	-0.00049	0.68108
L/D	0.0115	1.7095	-0.1046	-0.15694	-0.54935
SFC	1.98E-20	-5.07E-18	-2.70E-17	0.08544	0
W_E	-4.40E-05	0.0061	0.13083	-1.03986	0.71531
ESF	-4.19E-05	0.00581	0.12457	-0.99059	0.68108
R	-0.00077	-0.12692	-0.12581	-0.07299	0.10115
num \ den	h	M	AR	Sweep	S_{ref}
W_T	-0.33931	0.31958	0.08208	0.2537	0.55182
W_F	0	0	-0.36043	0	1.09211
Θ	-1.93E-13	-6.15E-14	-0.10766	3.77E-14	-0.10766
L	-0.33931	0.31958	0.08208	0.2537	0.55182
D	-2.1339	2.00984	3.37E-06	-0.83983	0.99838
L/D	1.84108	-1.6507	0.08207	1.10064	-0.43675
SFC	0.12946	0.05555	2.31E-17	-1.86E-16	0
W_E	-2.24115	2.11086	3.54E-06	-0.88204	1.04857
ESF	-2.1339	2.00984	3.37E-06	-0.83983	0.99838
R	2.07616	-1.04784	-0.39618	0.82904	0.15535

Table 2: Normalized Y Derivatives w.r.t. X and Z

the polynomial formulation. This limitation is not a characteristic of the BLISS method itself, as the polynomial functions would not be required in a large scale optimization problem. With the move limits ranging from 10 to 70 %, the procedure convergence was satisfactory through the move limits of 60% for all initial points tested. However, in nearly all cases, no additional improvement in convergence rate was recorded for move limits greater than 20%. For instance, the objective function was advanced to within 1% of the benchmark in 5 passes for move limits 20 and 30%. Onset of an erratic behavior was observed with move limits increased past 60%, the procedure converged or diverged dependent on the starting point.

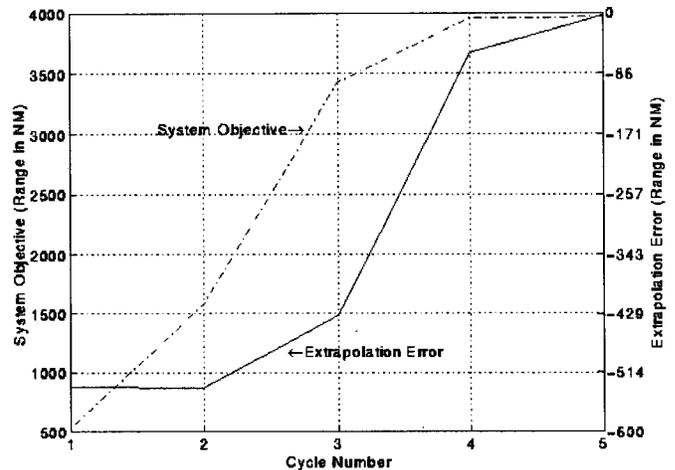


Figure 5: Range and Extrapolation Error Histogram

Table 1 displays a sample of typical results for the move limits value of 20%. It shows that the initial design range was extremely poor, only 536 nm. BLISS/B

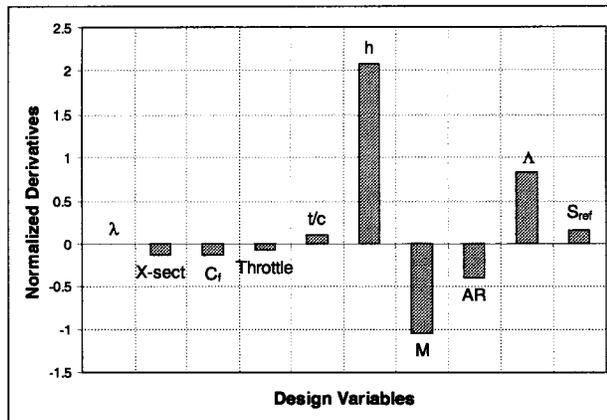


Figure 6: Range Sensitivities (1st cycle)

improvements advanced the range to 3964 nm. The range converged monotonically, although in some cases small amplitude oscillations were observed. Comparison of the extrapolated and actual values of the objective and constraints shows reasonable accuracy and conservatism of the extrapolations. The optimal values of the design variables reflect numerous tradeoffs typical for aircraft design. For instance, optimal t/c resulted, in part, from a trade-off between the wave drag and structural weight. Table 2 shows normalized (logarithmic) derivatives of all Y_s , including the range, w.r.t. all the X and Z variables, sampled in Cycle 1 to illustrate sensitivity of the system solution to design variables.

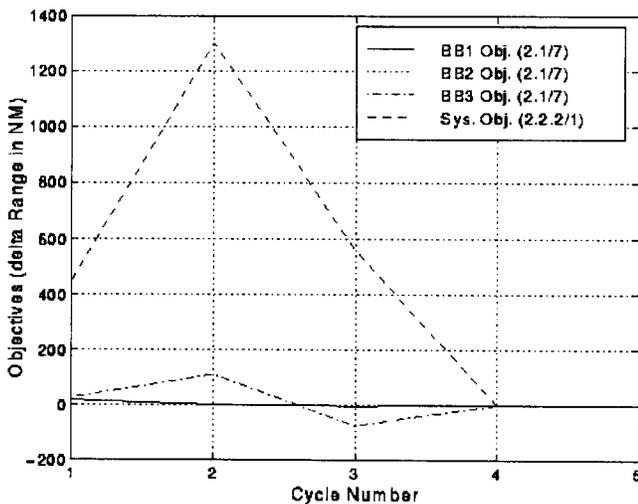


Figure 7: BB and System Contributions to Range

Figure 5 illustrates the range histogram, and depicts the extrapolation error as being effectively controlled by the move limits. Range sensitivities to X and Z variables are shown in Figure 6. As expected, altitude and Mach number have the largest effect on the objective function, while taper ratio has the smallest.

Figure 7 shows the individual BB and system contributions to the range objective in each cycle. Here it is observed that, in this particular case, the contribution of system level variables is significantly larger than that of the local variables in the extrapolation of range.

This test case was also implemented in a software package for system analysis and optimization called iSIGHT (iSIGHT, 1998). The iSIGHT and MATLAB results cross-check was completely satisfactory.

4.2 Electronic Package optimization

The electronic packaging was introduced as an MDO problem in Renaud, 1993. Its electrical and thermal subsystems are coupled because component resistance is influenced by operating temperatures and the temperatures depend on resistance.

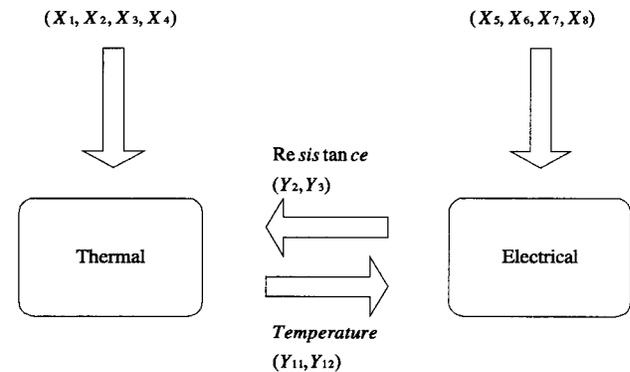


Figure 8: Electronic Packaging Data Dependencies

The objective of the problem is to maximize the watt density for the electronic package subject to constraints. The constraints require the operation temperatures for the resistors to be below a threshold temperature and the current through the two resistors to be equal. The system diagram in Figure 8 shows the data dependencies for two BBs, representing electrical resistance analysis and thermal analysis. As Figure 8 indicates, there are no “natural” Z 's in this case. Therefore, Z 's were created as targets imposed on each of the Y 's and the BBOPT's were required to match the Y values to those Z targets (similar as it is done in

the Collaborative Optimization method). Details of the electronic packaging problem may be found in Padula, 1996.

This test case was implemented in iSIGHT using BLISS/A and B. A benchmark result was obtained by executing an all-in-one optimization from various starting points (“A-in-O” column). BLISS/A and B were started from the same points. Table 4 displays the benchmark and the BLISS/A and B results as showing a good agreement. Table 4 also indicates a comparison of the computational labor (the “Work” column) measured by the number of BB evaluations necessary to converge the fixed-point iterations in BBAs and in SA, all repeated as needed to compute derivatives by finite-differences in a gradient-guided optimization. As Table 4 shows, the BLISS/B computational labor was substantially lower than the benchmark case and that of BLISS/A was between the two.

Method	Case	Initial Design Objective	Init. Des. Max Constr. Viol.	Final Design Objective	Fin. Des. Max Constr. Viol.	Work
A-in-O	1	7.79440E+01	2.16630E-08	6.39720E+05	1.22E-03	498
	2	6.83630E+03	-2.89560E-01	6.39720E+05	1.22E-03	264
	3	1.51110E+03	-4.29240E-02	6.36540E+05	1.45E-03	264
	4	1.46070E+01	-1.02490E-03	6.36940E+05	1.42E-03	175
BLISS/A	1	7.79440E+01	2.16630E-08	6.39700E+05	1.20E-03	436
	2	6.83630E+03	-2.89560E-01	6.39050E+05	1.18E-03	508
	3	1.51110E+03	-4.29240E-02	6.39050E+05	-4.89E-04	174
	4	1.46070E+01	-1.02490E-03	6.39290E+05	3.70E-04	313
BLISS/B	1	7.79440E+01	2.16630E-08	6.39720E+05	1.22E-03	365
	2	6.83630E+03	-2.89560E-01	6.39720E+05	1.22E-03	207
	3	1.51110E+03	-4.29240E-02	6.39720E+05	1.22E-03	114
	4	1.46070E+01	-1.02490E-03	6.39720E+05	1.22E-03	105

Table 4: Electronic Packaging Data

5. BLISS Status, Assessment, and Concluding Remarks

A method for engineering system optimization was developed to decompose the problem into a set of local optimizations (large number of detailed design variables) and a system-level optimization (small number of global design variables). Optimum sensitivity data link the subsystem and system level optimizations. There are two variants of the method, BLISS/A and BLISS/B, that differ by the details of that linkage. In the paper, the method algorithm was laid out in detail for a system of three subdomains (modules). Its generalization to NB subdomains is straightforward. The same algorithm may be used to decompose any of the local optimizations, hence optimization may be conducted at more than two levels.

MATLAB and iSIGHT programming languages were used to implement and test the method prototype on a simplified, conceptual level supersonic business jet design, and a detailed design of an electronic device.

Dimensionality and complexity of the preliminary test cases was intentionally kept very low for an expeditious assessment of the method potential before more resources are invested in further development. Favorable agreement with the benchmark results and a satisfactory convergence observed in the above tests provided motivation for such development and future testing in larger applications.

Assessment of BLISS at the above development status is as follows. BLISS relies on linearization of a generally nonlinear optimization, therefore its effectiveness depends on the degree of nonlinearity. As any gradient-guided method, it guarantees a cycle-to-cycle improvement, but if the problem is non-convex, its convergence to the global optimum depends on the starting point and may strongly depend on the move limits. In this regard, BLISS’s strong points are in the procedure being open to human intervention between the cycles and in the autonomy of the subdomain optimizations in local variables. These optimizations may be conducted by any means deemed to be most suitable by disciplinary experts, hence non-convexity, and strong nonlinearities in terms of the local variables often encountered in subdomains, e.g., the local buckling in thin-walled structures, are isolated and prevented from slowing down the system-level optimization convergence. On the other hand, the optimization robustness may be adversely affected by the local constraints leaving and entering the active constraint set. Effect of the above on BLISS/A is much less than on BLISS/B. This is probably the only reason to continue the development of BLISS/A alongside with BLISS/B, even though BLISS/B has a distinct advantage in simplicity and a much lower computational cost. Once there is more information on the relative merits and demerits of both variants, the better variant may be selected.

The demand BLISS puts on the computer storage is the same the subdomains would require for their own, stand-alone optimizations, with exception of the generation and solution of the Global Sensitivity Equations. If there is a pair of BBs that exchange large number of the $y_{r,s,i}$ - quantities, dimensionality of the corresponding matrices that store the derivatives, and computational cost of these derivatives needed to form GSE, may become prohibitive. Some relief may be provided here by application of condensation techniques and by deleting from GSE those derivative matrices that are known to have negligible effect on the system behavior.

The principal advantage of BLISS appears to lie in its separating overall system design considerations from

the considerations of the detail. This makes the resulting mapping of its algorithm fit well on diverse, and potentially dispersed, human organizations. This advantage remains to be demonstrated in further development toward large-scale, complex applications.

6. Acknowledgement

Dr. Srinivas Kodiyalam of the Engineous Co. provided the test results for Aircraft Optimization and the Electronic Package Optimization using a software package iSIGHT. His contribution is gratefully acknowledged.

7. References

- Adelman, H. M.; and Haftka, R. T.: "Sensitivity Analysis of Discrete Systems," In *Structural Optimization: Status and Promise*, Kamat, M. P., ed., AIAA, Washington, D.C., 1993.
- AIAA/UTC/Pratt & Whitney Undergraduate Individual Aircraft Design Competition, "Engine Data Package for the Supersonic Cruise Business Jet RFP." 1995/1996.
- Alexandrov, N.: "Robustness Properties of a Trust Region Framework for Managing Approximations in Engineering Optimization", Proceedings of the 6th AIAA/NASA/USAF Multidisciplinary Analysis and Optimization Symposium, AIAA-96-4102, pp.1056-1059, Bellevue, WA, Sept. 4-6, 1996.
- Balling, R.J.; and Sobieszczanski-Sobieski, J.: "Optimization of Coupled Systems: A Critical Overview of Approaches," AIAA J., Vol. 34, No. 1, pp.6-17, Jan. 1996.
- Barthelemy, J.-F; and Sobieszczanski- Sobieski, J.: "Optimum Sensitivity Derivatives of Objective Functions in Nonlinear Programming," *AIAA Journal*, Vol. 21, No. 6, pp. 913-915, June 1983.
- Braun, R.D.; and Kroo, I.M.: "Development and Application of the Collaborative Optimization Architecture in A Multidisciplinary Design Environment," In *SIAM J., Optimization* 1996; Alexandrov, N., and Hussaini, M. Y., (eds.).
- Hock, W.; Schittkowski, K.: "Test Examples for Non-linear Programming Codes," *Lecture Notes in Economics and Mathematical Systems* (editors: Beckmann and Kunzi), 1981.
- iSIGHT Designers and Developers Manual, version 3.1, Engineous Software Inc., Morrisville, North Carolina, 1998.
- Haftka, R. T.; and Gurdal, Z.: "Elements of Structural Optimization," p.172; Kluwer Publishing, 1992.
- MATLAB manual, the MathWorks, Inc., July 1993, version 5.0, 1997, and MATLAB Optimization Toolbox, User's Guide, 1996.
- Olds, J.: "The suitability of selected multidisciplinary design and optimization techniques to conceptual aerospace vehicle design," Proc. 4th AIAA/NASA/USAF/OAI Symposium on Multidisciplinary Analysis and Optimization (held in Cleveland, OH). AIAA Paper No. 92-4791. 1992.
- Olds, J.: "System sensitivity analysis applied to the conceptual design of a dual-fuel rocket SSTO," Proc. 5th AIAA/NASA/USAF/ISSMO Symp. on Multidisciplinary Analysis and Optimization (held in Panama City Beach, FL). AIAA Paper No. 94-4339. 1994.
- Padula, S. L., Alexandrov, N., and Green, L. L.: "MDO Test Suite at NASA Langley Research Center." AIAA-96-4028. 1996. <http://fmad-www.larc.nasa.gov/mdob/MDOB/index.html>
- Renaud, J. E.; Gabriele, G. A.: "Improved coordination in non-hierarchic system optimization" AIAA J. 31, 2367-2373 . 1993.
- Renaud, J. E.; Gabriele, G. A.: "Approximation in non-hierarchic system optimization" AIAA J. 32, 198-205. 1994.
- Renaud, J. E.; Gabriele, G. A.: "Sequential global approximation in non-hierarchic system decomposition and optimization," In: Gabriele, G. (ed.) *Advances in Design Automation and Design Optimization*, 19th Design Automation Conf. (held in Miami, FL), ASME Publication DE-Vol. 32-1, pp. 191-200.; 1991.
- Renaud, J.E.: *An Optimization Strategy for Multidisciplinary Systems Design*, International Conference on Engineering Design, August 1993.
- Sobieszczanski-Sobieski, J.; Barthelemy, J.-F. M.; and Riley, K. M.: "Sensitivity of Optimum Solutions to Problems Parameters," *AIAA Journal*, Vol. 20, No. 9, pp. 1291-1299, September 1982.

Sobieszczanski-Sobieski, J.: "Optimization by decomposition: a step from hierarchic to non-hierarchic systems" Presented at the Second NASA/Air Force Symp. on Recent Advances in Multidisciplinary Analysis and Optimization (held in Hampton, VA), NASA CP-3031, Part 1. Also NASA TM-101494. 1988.

Sobieszczanski-Sobieski, J.: "Sensitivity of Complex, Internally Coupled Systems" *AIAA Journal*, Vol. 28, No. 1, pp. 153-160, 1990.

Sobieszczanski-Sobieski, J.: "Optimization by Decomposition in Structural and Multidisciplinary Applications". Chapter in *Optimization of Large Structural Systems*; NATO-ASI, Sept. 1991, Berchtesgaden; publ. by Kluwer 1993.

Sobieszczanski-Sobieski, J.; and Haftka, R. T.: "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments". *Structural Optimization*, pp. 1-23, Vol.14, No. 1, August 1997.

Sobieszczanski-Sobieski, J., James, B., and Dovi, A., "Structural Optimization by Multi-Level Optimization," *AIAA Journal*, Vol. 23, Nov. 1983.

Sobieski, I. P., and Kroo, I.M.: "Collaborative Optimization using Response Surface Estimation" *AIAA-98-0915*, 1998.

Stelmack, M.; and Batill, S.: "Neural Network Approximation of Mixed Continuous/Discrete Systems in Multidisciplinary Design," Univ. of Notre Dame, Notre Dame, IN. *AIAA-98-0916*, 1998.

Vanderplaats, G.N., and Cai, H.D.: "Alternative Methods for Calculating Sensitivity of Optimized Designs to Problem Parameters" *NASA CP-2457*, Proceedings of the Conference on Sensitivity Analysis in Engineering; NASA Langley Research Center, Hampton, VA, September 1986.

Appendix

This Appendix provides details of the Global Sensitivity Equations (GSE) applied to a system which optimizes BBs, the details of a technique for the BB Optimum Sensitivity Analysis, and the details of the aircraft range optimization model.

1. Global Sensitivity Equations

Derivatives of Y w.r.t. X, and Z, are obtained rigorously from the Implicit Function Theorem in Sobiesz-

zanski-Sobieski, 1990. The condensed derivation is provided below. It begins by recognizing that

$$(A1) \quad Y_{1,2} = Y_{1,2}(Z, X_2, Y_{2,1}, Y_{2,3})$$

$$(A2) \quad Y_{1,3} = Y_{1,3}(Z, X_3, Y_{3,1}, Y_{3,2})$$

$$(A3) \quad Y_{2,1} = Y_{2,1}(Z, X_1, Y_{1,2}, Y_{1,3})$$

$$(A4) \quad Y_{2,3} = Y_{2,3}(Z, X_3, Y_{3,1}, Y_{3,2})$$

$$(A5) \quad Y_{3,1} = Y_{3,1}(Z, X_1, Y_{1,2}, Y_{1,3})$$

$$(A6) \quad Y_{3,2} = Y_{3,2}(Z, X_2, Y_{2,1}, Y_{2,3})$$

where the independent variables are X and Z.

Observe that eq. A1-A6 are coupled by Y_i , e.g., $Y_{3,1}$ depends on $Y_{1,3}$ in eq. A5, and $Y_{1,3}$ depends on $Y_{3,1}$ in eq. A2. Consider for an example, the chain-differentiation w.r.t. $x_{1,j}$ applied to eq. A3. It yields

$$(A7) \quad D(Y_{2,1}, x_{1,j}) = d(Y_{2,1}, x_{1,j}) + d(Y_{2,1}, Y_1) D(Y_{1,3}, x_{1,j}) + d(Y_{2,1}, Y_3) D(Y_{3,1}, x_{1,j})$$

Repeating the above for the remaining equations, treating $Y_{2,1}$ as a subset of Y_1 , and collecting the terms leads to eq. (2.1/2 and 3).

The derivatives of Y w.r.t. z_k are obtained by simply replacing $x_{r,j}$ with z_k in eq. (2.1/2) to obtain

$$(A8) \quad [A] \{D(Y, z_k)\} = \{d(Y, z_k)\}$$

2. GSE/Optimized Subsystems

In the preceding section both X and Z are independent variables. By virtue of BBOPT conducted for constant Z and Y inputs, X becomes dependent on Z so that derivatives of X w.r.t. exist in addition to derivatives of Y w.r.t. Z. For example, optimal X_2 depends on Z, $Y_{2,1}$, and $Y_{2,3}$, that are parameters in the optimization of BB₂. Hence, to compute the derivatives of Y and X w.r.t. Z, we begin by rewriting the functional relationships in eq. A1-A6, adding the new dependencies in all three BBs in the system,

$$(A9) \quad Y_{1,2} = Y_{1,2}(Z, X_2, Y_{2,1}, Y_{2,3})$$

$$(A10) \quad Y_{1,3} = Y_{1,3}(Z, X_3, Y_{3,1}, Y_{3,2})$$

$$(A11) \quad Y_{2,1} = Y_{2,1}(Z, X_1, Y_{1,2}, Y_{1,3})$$

$$(A12) \quad Y_{2,3} = Y_{2,3}(Z, X_3, Y_{3,1}, Y_{3,2})$$

$$(A13) \quad Y_{3,1} = Y_{3,1}(Z, X_1, Y_{1,2}, Y_{1,3})$$

$$(A14) \quad Y_{3,2} = Y_{3,2}(Z, X_2, Y_{2,1}, Y_{2,3})$$

$$(A15) \quad X_1 = X_1(Z, Y_{1,2}, Y_{1,3})$$

$$(A16) \quad X_2 = X_2(Z, Y_{2,1}, Y_{2,3})$$

$$(A17) \quad X_3 = X_3(Z, Y_{3,1}, Y_{3,2})$$

The same Implicit Function Theorem that is the basis of the GSE derivation may be applied to the above equations to obtain $D(Y,Z)$. For example, by applying chain-differentiation to $Y_{2,1}$ treated as a subset of Y_2 , we obtain

$$(A18) \quad D(Y_{2,z_k}) = d(Y_{2,z_k}) + d(Y_2, X_2)D(X_{2,z_k}) + d(Y_2, Y_1)D(Y_{1,z_k}) + d(Y_2, Y_3)D(Y_{3,z_k})$$

and for X_2 , again as one example:

$$(A19) \quad D(X_{2,z_k}) = d(X_{2,z_k}) + d(X_2, Y_1)D(Y_{1,z_k}) + d(X_2, Y_3)D(Y_{3,z_k})$$

In the above, the D-terms are the total derivatives we seek, while the d-terms are the partial derivatives of two, different kinds. The derivatives of Y_r w.r.t. Y_s and Y_r w.r.t. X_r are obtained from BBSA_r using any sensitivity analysis algorithm appropriate for the particular BB_r (including the option of finite differencing). The derivatives of X_r w.r.t. z_k and X_r w.r.t. Y_s are produced by an analysis of optimum for sensitivity to parameters, BBOSA_r, explained in later in this Appendix.

As a mathematical digression, one should mention at this point that the derivatives termed partial in the above would be called total in both BBSA and BBOSA. This is not a contradiction. It is so because the partial and total derivatives are hierarchically related in a multilevel system of parents and children. What is a total derivative in a child is partial at the parent level. In the application herein, the system of coupled three BBs is a parent, each BB is a child.

The chain-derivative expressions for Y_1, Y_3, X_1 and X_3 look similar to eq. A18 and A19, differences are only in the subscripts. When the entire set of six chain-derivative expressions is written it forms a set of simultaneous, algebraic equations in which the total derivatives such as $D(Y_{2,z_k})$ and $D(X_{2,z_k})$ appear as unknowns. **This is a new generalization of GSE,**

termed GSE/OS for GSE/Optimized Subsystems. For the case of three-BB system, these equations may be presented in a matrix format like this

$$(A20) \quad [M_{yy}]\{D(Y,z_k)\} + [M_{yx}]\{D(X,z_k)\} = d(Y,z_k) \\ [M_{xy}]\{D(Y,z_k)\} + [M_{xx}]\{D(X,z_k)\} = d(X,z_k)$$

The internal structure of the M-matrices in the above is

for $[M_{yy}]$:

$$\left[\begin{array}{c|c|c} I & -d(Y_1, Y_2) & -d(Y_1, Y_3) \\ \hline -d(Y_2, Y_1) & I & -d(Y_2, Y_3) \\ \hline -d(Y_3, Y_1) & -d(Y_3, Y_2) & I \end{array} \right]$$

for $[M_{yx}]$:

$$\left[\begin{array}{c|c|c} -d(Y_1, X_1) & 0 & 0 \\ \hline 0 & -d(Y_2, X_2) & 0 \\ \hline 0 & 0 & -d(Y_3, X_3) \end{array} \right]$$

for $[M_{xy}]$:

$$\left[\begin{array}{c|c|c} 0 & -d(X_1, Y_2) & -d(X_1, Y_3) \\ \hline -d(X_2, Y_1) & 0 & -d(X_2, Y_3) \\ \hline -d(X_3, Y_1) & -d(X_3, Y_2) & 0 \end{array} \right]$$

and for $[M_{xx}]$:

$$\left[\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & I & 0 \\ \hline 0 & 0 & I \end{array} \right]$$

Again, in the above, all $Y_{r,s}$ are folded into Y_r for compactness, and the terms are falling into the previously introduced categories as follows:

- M_{yy}, M_{yx} , and $d(Y,z_k)$ ----- from BBSA
- M_{xy} and $d(X,z_k)$ ----- from BBOSA

As in GSE, one may obtain $D(Y_{2,z_k})$ and $D(X_{2,z_k})$ for all $z_k, k = 1 \rightarrow NZ$ by means of one of the efficient techniques for linear equations with many right-hand-sides.

3. Black Box Optimum Sensitivity Analysis (BBOSA)

Analysis of optimum for sensitivity to parameters (also called the post-optimum analysis) is preceded by solv-

ing a BB optimization problem

(A21) Given: P
 Find: X
 Minimize: (X,P)
 Satisfy: $G(X,P) \leq 0$, including side-constraints and move limits

in BBOPT_r.

After ϕ_{min} , and X_{opt} are found, one may seek sensitivity of these quantities to the change of P in form of the derivatives $D(\phi_{min},P)$ and $D(X_{opt},P)$.

Vanderplaats and Cai, 1986, review techniques, rigorous and approximate, available for calculating $D(X_{opt},P)$. The technique adapted for the BLISS/A purposes comprises the following steps executed for BB_r:

where P are parameters kept constant while an optimizer manipulates X.

1. Choose parameter P_k , an element of Z or Y, and increment it by a ΔP

In the BLISS application, the parameters P in BB_r are Z_k , and $Y_{r,s}$, because these quantities are kept constant

2. Use derivatives from SSA to extrapolate F and G_o

Table A1: BB Definitions

BB	Inputs	Internal	Outputs
Structures	AR, Λ , t/c , S_{REF} , W_{FO} , W_O , W_E , L, N_Z , λ , x	$t = \frac{t}{c} \frac{S_{REF}}{\sqrt{S_{REF} AR}}$; $b/2 = \sqrt{S_{REF} AR / 2}$; $R = \frac{1 + 2\lambda}{3(1 + \lambda)}$; $\theta =$ $pf(x, b/2, R, L)$; $Fo1 = pf(x)$; $W_w = (0.0051(W_T N_Z)^{0.557} S_{REF}^{0.649})$ $AR^{0.5} (t/c)^{-0.4} (1 + \lambda)^{0.1} (0.1875 S_{REF})^{0.1} / \cos(\Lambda) Fo1$; $W_{FW} =$ $(5 S_{REF} / 18) (2/3)^{42.5}$; $W_F = W_{FW} + W_{FO}$; $W_T = W_O + W_w + W_F$ $+ W_E$; $\sigma 1 \rightarrow \sigma 5 = pf(t/c, L, x, b/2, R)$; Constraints $\sigma 1 \rightarrow \sigma 5 \leq 1.09$; $0.96 \leq \theta \leq 1.04$	$W_T, W_F,$ θ
Aero-dynamics	M, h, Λ , AR, t/c , S_{REF} , W_T , θ , ESF, $C_{Dmin, M < 1}$, C_f	if $h < 36089ft$, $V = M1116.39 \sqrt{1 - (6.875e - 06)h}$, $\rho = (2.377e - 03)(1 - (6.875e - 06h))^{4.2561}$, $V = M968.1$, $\rho = (2.377e - 03)e^{-(h - 36089) / 20806.7}$, if $h > 36089ft$; $C_L = \frac{W_T}{0.5\rho V^2 S_{REF}}$; $Fo1 = pf(ESF, C_f)$; $C_{Dmin} =$ $C_{Dmin, M < 1} Fo1 + 3.05 (t/c)^3 \cos(\Lambda)^2$; $k = 1 / (\pi 0.8AR)$; $Fo2 = pf(\theta)$; $C_D = (C_{Dmin} + k C_L^2) Fo2$; $L = W_T$; $D = C_D 0.5\rho V^2 S_{REF}$; $dp / dx = pf(t/c)$; Constraints $dp / dx \leq 1.04$, evaluated at system level	$L, D, \frac{L}{D}$
Propulsion	M, h, D, W_{BE} , T	$T = T * 16168.6$; $Temp = pf(M, h, T)$; $ESF = (D/3) / T$; $SFC = 1.1324 + 1.5344M - (3.2956e - 05)h - (1.6379e - 04)T$ $- 0.31623M^2 + (8.2138e - 06)Mh - (10.496e - 05)TM - (8.574e - 11)h^2$ $+ (3.8042e - 09)Th + (1.0600e - 08)T^2$; $W_E = 3W_{BE} ESF^{1.05}$; $T_{UA} = 11484 + 10856M - 0.50802h + 3200.2M^2 - 0.29326Mh$ $+ (6.8572e - 06)h^2$; Constraints $0.5 \leq ESF \leq 1.5$; $T \leq T_{UA}$; $Temp \leq 1.02$	SFC, $W_E,$ ESF
Range	M, h, $\frac{L}{D}$, $W_T, W_F,$ SFC	$\theta = 1 - 6.875e - 06 * h$, if $h < 36089ft$; $\theta = 0.7519$ if $h > 36089ft$; $R = \frac{M(L/D)661\sqrt{\theta}}{SFC} \ln\left(\frac{W_T}{W_T - W_F}\right)$	R
Constants		$W_{FO} = 2000lb$; $W_O = 25000lb$; $N_Z = 6g$; $W_{BE} = 4360lb$; $C_{Dmin, M < 1} = 0.01375$	
Side Constraints		$0.1 \leq \lambda \leq 0.4$; $0.75 \leq x \leq 1.25$; $0.75 \leq C_f \leq 1.25$; $0.1 \leq T \leq 1.0$; $0.01 \leq t/c \leq 0.09$; $30000 \leq h \leq 60000$; $1.4 \leq M \leq 1.8$; $2.5 \leq AR \leq 8.5$; $40 \leq \Lambda \leq 70$; $500 \leq S_{REF} \leq 1500$	

linearly and by Linear Programming solve

Find X
 Minimize F
 Satisfy $G_o \leq 0$
 $XL \leq X \leq XU$; where XL and XU incorporate the side constraints and the move limits;

to obtain X_{opt} .

3. Approximate $D(X,P) = \Delta X_{opt}/\Delta P$

4. Repeat from #1 for all elements of Z and Y input into BB_r .

Repeated for all BBs , the above procedure yields a set of $D(X,Z)$ and $D(X,Y)$ to be entered as $d(X,Z)$ and $d(X,Y)$ into GSE/OS, eq. A20. Solution of eq. A20 provides $D(X,Z)$ and $D(Y,Z)$. The latter is substituted into eq. (2.2.3/2), and $D(\Phi,Z)$, extracted from $D(Y,Z)$, goes into eq. (2.2.3/1).

4. A/C Range Optimization Model.

Table A1 shows the equations used in each of the BBs for the aircraft model. Polynomial functions are represented by 'pf()' with independent variables in the parentheses. Each polynomial function is of the form:

$$(A22) \quad PF = A_o + A_i * S^T + (1/2) * S * A_{ij} * S^T$$

Where S is the vector of independent variables, and A_o , A_i , and A_{ij} are coefficient terms.

In calculating the polynomial functions using eq. A22, terms in the S vectors are in the same order as they appear in pf() in Table A1. The off diagonal terms of A_{ij} are random numbers between 0 and 1. For this model, they are

$$A_{ij} = \begin{bmatrix} -- & 0.3970 & 0.8152 & 0.9230 & 0.1108 \\ 0.4252 & -- & 0.6357 & 0.7435 & 0.1138 \\ 0.0329 & 0.8856 & -- & 0.3657 & 0.0019 \\ 0.0878 & 0.7248 & 0.1978 & -- & 0.0169 \\ 0.8955 & 0.4568 & 0.8075 & 0.9239 & -- \end{bmatrix}$$

The remaining coefficient are:

- Θ ---> $A_o = [1.0]$; $A_i = [0.3 \ -0.3 \ -0.3 \ -0.2]$;
 $A_{ii} = [0.4 \ -0.4 \ -0.4 \ 0]$;
- $Fo1$ ---> $A_o = [1.0]$; $A_i = [6.25]$; $A_{ii} = [0]$;
- σ_1 ---> $A_o = [1.0]$; $A_i = [-0.75 \ 0.5 \ -0.75 \ 0.5$

- 0.5]; $A_{ii} = [-2.5 \ 0 \ -2.5 \ 0 \ 0]$;
- σ_2 ---> $A_o = [1.0]$; $A_i = [-0.5 \ 0.333 \ -0.5 \ 0.333 \ 0.333]$; $A_{ii} = [-1.111 \ 0 \ -1.111 \ 0 \ 0]$;
- σ_3 ---> $A_o = [1.0]$; $A_i = [-0.375 \ 0.25 \ -0.375 \ 0.25 \ 0.25]$; $A_{ii} = [-0.625 \ 0 \ -0.625 \ 0 \ 0]$;
- σ_4 ---> $A_o = [1.0]$; $A_i = [-0.3 \ 0.2 \ -0.3 \ 0.2 \ 0.2]$; $A_{ii} = [-0.4 \ 0 \ -0.4 \ 0 \ 0]$;
- σ_5 ---> $A_o = [1.0]$; $A_i = [-0.25 \ 0.1667 \ -0.25 \ 0.1667 \ 0.1667]$; $A_{ii} = [-0.2778 \ 0 \ -0.2778 \ 0 \ 0]$;
- $Fo2$ ---> $A_o = [1.0]$; $A_i = [0.2 \ 0.2]$; $A_{ii} = [0 \ 0]$;
- $Fo3$ ---> $A_o = [1.0]$; $A_i = [0]$; $A_{ii} = [0.04]$;
- dp/dx ---> $A_o = [1.0]$; $A_i = [0.2]$; $A_{ii} = [0]$;
- $Temp$ ---> $A_o = [1.0]$; $A_i = [0.3 \ -0.3 \ 0.3]$;
 $A_{ii} = [0.4 \ -0.4 \ 0.4]$;

Equations for SFC and the upper constraint bound on throttle setting in the Propulsion BB are polynomials representing surfaces fit to engine deck data (AIAA/UTC/Pratt & Whitney, 1995/96).