Optimization of Coupled Systems: A Critical Overview of Approaches

R. J. Balling
Brigham Young University, Provo, Utah 84602-4081
and
J. Sobieszczanski-Sobieski
NASA Langley Research Center, Hampton, Virginia 23681-0001

Nomenclature

A_i	= analyzer discipline i
b	= sparsity factor proportional to bandwidth
CA_i	= number of floating-point multiplications (FLMs) for analyzer discipline i
CE_i	= number of FLMs for evaluator discipline i
$CGSE$	= number of FLMs to solve global sensitivity equations (GSE)
CO	= number of FLMs for optimization
CO_i	= number of FLMs for optimizer discipline i
CS	= number of FLMs for sensitivity analysis
c_i	= cumulative design function discipline i
d_i	= discrepancy function discipline i
E_i	= evaluator discipline i
f	= system objective
f_i	= design objective functions discipline i
g_i	= design constraint functions discipline i
MA	= number of nonlinear analysis iterations
MC	= number of fixed-point iterations
MI	= number of iterations for iterative linear equation solver
MO	= number of nonlinear optimization iterations
NS_i	= number of state variables discipline i
NV	= number of optimization variables
NX	= number of system design variables
NX_i	= number of design variables discipline i
NY	= total number of coupling variables
NY_i	= number of coupling variables input to discipline i
NY_{ij}	= number of coupling variables discipline i to discipline j
O_i	= optimizer discipline i
p_i	= responsibility fraction in discipline i for discipline j
r_i	= residuals in state equations discipline i
SA	= system analyzer
SO	= system optimizer
s_i	= state variables discipline i
x	= system design variables
x_i	= design variables discipline i
y_{ij}	= coupling functions discipline i to discipline j
y_{ij}^*	= coupling variables discipline i to discipline j

Introduction

T HE increased demands of economic competition and the complexity of engineering systems have led to the rapid growth of the field of multidisciplinary design optimization (MDO) over the past decade.[1] The field of MDO is concerned with how to efficiently analyze and optimally design a system governed by multiple coupled disciplines or made up of coupled components. It is a part of the concurrent engineering technology that may well be an enabling technology for complex advanced systems.[2]

Several approaches to problem formulation have emerged in a rather ad hoc fashion over the years. A unified overview of MDO approaches is given in this paper. Fundamental approaches are defined and endowed with consistent names, and a compact notation is introduced that uniquely distinguishes between the large number of

Dr. Richard Balling received B.S. and B.A. degrees in civil engineering and mathematics, respectively, from the University of Utah in 1978. He received M.S. and Ph.D. degrees in structural engineering from the University of California at Berkeley in 1979 and 1982, respectively. He is a professor in the Department of Civil and Environmental Engineering at Brigham Young University, where he has taught since 1982. He is a coauthor of the engineering optimization software packages, OPTDES.BYU and OPTDESX. He took sabbatical leave in 1993–94 to work at the Institute for Computer Applications in Science and Engineering at NASA Langley Research Center.

Jaroslaw Sobieszczanski-Sobieski holds degrees through the doctorate in technical sciences from the Technical University of Warsaw (TUW), Poland, and has served concurrently with industry design and consulting and in faculty positions at TUW from 1955 to 1966, at St. Louis University from 1966 to 1971, at George Washington University from 1972 to 1991, and at the University of Virginia from 1991 to the present and performed postdoctoral research at the Technical University of Norway, Trondheim, 1964–65 and summer 1966. On the staff of NASA Langley since 1971, Dr. Sobieski has held several research and supervisory positions in structural and multidisciplinary analysis and design optimization and is currently Multidisciplinary Research Coordinator of the Center. He is a recipient of the NASA Medal for Exceptional Engineering Achievement, an AIAA Fellow, and the Founding Chairman of the AIAA Technical Committee for Multiplinary Design Optimization and has had several technical publications in professional journals and books.

variations that exist. Some of the fundamental approaches presented here have been significantly modified from the way they were originally introduced, and at least one may be regarded as new. This overview was inspired by the previous work of Ref. 3, which was one of the first attempts to classify approaches to MDO. That work was limited to single-level optimization approaches, whereas the overview presented in this paper covers both single-level and multilevel optimization approaches.

In the past, MDO approaches have been categorized as either hierarchic or nonhierarchic according to the types of systems to which they apply.[4] In hierarchic systems, children disciplines are coupled only to parent disciplines and not to each other. Nonhierarchic systems are more general since no restrictions are placed on how disciplines are coupled. MDO approaches are presented here in a form applicable to general nonhierarchic systems. Approaches that have traditionally been thought to be limited to hierarchic systems are formulated here to apply to nonhierarchic systems.

The fundamental approaches are compared in this paper. Typical values for problem size and iteration parameters are assumed, and the order of magnitude of computational effort is estimated for several test cases and implementation options. Managerial strengths and weaknesses of the fundamental approaches are also noted. Finally, recommendations for needed research are advanced.

Coupled System

Consider the three-discipline system shown in Fig. 1 as a basis for discussion. Each box in the figure represents a module that transforms input to output. These modules may correspond to disciplines or components, but for uniformity the term discipline will be used throughout to refer to a module. The three-discipline system is small enough to keep the discussion simple but large enough to develop patterns that extend to systems that include more disciplines. The system might be an aircraft wing where disciplines 1, 2, and 3 represent aircraft performance, aerodynamics, and structures, respectively. Alternatively, the disciplines might represent physical components of a system such as the fuselage, the wings, and the propulsion system. In either case, the disciplines are coupled to each other by the output-to-input data exchanges. The variables and functions shown in Fig. 1 will now be explained. Table 1 identifies what these variables and functions represent for the aircraft wing example. The reader is encouraged to refer back to this table to put an instance of a physical meaning on the variables, functions, and other terms used in the description of each of the fundamental approaches. This will also help to understand in physical terms how the fundamental approaches work and how they differ.
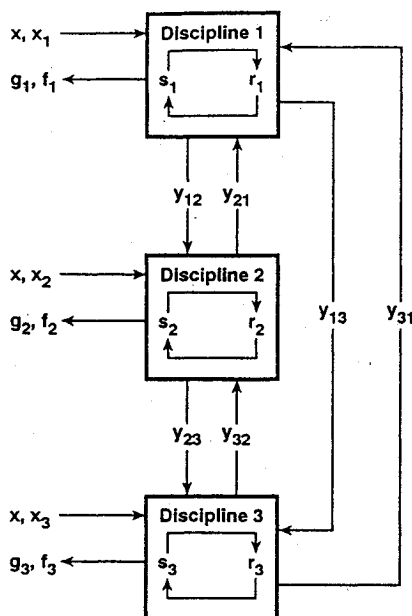

Fig. 1 Three-discipline coupled system.

Table 1 Aircraft wing multidisciplinary optimization problem: examples of variables and design functions

Symbols	Examples
s_1	Fuel required to fly a given mission
s_2	Flow velocities
s_3	Nodal displacements
y_{12}	Angle of attack for various flight regimes
y_{13}	Fuel volume to be contained in the wing
y_{21}	Aerodynamic drag for given angle of attack and Mach number
y_{23}	Aerodynamic loads
y_{31}	Structural weight
y_{32}	Displacements that alter aerodynamic shape
x	Wing planform geometry; airfoil depth-to-chord ratio
x_1	Flight profile: Mach numbers vs attitude data
x_2	Airfoil geometry outside of structural box
x_3	Cross-sectional dimensions of the wing structural box
g_1	Limits on the takeoff and landing run lengths; required go-around climb rate; required range for a given payload
g_2	Aerodynamic pressure distribution limits; angle-of-attack limit
g_3	Stress and displacement limits; required flutter velocity
f_1	Minimum takeoff gross weight
f_2	Minimum drag
f_3	Minimum structural weight

It is assumed that all functions are calculated within the disciplines. This calculation operates on a set of state equations for values of state variables. The vectors s_1, s_2, and s_3 contain the state variables for each of the disciplines.

The vectors r_1, r_2, and r_3 are the residuals in the state equations for the disciplines. Disciplinary analyzers seek values for the state variables that reduce the residuals in the state equations to zero. Disciplinary evaluators evaluate the residuals in the state equations for given values of the state variables. Obviously, the computational effort required by a disciplinary evaluator is significantly less than that required by a disciplinary analyzer. An additional task that is embedded in either disciplinary analyzers or disciplinary evaluators is computation of the coupling functions, the design constraint functions, and the design objective functions, as will be explained later.

The vectors y_{12}, y_{13}, y_{21}, y_{23}, y_{31}, and y_{32} are the coupling functions. Note that y_{ij} contains those functions computed in discipline i that are needed as input to discipline j. It is these coupling functions that complicate the order of execution of the disciplinary analyzers (or evaluators). By associating with each vector of coupling functions a corresponding vector of coupling variables (y_{12}^*, y_{13}^*, y_{21}^*, y_{23}^*, y_{31}^*, and y_{32}^*), the disciplinary analyzers (or evaluators) may be executed in parallel. Each receives coupling variables as input and computes coupling functions as output. One of the tasks of MDO is to satisfy coupling equality between each coupling variable and its corresponding coupling function.

The vectors x, x_1, x_2, and x_3 are mutually exclusive sets of design variables. Design variables represent the independent inputs that distinguish one design from another. Note that x contains system design variables needed by more than one discipline. The vectors x_1, x_2, and x_3 contain disciplinary design variables associated with disciplines 1, 2, and 3, respectively.

The vectors g_1, g_2, and g_3 contain the design constraint functions. The design constraints normally guard against failure and otherwise unacceptable behavior. Only inequality constraints are considered here, and it is assumed that each constraint has been formulated such that zero is its allowable value, and it is satisfied when less than zero.

The vectors f_1, f_2, and f_3 contain the design objective functions. As mentioned before, the design constraint and design objective functions are computed either in disciplinary analyzers or disciplinary evaluators. Design objectives normally minimize cost and maximize benefits and, therefore, may be competing. It is assumed that each objective has been formulated such that it is improved through minimization, and the value of zero is associated with its selected target value. It is possible to cast objectives as constraints and vice versa. Thus, constraint and objective functions, will collectively be called design functions, and their associated inequality constraints and minimization will collectively be called the design criteria.

The task at hand is to determine the values of the design, state, and coupling variables that satisfy the state equations, the coupling equalities, and the design criteria.

Fundamental Approaches

The fundamental approaches to MDO formulation vary in two aspects. First is the distinction between single-level optimization and multilevel optimization. In multilevel optimization approaches, the disciplinary design variables are determined by disciplinary optimizers and the system design variables are determined by the system optimizer. In single-level optimization approaches, both disciplinary and system design variables are determined by the system optimizer.

Second is the choice between simultaneous analysis and design (SAND) and nested analysis and design (NAND). This distinction can be made at both the system and the discipline levels. At the discipline level, SAND implies that the disciplinary design and state variables are determined simultaneously by the optimizer, whereas NAND implies that the optimizer determines only the disciplinary design variables and requires determination of the state variables at each iteration. Thus, at each iteration of the optimizer, disciplinary evaluators are called for SAND, whereas disciplinary analyzers are called for NAND.

At the system level, SAND implies that system design variables and coupling variables are determined simultaneously by the system optimizer, whereas NAND implies that the system optimizer determines only the system design variables and requires determination of the coupling variables at each iteration by calling a system analyzer.

Each fundamental approach has a three part name. The first part indicates whether the approach is a multilevel optimization or a single-level optimization approach. The middle and last parts of the name indicate SAND vs NAND at the system and discipline levels, respectively.

Single-SAND-SAND

In this approach all variables are determined simultaneously by the system optimizer, and disciplinary evaluators are called at each iteration to compute all of the functions. Figure 2 displays boxes corresponding to the system optimizer and to the disciplinary evaluators. This figure, as well as those that follow, shows where the data originate and where they are being used. To that end, individual variables and functions are listed within the boxes where they are determined, and the arrows point to the boxes where each is needed. The figures are not flowcharts and do not convey information about
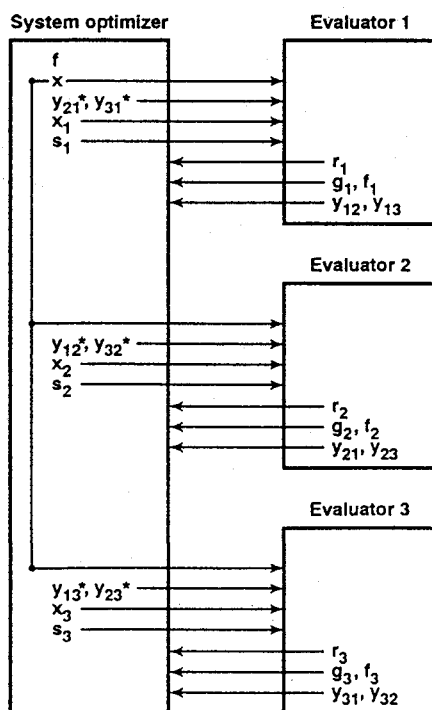

Fig. 2   Single-SAND-SAND approach.

sequencing of the execution of the boxes, except that the execution of the boxes to the right is nested within the execution of the boxes to the left. The execution sequencing is explained in the narrative and by using a special notation introduced later. The Single-SAND-SAND approach has been referred to as the all-at-once approach.[3] It has been demonstrated on single-discipline examples,[5-7] but multidisciplinary applications are scarce. The system optimization problem is as follows.

Find:

$$f, x, x_1, x_2, x_3, y_{12}^*, y_{13}^*, y_{21}^*, y_{23}^*, y_{31}^*, y_{32}^*, s_1, s_2, s_3$$

Minimize:

$$f$$

Satisfy:

$$g_1 < 0, \qquad g_2 < 0, \qquad g_3 < 0$$

$$f_1 < f, \qquad f_2 < f, \qquad f_3 < f$$

$$y_{12} = y_{12}^*, \qquad y_{13} = y_{13}^*$$

$$y_{21} = y_{21}^*, \qquad y_{23} = y_{23}^*$$

$$y_{31} = y_{31}^*, \qquad y_{32} = y_{32}^*$$

$$r_1 = 0, \qquad r_2 = 0, \qquad r_3 = 0$$

Note the addition of the scalar design variable f, the system objective. By minimizing this variable and constraining it to be greater than each of the disciplinary objectives, the maximum of the objectives is effectively minimized. This multiobjective formulation is known as the minimax formulation and is just one of several formulations. Note that the minimax formulation treats all design functions as inequalities whether they be constraints or objectives. As with all multiobjective formulations, the results largely depend on the scaling and the choice of allowable/target values of the design functions.

Single-SAND-NAND

This approach is like the Single-SAND-SAND approach except that disciplinary analyzers rather than disciplinary evaluators are called at each iteration of the system optimizer as shown in Fig. 3. Note the disappearance of the state variables and the residuals since these become internal to the disciplinary analyzers. This approach is also referred to as the individual discipline feasible approach.[3] It has been demonstrated on nonhierarchic, multidisciplinary examples.[8] The system optimization problem is as follows.

Find:

$$f, x, x_1, x_2, x_3, y_{12}^*, y_{13}^*, y_{21}^*, y_{23}^*, y_{31}^*, y_{32}^*$$

Minimize:

$$f$$

Satisfy:

$$g_1 < 0, \qquad g_2 < 0, \qquad g_3 < 0$$

$$f_1 < f, \qquad f_2 < f, \qquad f_3 < f$$

$$y_{12} = y_{12}^*, \qquad y_{13} = y_{13}^*$$

$$y_{21} = y_{21}^*, \qquad y_{23} = y_{23}^*$$

$$y_{31} = y_{31}^*, \qquad y_{32} = y_{32}^*$$

Single-NAND-NAND

In this approach, a system analyzer is called at each iteration of the system optimizer to determine the values of the coupling variables that match their corresponding coupling functions. The
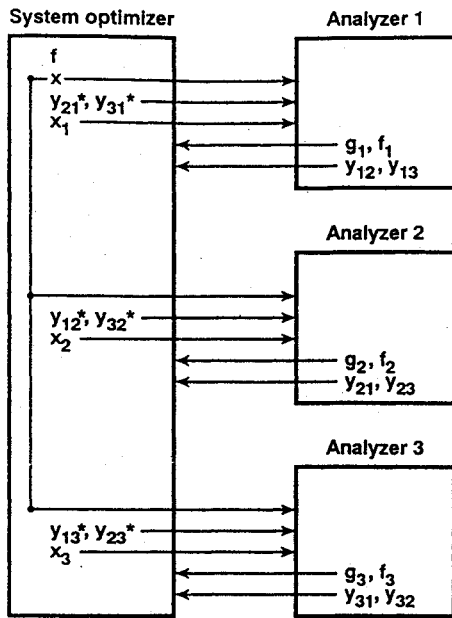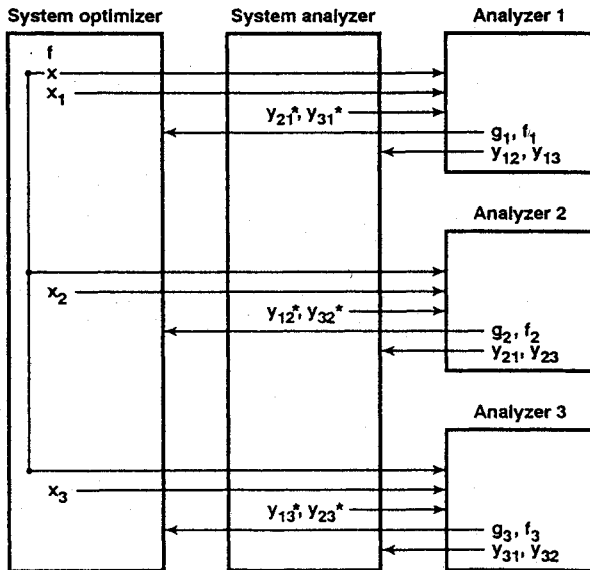
Fig. 3 Single-SAND-NAND approach.

Fig. 4 Single-NAND-NAND approach.

system analyzer itself is typically iterative, and it calls upon disciplinary analyzers at each iteration as shown in Fig. 4. This approach is also referred to as the multidiscipline feasible approach.[3] It has been demonstrated more than any other approach on nonhierarchic, multidisciplinary examples.[8-11] The system optimization problem is as follows.

Find:

$$f, x, x_1, x_2, x_3$$

Minimize:

$$f$$

Satisfy:

$$g_1 < 0, \qquad g_2 < 0, \qquad g_3 < 0$$
$$f_1 < f, \qquad f_2 < f, \qquad f_3 < f$$

The first three fundamental approaches that have now been presented illustrate the difference between SAND and NAND. The NAND approaches have received the most attention. The SAND approaches add coupling and state variables to the set of variables determined by the optimizer. It is likely that this increased set of

variables will affect the convergence properties of the entire process. Whether the total computational effort increases or decreases depends on the problem at hand.

Multi-SAND-SAND

In this approach, disciplinary optimizers that simultaneously determine the disciplinary design and state variables are called at each iteration of the system optimizer as shown in Fig. 5. The system optimizer simultaneously determines the system design and coupling variables. This new approach has never been demonstrated. The system optimization problem is as follows.

Find:

$$f, x, y_{12}^*, y_{13}^*, y_{21}^*, y_{23}^*, y_{31}^*, y_{32}^*$$

Minimize:

$$f$$

Satisfy:

$$d_1 < 0, \qquad d_2 < 0, \qquad d_3 < 0$$

The scalars d_1, d_2, and d_3 are discrepancy functions computed by the disciplinary optimizers. The disciplinary optimization problem for discipline i is as follows.

Find:

$$x_i, s_i, d_i$$

Minimize:

$$d_i$$

Satisfy:

$$g_i < d_i, \qquad f_i - f < d_i \qquad (1)$$
$$y_{ij} - y_{ij}^* < d_i, \qquad y_{ij}^* - y_{ij} < d_i, \qquad \text{for} \quad j \neq i \quad (2)$$
$$r_i < d_i, \qquad -r_i < d_i \qquad (3)$$

Note that the disciplinary optimizer seeks to minimize the maximum discrepancy in the design criteria (1), the coupling equality (2), and the state equations (3). This max norm formulation may cause the discrepancy functions in the system optimization problem to be nonsmooth, and an appropriate system optimization algorithm should be used.[12]
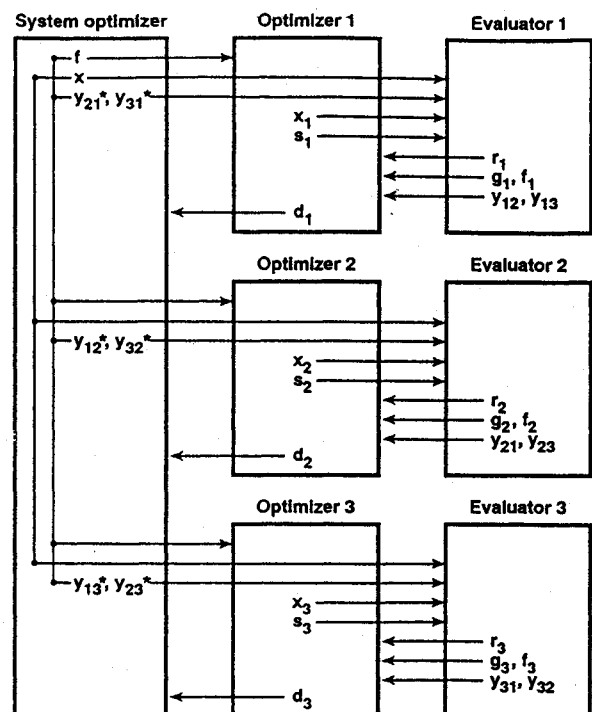

Fig. 5 Multi-SAND-SAND approach.

Multi-SAND-NAND

The only difference between this approach and the Multi-SAND-SAND approach is that disciplinary analyzers are called at each iteration of the disciplinary optimizers as shown in Fig. 6. The system optimization problem is identical to the system optimization problem for the Multi-SAND-SAND approach. The disciplinary optimization problem for discipline i is as follows.

Find:

$$x_i, d_i$$

Minimize:

$$d_i$$

Satisfy:

$$g_i < d_i, \qquad f_i - f < d_i \qquad (4)$$

$$y_{ij} - y_{ij}^* < d_i, \qquad y_{ij}^* - y_{ij} < d_i, \qquad \text{for} \quad j \neq i \qquad (5)$$

This approach has been referred to as multilevel optimization by hierarchic linear decomposition.[13] Various forms of the approach have been widely demonstrated on hierarchic, multidisciplinary examples.[14,15] Note that the preceding formulation is equally applicable to nonhierarchic examples. Nevertheless, it has not yet been demonstrated on nonhierarchic multidisciplinary examples.

Again, note the minimization of the max norm of the discrepancy in design criteria (4) and coupling equality (5). Alternative formulations for the disciplinary optimization problem have been suggested. Originally it was proposed to minimize the norm of discrepancy in design criteria while satisfying coupling equality.[13]

Find:

$$x_i, d_i$$

Minimize:

$$d_i$$

Satisfy:

$$g_i < d_i, \qquad f_i - f < d_i$$

$$y_{ij} = y_{ij}^* \qquad \text{for} \quad j \neq i$$

The converse formulation that minimizes the norm of discrepancy in coupling equality while satisfying the design criteria has also been suggested.[16,17]
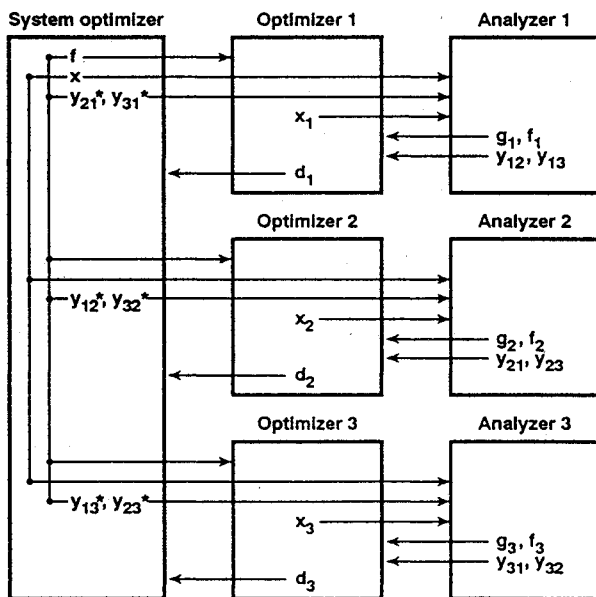
Find:

$$x_i, d_i$$

Minimize:

$$d_i$$

Satisfy:

$$g_i < 0, \qquad f_i < f$$

$$y_{ij} - y_{ij}^* < d_i, \qquad y_{ij}^* - y_{ij} < d_i, \qquad \text{for} \quad j \neq i$$

Both of these latter formulations suffer from the possibility that there may be no feasible solution to the disciplinary optimization problem given the values of the system design variables and coupling variables sent down from the system.

Alternatives to the max norm have also been proposed for the disciplinary optimization problem. One family of norms that has been suggested is the KS family named for Ref. 18.

Find:

$$x_i$$

Minimize:

$$d_i = (1/\rho)\ln\!\left(\Sigma \exp(\rho g_i) + \Sigma \exp[\rho(f_i - f)]\right.$$

$$\left. + \tfrac{1}{2}\Sigma_{j \neq i}\left\{\Sigma \exp\!\left[\rho\!\left(y_{ij} - y_{ij}^*\right)\right] + \Sigma \exp\!\left[\rho\!\left(y_{ij}^* - y_{ij}\right)\right]\right\}\right)$$

The parameter ρ is a positive real number, the sum $\Sigma_{j \neq i}$ is over the disciplines other than discipline i, and the other sums are over the elements of the vectors involved. As ρ goes to infinity, the KS norm becomes equivalent to the max norm. Otherwise, the KS norm is greater than the max norm by an amount that is bounded by $(1/\rho)\ln(m)$ where m is the sum of the lengths of the vectors g_i, f_i, and y_{ij} for $j \neq i$.

Another family of norms that was used in Ref. 16 is the l_p family.

Find:

$$x_i$$

Minimize:

$$d_i = \left\{\Sigma[\max(0, g_i)]^p + \Sigma[\max(0, f_i - f)]^p \right.$$

$$\left. + \Sigma_{j \neq i}\left[\Sigma\left|y_{ij} - y_{ij}^*\right|^p\right]\right\}^{1/p}$$

The parameter p is a positive integer and the sums are the same as for the KS family of norms. The l_2 norm is the familiar euclidean norm, and as p goes to infinity, the l_p norm becomes equivalent to the max norm. Further discussion of the use of these norms in MDO is given elsewhere.[12]

Multi-NAND-NAND

It is difficult to formulate an approach that utilizes both disciplinary optimizers and a system analyzer. An adequate formulation that nests the former inside the latter has not been developed. A Multi-NAND-NAND formulation that first executes the system analyzer and then executes the disciplinary optimizers is possible as shown in Fig. 7. Thus, at each iteration of the system optimizer, the disciplinary design variables are held fixed while a system analyzer is executed to determine the coupling variables, and then the coupling variables are held fixed while disciplinary optimizers are executed to determine the disciplinary design variables. Disciplinary analyzers are called at each iteration of both the system analyzer and the disciplinary optimizers.

A Multi-NAND-NAND approach was first proposed in Ref. 19 and subsequently developed, augmented with heuristics, and demonstrated on nonhierarchic, multidisciplinary examples.[20,21] The formulation presented here is somewhat similar. In addition to determining the coupling variables, the system analyzer computes a scalar cumulative design function for each discipline: $c_i = \max(g_i, f_i - f)$. These functions as well as their sensitivities with respect to disciplinary design variables are sent as constants to the disciplinary optimizers. The optimization problem for discipline i is as follows.
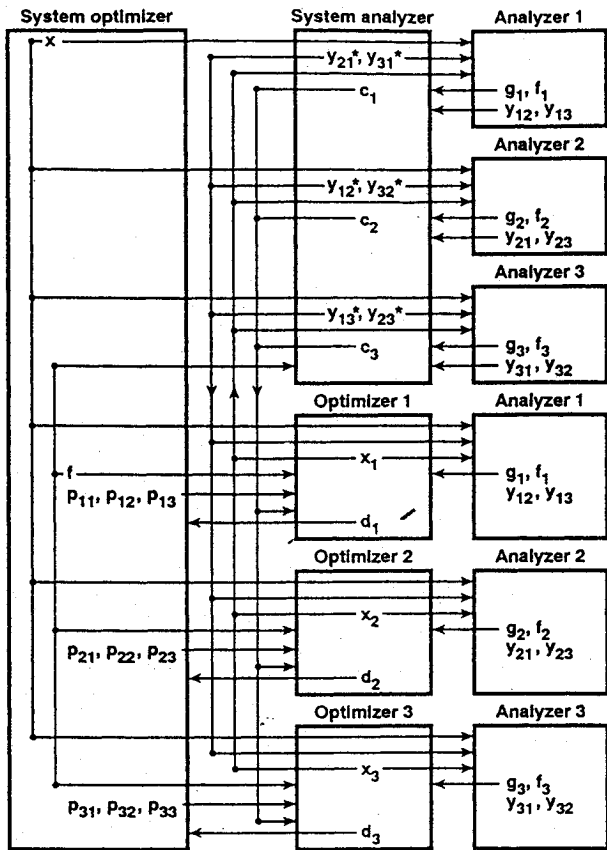

Fig. 6   Multi-SAND-NAND approach.

Fig. 7 Multi-NAND-NAND approach.

Find:

$$x_i, d_i$$

Minimize:

$$d_i$$

Satisfy:

$$g_i - c_i(1 - p_{ii}) < d_i, \qquad f_i - f - c_i(1 - p_{ii}) < d_{ii} \qquad (6)$$

$$c_i^* - c_i(1 - p_{ij}) < d_i \qquad \text{for} \quad j \neq i \qquad (7)$$

The functions c_j^* for $j \neq i$ are first-order approximations of the cumulative design functions for the other disciplines based on the constants and sensitivities received from the system analyzer. Thus, discipline i has some responsibility for satisfying the design constraints of the other disciplines. The scalars p_{ij} for $j \neq i$ in Eq. (7) are the responsibility fractions allocated to discipline i. Similarly, some of the responsibility for satisfying the design constraints of discipline i is shifted to the other disciplines. The scalar p_{ii} in Eq. (6) is the responsibility fraction that remains with discipline i. The system optimization problem ensures that the responsibility fractions sum to unity.

Find:

$$f, x, p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, p_{31}, p_{32}, p_{33}$$

Minimize:

$$f$$

Satisfy:

$$d_1 < 0, \qquad d_2 < 0, \qquad d_3 < 0$$

$$p_{11} + p_{21} + p_{31} = 1, \qquad p_{12} + p_{22} + p_{32} = 1$$

$$p_{13} + p_{23} + p_{33} = 1$$

Variations on the Fundamental Approaches

The fundamental approaches may be modified by 1) mixing, 2) sequencing, and 3) composing. The fundamental approaches may be mixed by varying the usage of disciplinary optimizers, analyzers, and evaluators across disciplines. For example, one discipline may possess several disciplinary design variables, whereas another may have few or even none. It may be wise to employ a disciplinary optimizer with the former (multilevel approach) but not with the latter (single-level approach). Similarly, disciplinary analyzers may be used in some disciplines (NAND approach), whereas disciplinary evaluators are used in others (SAND approach).

Sequencing requires that disciplinary analyzers (or evaluators) are executed sequentially rather than in parallel. This allows deletion of some of the coupling variables. In the three-discipline example, if analyzer 1 is always executed before analyzers 2 and 3, which are still executed in parallel, the coupling functions y_{12} and y_{13} output from analyzer 1 may be sent directly as input to analyzers 2 and 3, and there is no need for the coupling variables y_{12}^* and y_{13}^*. However, the coupling variables y_{21}^*, y_{23}^*, y_{31}^*, and y_{32}^* are still needed. The loss of parallelism may be small if analyzer 1 is significantly longer computationally than analyzers 2 and 3.

Composing allows one to treat a disciplinary optimizer in a multilevel approach as a system optimizer for a system within the discipline. In this way multilevel approaches may be extended to more than two levels. For example, because of the potentially large number of structural design variables, the structures discipline may be treated as a coupled system composed of subdisciplines corresponding to spars, ribs, stringers, and skin as well as the assembled structure. Optimizers, analyzers, and/or evaluators may then be employed for each subdiscipline. The coupling variables between the subdisciplines in the structural system are viewed as the state variables of the structures discipline in the wing system.

Because of the large number of possible variations, it is impractical to try to give names to each. Nevertheless, a compact notation that captures the subtle differences between variations can be introduced (see the preceding Nomenclature). In this notation, [ ] = nested execution, ‖ = parallel execution, and →= sequential execution.

Nested execution has the highest precedence, followed by parallel execution and finally by sequential execution. Using this notation, the compact formulas for the fundamental approaches are as follows.

Single-SAND-SAND:

$$SO[E_1 \| E_2 \| E_3]$$

Single-SAND-NAND:

$$SO[A_1 \| A_2 \| A_3]$$

Single-NAND-NAND:

$$SO[SA[A_1 \| A_2 \| A_3]]$$

Multi-SAND-SAND:

$$SO[O_1[E_1] \| O_2[E_2] \| O_3[E_3]]$$

Multi-SAND-NAND:

$$SO[O_1[A_1] \| O_2[A_2] \| O_3[A_3]]$$

Multi-NAND-NAND:

$$SO[SA[A_1 \| A_2 \| A_3] \rightarrow O_1[A_1] \| O_2[A_2] \| O_3[A_3]]$$

Each of the preceding formulas codes full information about the execution sequencing and may be regarded as a surrogate for a flowchart. Thus, a concise definition of each fundamental approach consists of its data flow diagram given in one of the Figs. 2–7 and of its execution sequence recipe coded in its preceding corresponding formula.

The following three-level approach is an interesting and plausible example that involves mixing, sequencing, and composing. Discipline 1 is aircraft performance, discipline 2 is aerodynamics, discipline 3 is structures, discipline 4 is the assembled structure, and disciplines 5–N are spars, ribs, stringers, skin, etc.:

$$SO[A_1 \rightarrow A_2 \rightarrow SO_3[A_4 \rightarrow O_5[A_5] \| \cdots \| O_N[A_N]]]$$

Discipline 3 is itself a system, and as with all systems, it is assumed that all design and coupling functions are computed within its disciplines (4–N). Thus, there are no f, g, nor y with "3" subscripts. There may, however, be system design variables associated with the structures system, x_3. Assume for now that disciplines 5–N are coupled only to discipline 4, and not to each other nor to disciplines 1 and 2. The optimization problem for the entire system is as follows.

Find:

$$f, x, x_1, x_2, y_{21}^*, y_{41}^*, y_{42}^*$$

Minimize:

$$f$$

Satisfy:

$$g_1 < 0, \qquad f_1 - f < 0$$

$$g_2 < 0, \qquad f_2 - f < 0$$

$$d_3 < 0$$

The optimization problem for the structures system (system or discipline 3) is as follows.

Find:

$$x_3, x_4, y_{54}^*, \ldots, y_{N4}^*, d_3$$

Minimize:

$$d_3$$

Satisfy:

$$g_4 < d_3, \qquad f_4 - f < d_3$$

$$y_{41} - y_{41}^* < d_3, \qquad y_{41}^* - y_{41} < d_3$$

$$y_{42} - y_{42}^* < d_3, \qquad y_{42}^* - y_{42} < d_3$$

$$d_4 < d_3, \ldots, \qquad d_N < d_3$$

The optimization problem for discipline i where i ranges from 5 to N is as follows.

Find:

$$x_i, d_i$$

Minimize:

$$d_i$$

Satisfy:

$$g_i < d_i, \qquad f_i - f < d_i$$

$$y_{i4} - y_{i4}^* < d_i, \qquad y_{i4}^* - y_{i4} < d_i$$

Implementation Options

A limited but necessary discussion of implementation options is given in this section. Different methods that may be implemented within evaluators, analyzers, and optimizers will be mentioned here to establish a background for the next section where computational effort is estimated. Since analyzers and optimizers may be gradient based, methods of calculating sensitivities are also described. It should be mentioned that implementation options also include the choice between serial and parallel computing that will be treated in the next section rather than here.

Disciplinary Analyzers and Evaluators

Disciplinary evaluators determine the residuals in the state equations, whereas disciplinary analyzers determine the values of the state variables that reduce the residuals to zero. Most existing engineering software packages are of the disciplinary analyzer variety. They typically exploit specialized techniques and models that have evolved within the discipline, and they may solve the state equations directly or iteratively.

The derivatives of outputs from disciplinary analyzers or evaluators will respect to inputs are referred to as disciplinary sensitivities. These sensitivities may always be computed by a finite difference approximation (FD); however, computational efficiency and accuracy may be gained by direct analytical sensitivity (DAS) based on implicit differentiation of the state equations.[22] This option has become available in an increasing number of analyzers. A recently introduced alternative is automatic differentiation.[23]

Disciplinary Optimizers

Disciplinary optimizers solve the disciplinary optimization problems of the various multilevel optimization approaches. These optimizers call disciplinary analyzers or evaluators to compute their constraints. In some disciplines, specialized techniques and models have been developed for efficient optimization (e.g., the optimality criteria methods of the structures discipline). In other disciplines, general purpose nonlinear programming algorithms may be used. Some disciplines might involve combinatorial or stochastic search if discrete or topological variables are present. Still in other disciplines, optimization may be quite informal, consisting of empirical rules and procedures of design practice.

The only function that a disciplinary optimizer returns to the system optimizer is its objective, the discrepancy function d_i. Calculation of the derivative of the optimal objective with respect to fixed parameters is known as optimum sensitivity analysis (OSA). It has been shown that when formal optimization algorithms are used in disciplinary optimization, OSA can be calculated directly from the Lagrange multipliers and disciplinary sensitivities.[24]

System Analyzer

The system analyzer determines the values of coupling variables that match their corresponding coupling functions as computed by disciplinary analyzers. There are two basic iterative techniques commonly used. The first is fixed-point iteration in which the coupling functions computed from one iteration are used as the coupling variables at the start of the next iteration. Iterations continue until the change in coupling variables is negligible. The second technique is Newton's method that employs not only the coupling functions but their derivatives with respect to the coupling variables at each iteration to accelerate convergence. Sequencing may be applied to either technique to eliminate some of the coupling variables altogether.

The derivatives of the converged coupling variables can be computed by solving a linear set of equations known as the global sensitivity equations (GSE).[25] The coefficients and right-hand-side vectors in the GSE consist of disciplinary sensitivities. The GSE method for system sensitivity analysis is an alternative to FD at the system level, just as DAS is an alternative to FD at the disciplinary level.

System Optimizer

The system optimizer solves the system optimization problem of the various approaches. This optimizer may call disciplinary optimizers, analyzers, and/or evaluators as well as a system analyzer. Therefore, it is imperative that the system optimizer calls for constraint evaluation as few times as possible. It is recommended that the system optimizer repeatedly constructs approximations of the constraints and optimizes the approximate problem. As this succession of approximate optimization problems proceeds, move limits on the variables should be adaptable to ensure robust convergence.

Two families of constraint approximations are possible. The first is the local approximation family in which the approximation is based solely on the exact values of the constraints and their gradients evaluated at the current design. A first-order approximation of the constraints can then be constructed. The second family is the global approximation family in which a few exact constraint evaluations are performed throughout optimization variable space, and the results are fitted with simple approximate functions.[26] Statistical methods

from the design of experiments are useful for selecting the best locations to perform exact constraint evaluation.

Computational Cost

The section presents an attempt at comparing the fundamental approaches introduced in the foregoing, based on their computational cost regarded as the primary concern to the user. However, not all user's concerns may be reduced to cost because if one interprets the disciplines as human organizations, the managerial advantages to be discussed later become another set of considerations important to users.

Computational Cost Measure

Computational cost may be measured in a number of ways, e.g., the CPU time, the elapsed real time (ET), the required memory, etc. This section presents cost evaluation of the fundamental approaches introduced in the foregoing, based only on their ET for computing, exclusive of the time needed for the data transmission and other overhead functions. It is difficult to provide a truly objective cost evaluation because the problem dependence of the assumptions and parameters involved taints any cost comparison of the foregoing approaches as being speculative. Furthermore, there are many variations of the fundamental approaches as well as many options for serial implementation (SI) and parallel implementation (PI).

To circumvent the aforementioned difficulties, the system at hand was examined at only two levels of complexity as measured by the number of interacting disciplines: a very simple system of two disciplines and one whose number of disciplines was increased by one order of magnitude (OOM) to 20. Four implementation options were investigated as given in Table 2.

For each of the implementation options, the ET was calculated at each of the two levels of complexity using formulas that link it to the parameters of the optimization and analysis problems. The parameters that define the analysis and optimization dimensionalities and affect ET are the lengths NX, NX_i, NY_{ij}, and NS_i of the previously defined variable vectors x, x_i, y_{ij}, and s_i. Additional parameters relative to the number of iterations required for the various iterative procedures will be defined later. These parameters were systematically varied in the ET calculations in a series of numerical experiments to obtain data for the cost comparison.

Evaluation of ET

The computation time estimates for a conventional SI were based on the assumption that the ET value is proportional to the number of the floating-point multiplications (FLMs). Omitting the proportionality coefficient, the ET is simply equated to the FLM count formulas commonly used in numerical methods.

The number of FLMs required to solve a nonsymmetric dense linear system of state equations of discipline i by a direct method is approximately $NS_i^3/3$. If the state equations are nonlinear, then the direct method may be embedded within an iterative scheme, and the number of FLMs required for analysis may be approximated as

$$CA_i = MA \bullet NS_i^3/3$$

where MO is the number of iterations. The preceding formula was used in the estimates of the ET given in this paper, even though it is realized that if the state equations are sparse, the CA_i may be reduced significantly to $CA_i = b \bullet MA \bullet NS_i^2$ where coefficient b depends on the degree of sparsity. Furthermore, if the state equations are sparse and solved iteratively rather than directly, CA_i may be reduced further to $CA_i = b \bullet MI \bullet NS_i$ where MI is the number of iterations.

Table 2 Implementation options

Option	SI	PI	FD	OSA	GSE	DAS	Designation
1	0	——	0	0	0	——	SI/FD
2	0	——	——	0	0	0	SI/DAS
3	——	0	0	——	——	——	PI/RD
4	——	0	——	0	0	0	PI/DAS

The number of FLMs required to evaluate the state equations (rather than solve them) was taken as the product of a matrix and a vector, and hence

$$CE_i = NS_i^2$$

The number of FLMs required to solve a dense linear optimization problem with NV optimization variables is approximately $1.5 \bullet NV^3$. The optimization variables in the various system and disciplinary optimization problems depend on the approach and may include design, coupling, and state variables. If nonlinear optimization problems are solved by solving a sequence of linearized problems, the number of FLMs may be approximated as

$$CO = MO \bullet 1.5 \bullet NV^3$$

where MA is the number of linearized problems solved. The preceding formula was used in the estimates of the ET given in this paper. In addition to this formula, each linearized problem requires a single sensitivity analysis (or evaluation).

The number of FLMs required for a single sensitivity analysis with respect to NV variables depends on the implementation option:
SI/FD:

$$CS = \Sigma_i[CA_i \bullet (1 + NV)]$$

SI/DAS:

$$CS = \Sigma_i(CA_i + CE_i \bullet NV)$$

PI/FD:

$$CS = \max_i (CA_i)$$

PI/DAS:

$$CS = C \max_i (CA_i + CE_i)$$

where the sums and maximums are taken over the disciplines. Note that for the DAS implementations it was assumed that sensitivities can be calculated cheaply via backsubstitutions after the nonlinear iterations are converged. It should be noted that in the DAS implementations, sensitivities with respect to any state variables are cost free since these sensitivities are simply the coefficients in the state equations themselves. Note that the parallel implementations assume an unlimited number of processors where each has enough memory to execute a disciplinary analysis. This is an assumption of a coarse-grained implementation on an "ultimate" machine that may either be a single multiprocessor computer or an equivalent computer network. It skirts the complex issues of the efficiency of parallel processing involved in the internal parallelization of an analysis code.

In the estimates given in this paper, it was assumed that for those approaches involving a system analyzer, the system analyzer was based on fixed-point iteration. The parameter that indicates the number of required fixed-point iterations is MC.

The number of FLMs required to triangularize the linear GSE equations is $NY^3/3$ where NY is the total number of coupling variables for all disciplines. The number of FLMs required to solve the GSE equations by backsubstitution with respect to NV optimization variables is $NV \bullet NY^2$.

The formulas used in this paper for the ET in terms of number of FLMs for the different approaches and implementation options are given in the Appendix.

Numerical Experiments

Experiments were conducted using the Appendix formulas to determine ET values and their dependence on selected parameters for the options from Table 2 and for tests cases described in this section.

For comparison and display purposes, the ET data are normalized by the ET of the most expensive analysis. The base 10 logarithm is then taken of the normalized ET to emphasize its OOM rather than the ET value itself. The differences in ET are considered meaningful only if they exceed at least one-half of an OOM. This is done to render the comparisons meaningful despite the fact that it is a function

of parameters that are very problem dependent and are, therefore, soft data subject to assumptions of the parameter values that may vary over many OOMs.

Two-Discipline Case

For the two-discipline system, the baseline values of the problem parameters are suggested by the case of minimum weight optimization for a flexible wing treated as a system of two disciplines: aerodynamics and structures. The aerodynamic analysis ($i = 1$) by a computational fluid dynamics (CFD) code might be expected to be much more expensive in terms of ET than the structural finite element analysis ($i = 2$), hence the setting of $NS_1 = 100,000$ and $NS_2 = 10,000$. The state variables for aerodynamics s_1 are the flow data at the grid points. For the structure they are the nodal displacements. The number of design variables x that affect directly both disciplines is set at $NX = 10$ to represent the overall wing shape geometry. The number of design variables x_1 that affect directly the aerodynamics only is $NX_1 = 10$ to control the shape of those parts of the wing that are structurally inert (outside of the structural box), for instance, the leading-edge radius. The variables x_2 are cross-sectional gauges that directly affect the structures only. Typically, their number is larger than NX and NX_1, hence, $NX_2 = 100$. The coupling variables y_{12} and y_{21} are aerodynamic loads and structural displacements, respectively. Using a modal representation of both, their number may be kept reasonably low, e.g., $NY_{12} = NY_{21} = 50$.

Based on the experience[27] one assumes $MC = 5$. This low number is reasonable only if the wing structure had been initialized to be sufficiently stiff; otherwise MC might be very large or the iteration may not even converge. The implication is that the discipline of structures would do a certain amount of prerequisite design work, assuming a constant magnitude and distribution of aerodynamic loads, before the coupled problem optimization begins. Similarly it is known from experience[28] that MO may range from a low value of less than 10–40, and it is, therefore, set at $MO = 30$. An assumed, nominal value for $MA = 5$ was used.

Twenty-Discipline Case

A hypothetical system of 20 disciplines was assumed as an abstract example, not rooted in any specific application but intended to shed light on the use of the fundamental approaches in a large engineering system. The parameter values were taken as $NX = 10$, $NX_i = 100$, and $NS_i = 10,000$ for all disciplines i, and $NY_{ij} = 5$ for all discipline $i - j$ pairs. These data defined variant 1 of this case that will be referred to as "analysis intensive" because the number of the state variables is so much larger than the number of the design variables. In variant 2 of the same case, referred to as "design intensive," the setting of $NX_i = NS_i = 1,000$ was used.

Experimental Results

The ET data are displayed by means of line graphs in Figs. 8–10, which are constructed as follows. The vertical scale is in log 10 of the normalized ET to illustrate the ET OOM as a multiple of that disciplinary analysis that is the most expensive one in the system at hand. To facilitate comparisons, the vertical scale is the same in all three figures. Each of the fundamental approaches is represented in the graph by one line, labeled with an approach acronym consisting of the first letters from its first, middle, and last names. For comparison convenience, the line for each approach links the discrete data points corresponding to the implementation options from Table 2. Note that options 3 and 4 from Table 2 are merged in the figures because it turned out that PI nearly obliterated the difference in ET between FD and DAS in all the tests.

Scanning the graphs by eye conveys a lot of information. For example, the left-to-right scan of a particular line shows how sensitive the corresponding fundamental approach is to the implementation options, including the choice between the serial (SI) and parallel (PI) computing and the choice between FD and DAS. The vertical scan reveals the differences between the fundamental approaches, accentuated by the use of solid and empty symbols for single-level and multilevel optimization approaches, respectively. Finally, simultaneous scanning of the graphs in Figs. 9–10 gives an idea about the

trends generated by changing the dimensionality and nature of the problem.

One must be cautious in drawing conclusions from the data in Figs. 8–10 about the relative cost advantages for each fundamental approach. For example, the ET reductions for FD from switching to PI from SI are shown by comparing the rightmost and the leftmost data point columns in these figures, and the corresponding benefit for DAS is illustrated by comparing the rightmost and the middle data point columns in these figures. These benefits must be tempered by the realization that they are, partially at least, due to the unlimited machine assumption. In comparing the ET cost of FD and DAS, one should also remember the accuracy advantages of DAS over FD. Finally, one needs to bear in mind that the PI considered herein is coarse grained, hence not representative of the full potential of parallel computing. This assumption is, at least to a degree, the cause of no ET reduction due to PI in S-S-S. Finally, a reminder is in order that ET differences below 0.5 OOM are not significant in the comparisons.
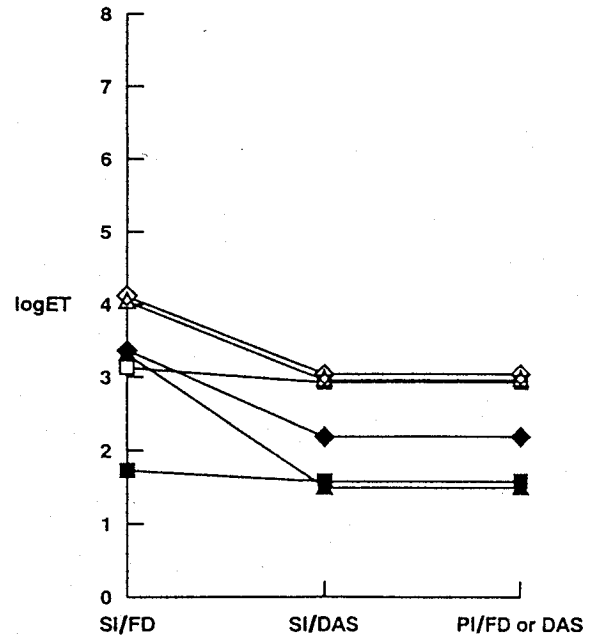
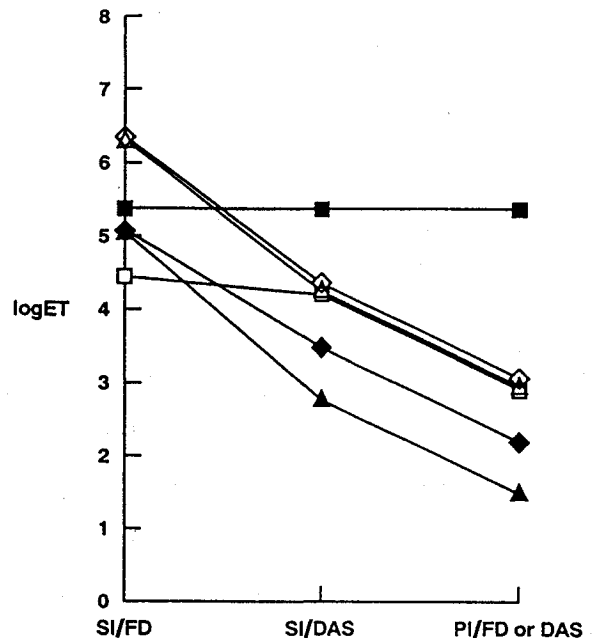Fig. 8   Two-discipline case: —■—, S-S-S; —▲—, S-S-N; —◆—, S-N-N; —□—, M-S-S; —△—, M-S-N; and —◇—, M-N-N.

Fig. 9   Twenty-discipline case, analysis intensive: —■—, S-S-S; —▲—, S-S-N; —◆—, S-N-N; —□—, M-S-S; —△—, M-S-N; and —◇—, M-N-N.
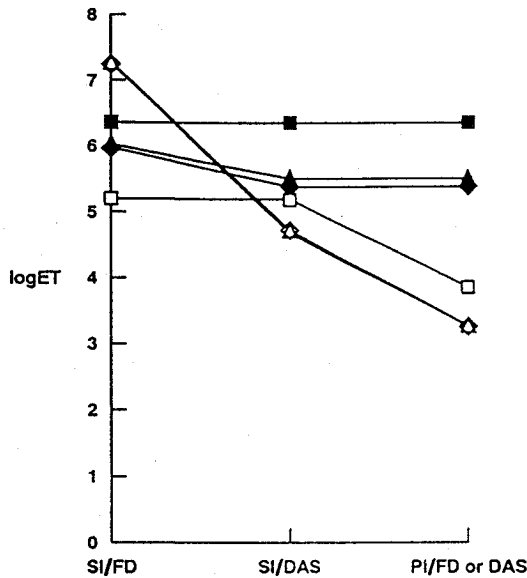
Fig. 10 Twenty-discipline case, design intensive: ─■─, S-S-S; ─▲─, S-S-N; ─◆─, S-N-N; ─□─, M-S-S; ─△─, M-S-N; and ─◇─, M-N-N.

The assumed values for the iteration parameters MA, MC, and MO were increased, one at a time, by a factor of 10 to assess the effect on the relative ranking of the approaches. Graphs similar to Figs. 8–10 were produced. Increasing MA tended to reduce ET in the SAND-SAND approaches relative to the others. Increasing MC increased ET in the SAND-NAND approaches, whereas increasing MO increased ET in the multilevel approaches. The FLM count formulas for analysis and optimization were also reduced to account for sparsity. This tended to reduce the SAND-SAND ET relative to the others.

Managerial Considerations in the Choice of Approach

These considerations may be as important as the ET comparisons. To develop a broad work front of people and machines, an engineering organization is customarily divided into groups that correlate with the disciplines in the fundamental approaches discussed herein. These groups are intended to work concurrently. The resulting compression of the project calendar time is the same motivation that underlies the development of concurrent processing in computer technology.

Clearly, the need to enable the human organization groups to work concurrently is supported by the multilevel optimization approaches. These approaches have an additional advantage because they allow disciplinary specialists to use their own methods, including informal ones, that have evolved over time, to solve their analysis and optimization problems autonomously. This includes the determination within disciplines of the values of disciplinary design variables and functions and may even extend to choices among discrete design alternatives. This is beneficial not only to the pace and quality of work but also to the human motivation.

Additional consideration is the dependence, at least in short to midterm, on the existing analysis and optimization software. The same approaches that favor autonomy of the disciplines also preserve usability of that software.

Finally, the real life budgetary and time limits rather than mathematical conditions often become de facto termination criteria. This favors the NAND approaches since they produce a succession of improving designs with meaningful analysis results available for each. In contrast, the SAND approaches produce a usable result only at the very end of their execution.

Concluding Remarks

The paper identified six fundamental approaches, named by three-part names referring to decomposition into levels and treatment of the variables: 1) single-level vs multilevel optimization (first name

single vs multi), 2) system-level simultaneous analysis and design vs analysis nested in optimization (middle names SAND vs NAND), and 3) discipline-level simultaneous analysis and design vs analysis nested in optimization (last name SAND vs NAND). A compact notation was introduced for these approaches to define concisely the multitude of variations that may be developed by mixing, sequencing, and composing the approaches. Moreover, each approach may be implemented in a number of options whose defining features are techniques for the gradient computation and serial or parallel computing. The approaches that have previously been identified in the literature as limited to hierarchic systems have been shown to be applicable to nonhierarchic systems.

The elapsed time (ET) was estimated for different implementation options and test cases. Certain conclusions become apparent from these limited results. First, no single approach is fastest for all test cases and implementation options. Conversely, no single approach can be identified as being always the slowest. Generally, the single-level optimization approaches are fastest on analysis-intensive problems, whereas the multilevel optimization approaches are fastest on design-intensive problems. This suggests mixing approaches by employing disciplinary optimizers only in disciplines with large numbers of design variables. Second, the choice of approach affects the ET as much or more than the choice of implementation (SI vs PI, FD vs DAS). Furthermore, the amount of savings in ET gained by going from SI to PI or from FD to DAS depends on the approach. In some cases the savings may be virtually nonexistent, whereas in others they may be dramatic.

These observations point to the recommendation that for a given real-world multidisciplinary problem that is expected to consume considerable resources, one should carefully consider the choice of approach. It would be wise to estimate the computational effort for alternative approaches in much the same way as was illustrated for the examples given in this paper and include memory requirements specific to the problem at hand. The managerial benefits of multilevel optimization approaches should also be considered. The corollary to the aforementioned is that if general purpose system development is attempted to support multidisciplinary optimization in engineering design, that system should be flexible enough to offer the users the choice among approaches.

Since the proper selection of approach is vital to the efficient solution of MDO problems, research is needed in the following areas for a better support of this selection process.

1) The fundamental approaches should be numerically demonstrated and compared on the same nonhierarchic multidisciplinary example.

2) The optimality equivalence of the fundamental approaches should be verified with mathematical rigor.

3) Formulas and parameter values used to estimate computational effort should be refined and tested on numerical results. They should also include the memory requirements.

4) Multilevel optimization approaches should be extended to allow informal and discrete methods for increased disciplinary flexibility.

Appendix: Elapsed Time

Estimation Formulas

$$NY_i = \Sigma_{j \neq i} NY_{ji}$$

= number of coupling variables input to discipline i

$$CA_i = MA \bullet NS_i^3 / 3$$

$$CE_i = NS_i^2$$

Single-SAND-SAND

$$CO = MO \bullet 1.5 \bullet [NX + \Sigma_i (NX_i + NS_i + NY_i)]^3$$

SI/FD:

$$\text{ET} = MO \bullet \Sigma_i [CE_i \bullet (1 + NX + NX_i + NS_i + NY_i)] + CO$$

SI/DAS:

$$ET = MO \bullet \Sigma_i[CE_i \bullet (1 + NX + NX_i + NY_i)] + CO$$

PI/FD:

$$ET = MO \bullet \max_i(CE_i) + CO$$

PI/DAS:

$$ET = MO \bullet \max_i(CE_i) + CO$$

Single-SAND-NAND

$$CO = MO \bullet 1.5 \bullet [NX + \Sigma_i(NX_i + NY_i)]^3$$

SI/FD:

$$ET = MO \bullet \Sigma_i[CA_i \bullet (1 + NX + NX_i + NY_i)] + CO$$

SI/DAS:

$$ET = MO \bullet \Sigma_i[CA_i + CE_i \bullet (NX + NX_i + NY_i)] + CO$$

PI/FD:

$$ET = MO \bullet \max_i(CA_i) + CO$$

PI/DAS:

$$ET = MO \bullet \max_i(CA_i + CE_i) + CO$$

Single-NAND-NAND

$$CO = MO \bullet 1.5 \bullet (NX + \Sigma_i NX_i)^3$$

$$CGSE = (\Sigma_i NY_i)^3 / 3 + (NX + \Sigma_i NX_i) \bullet (\Sigma_i NY_i)^2$$

SI/FD:
$$ET = MO \bullet \{\Sigma_i[MC \bullet CA_i + CA_i \bullet (NX + NX_i + NY_i)]$$
$$+ CGSE\} + CO$$

SI/DAS:
$$ET = MO \bullet \{\Sigma[MC \bullet CA_i + CE_i \bullet (NX + NX_i + NY_i)]$$
$$+ CGSE\} + CO$$

PI/FD:

$$ET = MO \bullet \max_i(MC \bullet CA_i) + CO$$

PI/DAS:

$$ET = MO \bullet \left[\max_i(MC \bullet CA_i) + \max_i(CE_i) + CGSE \right] + CO$$

Multi-SAND-SAND

$$CO = MO \bullet 1.5 \bullet (NX + \Sigma_i NY_i)^3$$

$$CO_i = MO \bullet 1.5 \bullet (NX_i + NS_i)^3$$

SI/FD:
$$ET = MO \bullet \Sigma_i[MO \bullet CE_i \bullet (1 + NX_i + NS_i) + CO_i$$
$$+ CE_i \bullet (NX + NY_i)] + CO$$

SI/DAS:
$$ET = MO \bullet \Sigma_i[MO \bullet CE_i \bullet (1 + NX_i) + CO_i$$
$$+ CE_i \bullet (NX + NY_i)] + CO$$

PI/FD:

$$ET = MO \bullet \max_i(MO \bullet CE_i + CO_i) + CO$$

PI/DAS:

$$ET = MO \bullet \max_i(MO \bullet CE)_i + CO_i + CE_i) + CO$$

Multi-SAND-NAND

$$CO = MO \bullet 1.5 \bullet (NX + \Sigma_i NY_i)^3$$

$$CO_i = MO \bullet 1.5 \bullet (NX_i)^3$$

SI/FD:

$$ET = MO \bullet \Sigma_i[MO \bullet CA_i \bullet (1 + NX_i) + CO_i$$
$$+ CA_i \bullet (NX + NY_i)] + CO$$

SI/DAS:

$$ET = MO \bullet \Sigma_i[MO \bullet (CA_i + CE_i \bullet NX_i) + CO_i$$
$$+ CE_i \bullet (NX + NY_i)] + CO$$

PI/FD:

$$ET = MO \bullet \max_i(MO \bullet CA_i + CO_i) + CO$$

PI/DAS:

$$ET = MO \bullet \max_i[MO \bullet (CA_i + CE_i) + CO_i + CE_i] + CO$$

Multi-NAND-NAND

$$CO = MO \bullet 1.5 \bullet (NX + \Sigma_i \Sigma_i 1)^3$$

$$CO_i = MO \bullet 1.5 \bullet (NX_i)^3$$

$$CGSE = (\Sigma_i NY_i)^3 / 3 + (NX + \Sigma_i NX_i) \bullet (\Sigma_i NY_i)^2$$

SI/FD:
$$ET = MO \bullet \{\Sigma_i[MC \bullet CA_i + CA_i \bullet (NX + NX_i + NY_i)]$$
$$+ CGSE + \Sigma_i[MO \bullet CA_i \bullet (1 + NX_{ii})$$
$$+ CO_i + CA_i \bullet NX]\} + CO$$

SI/DAS:
$$ET = MO \bullet \{\Sigma_i[MC \bullet CA_i + CE_i \bullet (NX + NX_i + NY_i)]$$
$$+ CGSE + \Sigma_i[MO \bullet (CA_i CE_i \bullet NX_i)$$
$$+ CO_i + CE_i \bullet NX]\} + CO$$

PI/FD:

$$ET = MO \bullet \left[\max_i(MC \bullet CA_i) + \max_i(MO \bullet CA_i + CO_i) \right] + CO$$

PI/DAS:

$$ET = MO \bullet \left\{ \max_i(MC \bullet CA_i) + \max_i(CE_i) + CGSE \right.$$
$$\left. + \max_i[MO \bullet (CA_i + CE_i) + CO_i + CE_i] \right\} + CO$$

References

[1] Anon., "Current State of the Art on Multidisciplinary Design Optimization (MDO)—An AIAA White Paper," AIAA, Washington, DC, 1991.

[2] Sobieszczanski-Sobieski, J., and Tulinius, J., "MDO Can Help Resolve the Designer's Dilemma," *Aerospace America*, Sept. 1991, pp. 32–35.

[3]Cramer, E. J., Dennis, J. E., Jr., Frank, P. D., Lewis, R. M., and Shubin, G. R., "Problem Formulation for Multidisciplinary Optimization," Center for Research on Parallel Computation, Rice Univ., Rept. CRPC-TR93334, Houston, TX; see also *SIAM Journal on Applied Mathematics* (to be published).

[4]Sobieszczanski-Sobieski, J., "Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design: Recent Advances and Results," *Journal of Aircraft*, Vol. 27, No. 12, 1990, pp. 993–1001.

[5]Haftka, R. T., "Simultaneous Analysis and Design," *AIAA Journal*, Vol. 23, No. 7, 1985, pp. 1099–1103.

[6]Ta'asan, S., Kuruvila, G., and Salas, M. D., "Aerodynamic Design and Optimization in One Shot," AIAA Paper 92-0025, Jan. 1992.

[7]Hou, G. W., Taylor, A. C., Mani, S. V., and Newman, P. A., "Formulation for Simultaneous Aerodynamic Analysis and Design Optimization," 2nd U.S. National Congress on Computational Mechanics, Washington, DC, Aug. 1993.

[8]Haftka, R. T., Sobieszczanski-Sobieski, J., and Padula, S. L., "On Options for Interdisciplinary Analysis and Design Optimization," *Structural Optimization*, Vol. 4, No. 2, 1992, pp. 65–74.

[9]Grossman, B., Haftka, R. T., Kao, P.-J., Polen, D. M., Rais-Rohani, M., and Sobieszczanski-Sobieski, J., "Integrated Aerodynamic Structural Design of a Transport Wing," AIAA Paper 89-2129, Aug.–Sept. 1989.

[10]Hajela, P., Bloebaum, C. L., and Sobieszczanski-Sobieski, J., "Application of Global Sensitivity Equations in Multidisciplinary Aircraft Synthesis," *Journal of Aircraft*, Vol. 27, No. 12, 1990, pp. 1002–1010.

[11]Sobieszczanski-Sobieski, J., Bloebaum, C. L., and Hajela, P., "Sensitivity of Control-Augmented Structure Obtained by a System Decomposition Method," *AIAA Journal*, Vol. 29, No. 2, 1991, pp. 264–270.

[12]Balling, R. J., and Sobieszczanski-Sobieski, J., "An Algorithm for Solving the System-Level Problem in Multilevel Optimization," AIAA Paper 94-4333, Sept. 1994.

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

[14]Wrenn, G. A., and Dovi, A. R., "Multilevel Decomposition Approach to the Preliminary Sizing of a Transport Aircraft Wing," *Journal of Aircraft*, Vol. 25, No. 7, 1988, pp. 632–638.

[15]Walsh, J. L., Young, K. C., Pritchard, J. I., Adelman, H. M., and Mantay, W., "Multilevel Decomposition Approach to Integrated Aerodynamic/Dynamics/Structural Optimization of Helicopter Rotor Blades," AHS Aerodynamics Specialists Conf., San Francisco, CA, Jan. 1994.

[16]Schmit, L. A., and Ramanathan, R. K., "Multilevel Approach to Minimum Weight Design Including Buckling Constraints," *AIAA Journal*, Vol. 16, 1973, pp. 97–104.

[17]Sobieszczanski-Sobieski, J., "Two Alternative Ways for Solving the Coordination Problem in Multilevel Optimization," *Structural Optimization*, Vol. 6, 1993, pp. 205–215.

[18]Kreisselmeier, G., and Steinhauser, R., "Application of Vector Performance Optimization to a Robust Control Loop Design for a Fighter Aircraft," *International Journal of Control*, Vol. 37, 1983, pp. 251–284.

[19]Sobieszczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems," *Recent Advances in Multidisciplinary Analysis and Optimization*, NASA CP-3301, Pt. 1, 1988.

[20]Bloebaum, C. L., "Formal and Heuristic System Decomposition Methods in Multidisciplinary Synthesis," NASA-CR-4413, Dec. 1991.

[21]Eason, E., Nystrom, G., Burlingham, A., and Nelson, E., "Nonhierarchic Multidisciplinary Optimal Design of a Commercial Aircraft," AIAA/USAF/NASA/ISSMO 5th Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, Sept. 1994.

[22]Adelman, H. A., and Haftka, R. T., "Sensitivity Analysis of Discrete Structural Systems," *AIAA Journal*, Vol. 24, No. 5, 1986, pp. 823–832.

[23]Bischof, C., Carle, A., Corliss, G., Griewank, A., and Hovland, P., "ADIFOR—Generating Derivative Codes from Fortran Programs," *Scientific Programming*, Vol. 1, No. 1, 1992, pp. 1–29.

[24]Barthelemy, J.-F. M., and Sobieszczanski-Sobieski, J., "Optimum Sensitivity Derivatives of Objective Functions in Nonlinear Programming," *AIAA Journal*, Vol. 21, No. 6, 1983, pp. 913–915.

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

[26]Box, G. E. P., and Draper, N. R., *Empirical Model-Building and Response Surface*, Wiley, New York, 1987.

[27]Sobieszczanski, J., McCullers, L. A., Ricketts, R. H., Santoro, N. J., Beskenis, S. D., and Kzertze, W. L., "Structural Design Studies of a Supersonic Cruise Arrow Wing Configuration," NASA CP-001, Pt. 2, Nov. 1976, pp. 659–683.

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