



AIAA 2000-4718

**Analytical and Computational Properties
of Distributed Approaches to MDO**

N. M. Alexandrov and R. M. Lewis (ICASE)

NASA Langley Research Center
Hampton, VA 23681

**8th AIAA/USAF/NASA/ISSMO Symposium
on Multidisciplinary Analysis & Optimization
6-8 September 2000 / Long Beach, CA**

ANALYTICAL AND COMPUTATIONAL PROPERTIES OF DISTRIBUTED APPROACHES TO MDO

NATALIA M. ALEXANDROV* AND ROBERT MICHAEL LEWIS†

Abstract

Historical evolution of engineering disciplines and the complexity of the MDO problem suggest that disciplinary autonomy is a desirable goal in formulating and solving MDO problems. We examine the notion of disciplinary autonomy and discuss the analytical properties of three approaches to formulating and solving MDO problems that achieve varying degrees of autonomy by distributing the problem along disciplinary lines. Two of the approaches—Optimization by Linear Decomposition and Collaborative Optimization—are based on bilevel optimization and reflect what we call a structural perspective. The third approach, Distributed Analysis Optimization, is a single-level approach that arises from what we call an algorithmic perspective. The main conclusion of the paper is that disciplinary autonomy may come at a price: in the bilevel approaches, the system-level constraints introduced to relax the interdisciplinary coupling and enable disciplinary autonomy can cause analytical and computational difficulties for optimization algorithms. The single-level alternative we discuss affords a more limited degree of autonomy than that of the bilevel approaches, but without the computational difficulties of the bilevel methods.

Key Words: Autonomy, bilevel optimization, distributed optimization, multidisciplinary optimization, multilevel optimization, nonlinear programming, problem integration, system synthesis

Introduction

We examine the mathematical statement of multidisciplinary optimization (MDO) problems or, more specifically, the *formulation of MDO problems as optimization problems*, and the consequences of problem formulation for the practical solution of the resulting computational problem by optimization algorithms. This distinction between problem formulations and algorithms used for solving the resulting optimization problems underlies

the work. For the purposes of this paper we view MDO as comprising a subset of the broader design problem that can be expressed as a nonlinear programming problem.

MDO problems are distinguished by the complexity of the constituent disciplinary analyses. Due to the required specialized knowledge, the disciplines have evolved into virtually autonomous subjects, and have developed independently, in large measure. Justifiably, significant research effort has been and continues to be expended on disciplinary modeling and solution techniques and, in many cases (e.g., structural optimization), on the disciplinary optimization methods. Analysis codes usually do take up most of the computational effort. However, the multidisciplinary synthesis scheme, realized via the MDO problem formulation, is equally important. It influences, and may determine, not only how many times the expensive analysis codes will have to be executed, but also whether the resulting problem can be efficiently implemented and solved—or solved at all—by available optimization algorithms. Therefore, optimization problem formulation holds profound computational consequences for the overall MDO process. The present work continues the effort in [1, 2] aimed at furthering the understanding of the analytical properties and computational implications of MDO problem formulations and at proposing efficient solution methods based on this understanding.

Because of the complexity and expense of the constituent analyses, most efforts in dealing with systematic MDO problem formulation focus on methods that aim at affording the user the maximum disciplinary autonomy. In this paper, we examine some notions of autonomy and consider three broad classes of MDO problem formulations—two bilevel optimization formulations and one single-level, distributed analysis formulation—in light of disciplinary autonomy, as well as the consequences of the techniques used for attaining autonomy via distributing the disciplinary subproblems.

The recurring theme of this and related papers [2–4] is the strong influence of the analytical features of problem formulation on the ability of nonlinear programming

*Member AIAA. NASA Langley Research Center, Hampton, VA, n.alexandrov@larc.nasa.gov.

†Member AIAA. ICASE, NASA Langley Research Center, Hampton, VA, buckaroo@icase.edu. This author's research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA.

Copyright ©2000 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 Government Purposes. All other rights are reserved by the copyright owner.

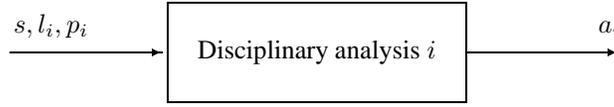


Figure 1: Disciplinary analysis in isolation

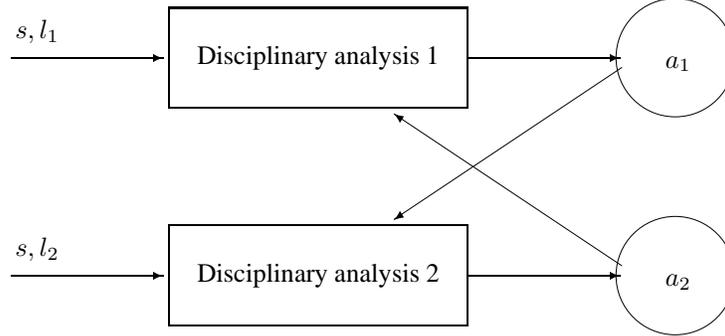


Figure 2: Flow of information in multidisciplinary analysis

algorithms to solve the problem reliably and efficiently. The specific conclusion of this work is that disciplinary autonomy may come at a price: In the bilevel approaches we discuss, the system-level constraints introduced to relax the interdisciplinary coupling and enable disciplinary autonomy can cause analytical and computational difficulties for optimization algorithms. The single-level alternative, Distributed Analysis Optimization (DAO), affords a more limited degree of autonomy than do the bilevel approaches, but without the computational difficulties of the bilevel methods. Details of the implementation for this formulation may be found in the companion paper [4].

The two-discipline model problem

For simplicity, we present our discussion for a two-discipline model problem. The disciplines might represent the aeroelastic interaction between aerodynamics (Discipline 1) and structural analysis (Discipline 2) for a wing in steady-state flow. Our description of the model problem closely follows that in [2, 3] because we wish to maintain a uniform notation in the description of MDO problem formulations.

Problem components

We assume that each disciplinary subsystem is based on a disciplinary analysis, depicted as the input-output relation in Fig. 1.

Each discipline takes as its input a set of design variables (s, l_i) and parameters p_i , and produces a set of analysis outputs a_i . The *system-level design variables* s are

those *shared* by both disciplines. The *disciplinary design variables* l_1 and l_2 are *local* to Disciplines 1 and 2, respectively. We use a_i to represent the totality of outputs from a given discipline. These outputs include all data that are passed to the other discipline as parameters and, possibly, quantities passed to design constraints and objectives. Parameters p_i are derived from the analysis outputs a_j , $j \neq i$, of the other discipline, and are not directly manipulated by the designer in Discipline i . In our aeroelastic example, for instance, the input p_1 from structures to aerodynamics would include the wing shape, while the input p_2 from aerodynamics to structures would include the aerodynamic loads.

The disciplinary input-output relations have the functional form

$$a_i = A_i(s, l_i, p_i).$$

The disciplinary analyses A_1 and A_2 are assumed to be independently solvable. That is, we assume that, given appropriate values of inputs (s, l_1, p_1) to Discipline 1, we can compute the disciplinary output a_1 via the Discipline 1 analysis

$$a_1 = A_1(s, l_1, p_1).$$

(By “appropriate” we mean input values for which the analysis is defined.) Continuing with our aeroelastic illustration, given values p_1 for the shape of the wing, we can compute the flow a_1 around it. Likewise, given appropriate values of inputs (s, l_2, p_2) to Discipline 2, we assume we can compute the disciplinary outputs a_2 as

$$a_2 = A_2(s, l_2, p_2).$$

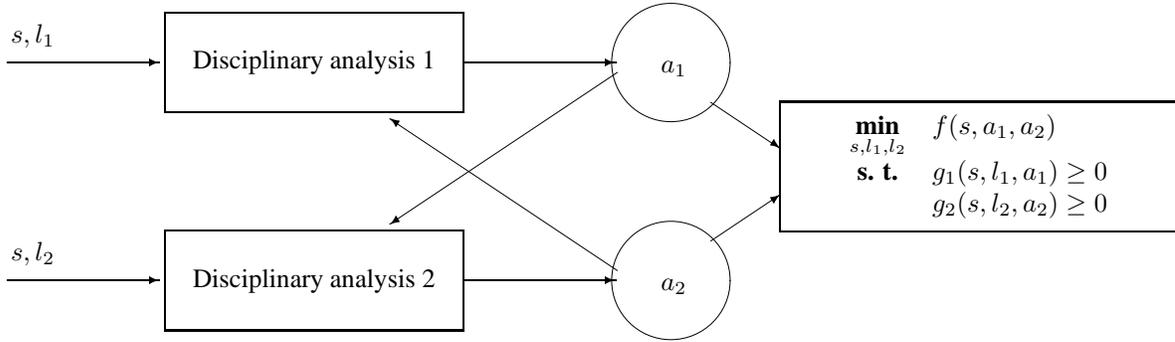


Figure 3: Flow of information in multidisciplinary optimization

In our example, given values of the aerodynamic loads p_2 , we can compute the structural response a_2 .

Multidisciplinary analysis

The coupled multidisciplinary analysis system (MDA) reflects the physical requirement that a solution simultaneously satisfy the two disciplinary analyses. The input parameters p_i to each discipline are now required to correspond to some (or all) of the outputs a_j from the other disciplinary analysis. This is depicted in Fig. 2.

We write the multidisciplinary analysis system as a simultaneous system of equations. Given (s, l_1, l_2) , we have

$$a_1 = A_1(s, l_1, a_2) \quad (1)$$

$$a_2 = A_2(s, l_2, a_1). \quad (2)$$

Solving the first equation results in the analysis outputs a_1 of Discipline 1, and solving the second equation produces the analysis outputs a_2 of Discipline 2. The multidisciplinary analysis thus implicitly defines a_1 and a_2 as functions of (s, l_1, l_2) :

$$a_1 = a_1(s, l_1, l_2), \quad a_2 = a_2(s, l_1, l_2).$$

Solving the coupled equations (1)–(2) leads to a full multidisciplinary analysis, in which the coupled disciplines give a physically consistent (and thus meaningful) result. The disciplinary responses a_i describe part of the behavior of the system. Again, if Discipline 1 represents aerodynamic analysis of the flow around a wing and Discipline 2 represents structural analysis of the wing, the MDA reflects the interaction between the flow field, which affects the shape of the wing, and the shape of the wing, which affects the flow field.

A fully integrated formulation

We now couple the two disciplines in connection with a design optimization problem. Given the need to satisfy

the MDA at a solution, the most natural formulation, arguably, is to impose an optimizer over the MDA. We will use this formulation to represent the original problem, i.e., the problem one ideally wishes to solve. The flow of information in this formulation is depicted in Fig. 3. Its mathematical statement is

$$\begin{aligned} \min_{s, l_1, l_2} & f(s, a_1(s, l_1, l_2), a_2(s, l_1, l_2)) \\ \text{s. t.} & g_1(s, l_1, a_1(s, l_1, l_2)) \geq 0 \\ & g_2(s, l_2, a_2(s, l_1, l_2)) \geq 0, \end{aligned} \quad (3)$$

where, given (s, l_1, l_2) , we solve the multidisciplinary analysis system (1)–(2) for the disciplinary analysis outputs $a_1(s, l_1, l_2)$ and $a_2(s, l_1, l_2)$. The function f represents the system-level objective.

To facilitate the discussion of one of the distributed optimization approaches (collaborative optimization), we have chosen a simplified model problem: each of the constraints g_i explicitly depends only on a single discipline's analysis outputs. There is no constraint that involves a_1 and a_2 jointly. The constraints g_1, g_2 are thus *disciplinary design constraints* associated solely with Disciplines 1 and 2, respectively. This choice of design constraints simplifies the exposition, but is not essential. A complete description of collaborative optimization, without the simplifying assumption, is given in [2].

Two perspectives on problem formulation

Most approaches to formulating MDO problems are motivated by the wish for computational autonomy of the disciplinary subsystems, by the need for computational efficiency, and by a desire to simplify problem synthesis.

The philosophy we call the *structural perspective* starts with the multidisciplinary system description and poses an optimization problem that is thought to correspond to certain physical or organizational characteristics of the problem. For instance, many approaches to solving MDO problems assume that the problem is to be decomposed along disciplinary lines and that it is desirable to

treat the disciplinary subproblems with as much autonomy as possible, coordinating the solutions of the subsystem problems in some manner that will result in a solution to the overall system design problem. Solution of the resulting system-level problem is then attempted by using available optimization software. In the structural perspective, then, the physical structure of the problem and organizational considerations are the main driving forces behind the choice of problem formulation.

Optimization by Linear Decomposition (OLD), or “hierarchical decomposition” [5–12], reflects the structural perspective, as does the approach that has recently received attention under the name Collaborative Optimization (CO) [13–16]. The underlying idea of Collaborative Optimization—the notion of decoupling the disciplines by introducing interdisciplinary consistency constraints, while minimizing a measure of interdisciplinary inconsistency in disciplinary subproblems—appeared previously in [17–19] and [10–12, 20, 21]. In OLD, the disciplines are given the autonomous task of minimizing disciplinary design infeasibility (the lower-level disciplinary problem) while maintaining system-level consistency. The system-level problem is to drive the design infeasibility to zero. In CO, the situation is reversed: the disciplines are given the autonomous task of minimizing system-level inconsistency while maintaining disciplinary design feasibility. In both approaches, optimization of the system-level objective, subject to interdisciplinary consistency, is performed in the system-level problem.

Multilevel approaches lead to multilevel nonlinear programs, which are well known to be difficult to solve [22]. This is a drawback of the purely structural perspective. In contrast, the *algorithmic perspective* takes as its starting point the abilities (and inabilities) of optimization algorithms, and seeks to formulate the MDO problem so that the resulting optimization problem can be solved reliably and efficiently by conventional optimization techniques. The problem is stated as the most general NLP, examined for structure and gradually reformulated to conform to organizational features as much as possible without sacrificing solubility by available algorithms. This approach is exemplified, for instance, in [23, 24].

These two approaches to formulating MDO problems are obviously interrelated, and the distinction is arguably subjective since it is rooted in the methodology used. However, as we discuss, one can run into trouble if one ignores the analytical and computational nature of the resulting optimization problem.

In the remainder of the paper, we contrast OLD and CO with DAO as representatives of these two perspectives. DAO is treated in more detail in a companion paper [4].

Formulations reflecting the structural perspective

In this section we give reformulations of our two-discipline model problem in terms of OLD and CO. One of our aims is to give a careful presentation with complete notation indicating the exact functional dependence of the various quantities (e.g., system-level consistency conditions) on the local and system-level design variables. Detail in notation helps clarify some of the salient characteristics of these problem formulations.

Collaborative Optimization

Again, in the interests of unified notation, the description of CO closely follows that in [2, 3]. To reformulate (3) along the lines of CO, we introduce new disciplinary design variables σ_1, σ_2 that relax the coupling between the subsystems through the shared system design variables s . The variables σ_i serve as local copies (at the level of the disciplinary subproblems) of the shared variables s . In general, Greek letters will denote new, auxiliary variables designed to serve at the subproblem level as copies of shared quantities.

CO is a bilevel approach in which a system-level coordination problem attempts to optimize the system-level objective resulting in the following system-level problem:

$$\begin{aligned} \min_{s, t_1, t_2} \quad & f(s, t_1, t_2) \\ \mathbf{s. t.} \quad & C(s, t_1, t_2) = 0, \end{aligned} \quad (4)$$

where there are N *interdisciplinary consistency constraints* $C = \{c_1, \dots, c_N\}$ which we describe presently. The system-level problem controls the system-level design variables s and interdisciplinary coupling variables (t_1, t_2) , which are system-level target values for the disciplinary inputs and outputs a_1 and a_2 .

The system-level problem issues design targets (s, t_1, t_2) to the constituent disciplines. In the lower-level problems, the disciplines design to match these targets, as follows. In Discipline 1, we are given (s, t_1, t_2) and compute $\bar{\sigma}_1(s, t_1, t_2)$ and $\bar{l}_1(s, t_1, t_2)$ as solutions of the following minimization problem in (σ_1, l_1) :

$$\begin{aligned} \min_{\sigma_1, l_1} \quad & \frac{1}{2} [\| \sigma_1 - s \|^2 + \| a_1(\sigma_1, l_1, t_2) - t_1 \|^2] \\ \mathbf{s. t.} \quad & g_1(\sigma_1, l_1, a_1(\sigma_1, l_1, t_2)) \geq 0, \end{aligned} \quad (5)$$

where a_1 is computed in this disciplinary optimization problem via the disciplinary analysis

$$a_1 = A_1(\sigma_1, l_1, t_2).$$

In general, we use overbars (e.g., $\bar{\sigma}_1, \bar{l}_1$) to indicate optimal solutions of subsystem problems as a function of system-level variables. In the disciplinary subproblem (5), the system-level variables (s, t_1, t_2) serve either as

parameters or targets that we try to match. An analogous problem for Discipline 2 defines solutions $\bar{\sigma}_2(s, t_1, t_2)$ and $\bar{l}_2(s, t_1, t_2)$ of the problem

$$\begin{aligned} \min_{\sigma_2, l_2} \quad & \frac{1}{2} [\| \sigma_2 - s \|^2 + \| a_2(\sigma_2, l_2, t_1) - t_2 \|^2] \\ \text{s. t.} \quad & g_2(\sigma_2, l_2, a_2(\sigma_2, l_2, t_1)) \geq 0. \end{aligned} \quad (6)$$

Again, a_2 is computed via the disciplinary analysis

$$a_2 = A_2(\sigma_2, l_2, t_1).$$

Following the terminology of OLD, we refer to the disciplinary objectives in CO as discrepancy functions, since they measure the discrepancy between system-level targets for disciplinary inputs and outputs and the closest values that can be obtained by disciplinary designers without violating disciplinary design constraints.

The introduction of disciplinary minimization subproblems of the form (5)–(6) is a distinctive characteristic of CO. The subproblems can be solved autonomously. By solving the subproblems, we eliminate the disciplinary design variables l_i from the system-level problem, and decouple the calculation of the disciplinary analysis outputs a_i . Information from the solutions of the disciplinary problems (5)–(6) is then used to define the system-level consistency constraints c_i . The type of system-level constraints used gives rise to a specific instance of CO.

The first instance of CO we discuss is the one in which CO has been most frequently presented (e.g., [13, 14, 16, 17]). In this formulation, the consistency condition is to drive to zero the value of the target mismatch objective in subproblems (5)–(6). At the system-level, the interdisciplinary consistency constraints are simply the optimal values of the objectives in (5)–(6). That is, the consistency constraints $C = (c_1, c_2)$ are defined as

$$c_1(s, t_1, t_2) = \frac{1}{2} [\| \bar{\sigma}_1(s, t_1, t_2) - s \|^2 + \| a_1(\bar{\sigma}_1(s, t_1, t_2), \bar{l}_1(s, t_1, t_2), t_2) - t_1 \|^2] \quad (7)$$

$$c_2(s, t_1, t_2) = \frac{1}{2} [\| \bar{\sigma}_2(s, t_1, t_2) - s \|^2 + \| a_2(\bar{\sigma}_2(s, t_1, t_2), \bar{l}_2(s, t_1, t_2), t_1) - t_2 \|^2], \quad (8)$$

where the bars over $\bar{\sigma}_1, \bar{\sigma}_2, \bar{l}_1, \bar{l}_2$ indicate that these values are the results of solving the disciplinary optimization subproblems for the given value of the system-level variables. We call this version CO₂, where the subscript “2” refers to the fact that the c_i are sums of squares.

An alternative to the system-level consistency conditions (7)–(8), giving rise to the second instance of CO, is to match the system-level variables directly with their subsystem counterparts computed in subproblems (5)–(6). The consistency constraints $C = (c_1, \dots, c_4)$ are

$$c_1(s, t_1, t_2) = \bar{\sigma}_1(s, t_1, t_2) - s$$

$$\begin{aligned} c_2(s, t_1, t_2) &= a_1(\bar{\sigma}_1(s, t_1, t_2), \bar{l}_1(s, t_1, t_2), t_2) - t_1 \\ c_3(s, t_1, t_2) &= \bar{\sigma}_2(s, t_1, t_2) - s \\ c_4(s, t_1, t_2) &= a_2(\bar{\sigma}_2(s, t_1, t_2), \bar{l}_2(s, t_1, t_2), t_1) - t_2. \end{aligned}$$

We denote this formulation CO₁ to indicate that the quantities in the system-level constraints are not sums of squares. Note that (c_1, c_2) are associated with Discipline 1, while (c_3, c_4) are associated with Discipline 2.

In either CO₁ or CO₂, we will call a value of the system-level variables (s, t_1, t_2) *realizable* for Discipline i if the optimal objective value in the corresponding disciplinary optimization problem (5) or (6) is zero. Realizable values of the system-level variables correspond to desirable designs. A design is desirable if the optimal objective value in the disciplinary optimization problem for Discipline i is zero. This means that Discipline i can exactly match the system-level input-output targets without violating the disciplinary design constraints. In general, there will be many realizable values of the system-level variables for a given discipline. A point (s, t_1, t_2) is feasible for the system-level problem when it is realizable for all the constituent disciplines.

Optimization by Linear Decomposition

OLD [5–12] maintains interdisciplinary consistency at the system level while seeking to minimize the violation of the disciplinary design constraints at the subsystem level. In this respect, OLD and CO complement one another.

In the lower-level problems, the disciplines use their local design degrees of freedom to minimize the violation of the disciplinary design constraints, subject to matching the target value for the disciplinary output that is fed into that discipline. This is effected as follows. In Discipline 1, we are given (s, t_1, t_2) and compute $\bar{l}_1(s, t_1, t_2)$ as a solution of the following minimization problem in l_1 :

$$\begin{aligned} \min_{l_1} \quad & c_1(s, l_1, t_1, t_2) \\ \text{s. t.} \quad & t_1 = a_1(s, l_1, t_2), \end{aligned} \quad (9)$$

The analysis output a_1 is computed in this disciplinary optimization problem via the disciplinary analysis

$$a_1 = A_1(s, l_1, t_2).$$

Note that in the disciplinary subproblem (9), the system-level variables (s, t_1, t_2) serve as parameters in the disciplinary optimization problem.

The disciplinary objective c_1 is any function with the following property:

For any (s, t_1, t_2) , we have

$$c_1(s, l_1, t_1, t_2) \leq 0$$

if and only if $g_1(s, l_1, a_1(s, l_1, t_2)) \geq 0$ for all l_1 satisfying $a_1(s, l_1, t_2) - t_1 = 0$.

Ideally, we also would like c_1 to be continuously differentiable. Following [11], we refer to c_1 as a *discrepancy function*.

There is an analogous problem for Discipline 2. Given (s, t_1, t_2) , we compute $\bar{l}_2(s, t_1, t_2)$ as a solution of the following minimization problem in l_2 :

$$\begin{aligned} \min_{l_2} \quad & c_2(s, l_2, t_1, t_2) \\ \text{s. t.} \quad & t_2 = a_2(\sigma_2, l_2, t_1). \end{aligned} \quad (10)$$

Again, a_2 is computed via the disciplinary analysis

$$a_2 = A_2(s, l_2, t_1).$$

The subproblems (9)–(10) can be solved autonomously. As in CO, we eliminate the disciplinary design variables l_i from the system-level problem via the solution of the disciplinary subproblems.

The optimal value of the objective in the disciplinary problems (9)–(10) defines the system-level consistency constraints c_i . The resulting system-level problem is

$$\begin{aligned} \min_{s, t_1, t_2} \quad & f(s, a_1(s, \bar{l}_1(s, t_1, t_2), t_2), a_2(s, \bar{l}_2(s, t_1, t_2), t_1)) \\ \text{s. t.} \quad & c_1(s, \bar{l}_1(s, t_1, t_2), t_1, t_2) \leq 0 \\ & c_2(s, \bar{l}_2(s, t_1, t_2), t_1, t_2) \leq 0. \end{aligned} \quad (11)$$

For OLD, we call a value of the system-level variables (s, t_1, t_2) *realizable* for Discipline i if the optimal objective value in the corresponding disciplinary optimization problem (9) or (10) is less than or equal to zero. As in CO, realizable values of the system-level variables correspond to desirable designs: if the optimal objective value in the disciplinary optimization problem for Discipline i is less than or equal to zero, then this means that Discipline i can exactly match the system-level input-output targets without violating the disciplinary design constraints. A point (s, t_1, t_2) is feasible for the system-level problem when it is realizable for all the constituent disciplines.

One choice of discrepancy function is

$$c_1(s, l_1, t_1, t_2) = \sum_j (\min(0, g_1^j(s, l_1, a_1(s, l_1, t_2))))^2 \quad (12)$$

$$c_2(s, l_2, t_1, t_2) = \sum_j (\min(0, g_2^j(s, l_2, a_2(s, l_2, t_1))))^2. \quad (13)$$

This objective is smooth (C^1). Also note that $c_i \geq 0$, so the system-level constraint $c_i \leq 0$ is tacitly an equality constraint $c_i = 0$. Another choice of discrepancy function is a relaxation of the problem via the Kreisselmeier-Steinhausner [25] (KS) approximation of $\max_j(g_i^j)$ (with g_i^j denoting the j -th component of the constraint vector g_i):

$$c_1(s, l_1, t_1, t_2) = \frac{1}{\rho} \ln \sum_j \exp(-g_1^j(s, l_1, a_1(s, l_1, t_2))) \quad (14)$$

$$c_2(s, l_2, t_1, t_2) = \frac{1}{\rho} \ln \sum_j \exp(-g_2^j(s, l_2, a_2(s, l_2, t_1))). \quad (15)$$

In this relaxation, we are approximating the disciplinary realizable sets with sets that are smoothly bounded and which are strictly inside the realizable sets. This is good, since it means the approximation errs on the side of feasibility.

Hybrids

There are also formulations that combine elements of OLD and CO [11, 12]. In these hybrids, the goal of the disciplinary subproblems is to minimize the discrepancy in both the system-level targets for the disciplinary inputs and outputs as well as the disciplinary design infeasibility. Recall the p -norm: if $v = (v^1, \dots, v^m)$, then

$$\|v\|_p = \sqrt[p]{\sum_{j=1}^m |v^j|^p}$$

The case $p = \infty$ corresponds to the max-norm

$$\|v\|_p = \max_j |v^j|.$$

The disciplinary subproblem for Discipline 1 is

$$\min_{l_1} c_1(s, l_1, t_1, t_2),$$

where the discrepancy function is

$$\begin{aligned} c_1(s, l_1, t_1, t_2) = & \left(\sum_j (\max(0, g_1^j(s, l_1, a_1(s, l_1, t_2))))^p \right. \\ & \left. + \|a_1(s, l_1(s, t_1, t_2), t_2) - t_1\|_p^p \right)^{1/p}. \end{aligned} \quad (16)$$

The discrepancy function and subproblem for Discipline 2 are similar. The cases $p = 1, 2$ and $p = \infty$ are of greatest interest.

One can also envision variants of CO along similar lines. For instance, one could consider alternative norms for the mismatch between the system-level targets and the disciplinary values of those variables. In place of (5), one might choose

$$\begin{aligned} \min_{\sigma_1, l_1} \quad & [\|\sigma_1 - s\|_p^p + \|a_1(\sigma_1, l_1, t_2) - t_1\|_p^p] \\ \text{s. t.} \quad & g_1(\sigma_1, l_1, a_1(\sigma_1, l_1, t_2)) \geq 0, \end{aligned}$$

and a similar problem for Discipline 2. As with OLD, the cases $p = 1, 2$ and $p = \infty$ are of greatest interest, with $p = 2$ being the discrepancy function discussed in our earlier description of CO. The analytical properties of the hybrid methods are similar to those of the CO and OLD.

Analytical properties

The two bilevel approaches considered here have analytical features that can have significant consequences for computation. As we discuss, the constraints in the system-level problem are of a nature that can make the system-level problem difficult to solve using conventional optimization algorithms. These features, in turn, derive from the multilevel nature of the formulations and the nature of the disciplinary subproblems used to eliminate the disciplinary design variables.

One encounters the following difficulties in the system-level problems associated with OLD and CO.

- If the system-level consistency constraints are smooth, then Lagrange multipliers do not exist for the system-level problem in the case of CO₂ and OLD with the discrepancy function (12)–(13).
- In some of the formulations, the system-level constraints have discontinuous derivatives at the solution of the system-level problem. This is the case, for instance, for CO₁.

The precise mathematical statements and proofs of these results appear elsewhere [2, 3]. We give here a sketch of what is going on, and explain how these difficulties arise from the nature of the system-level constraints and the bilevel nature of the approaches.

Analytical properties of CO. First, consider the system-level equality constraints in CO₂. These constraints vanish on the feasible region. Now, suppose the feasible region is an open set, as it is generally. Then, if these constraints are smooth (and they can be shown to be) then the constraint Jacobian must vanish on the feasible region. If $C(s, t_1, t_2)$ is the set of system-level constraints, then the usual first-order necessary condition characterizing a solution is that there exist multipliers λ for which

$$\nabla f(s, t_1, t_2) + \nabla C(s, t_1, t_2)\lambda = 0, \quad (17)$$

where ∇C is the transpose of the Jacobian of the system-level constraints. However, $\nabla C(s, t_1, t_2) = 0$ for all feasible (s, t_1, t_2) , so (17) can hold at a solution only if $\nabla f(s, t_1, t_2) = 0$. As a consequence, Lagrange multipliers for the CO₂ system-level problem do not exist unless the solution of the system-level problem is also an unconstrained stationary point of f , which is rarely the case. Thus, the system-level problem (4) that results in CO₂ fails to satisfy the standard first-order necessary conditions (Karush–Kuhn–Tucker or KKT conditions [26]) that characterize solutions.

The system-level constraints in CO₁ can be shown to have discontinuous first derivatives for values of the system-level targets that correspond to disciplinary designs on the boundary of one or more of the disciplinary

feasible regions. Unfortunately, it is generally the case that the solution of the system-level problem is such a point. The discontinuity occurs because one or more of the disciplinary design constraints become active at these points. (This is a manifestation of the well-known phenomenon of the nondifferentiable dependence of the solutions of nonlinear programs on parameters when the set of active constraints changes.) Because of the discontinuity of the constraint Jacobian, the usual Lagrange multiplier rule does not hold.

We emphasize that these effects necessarily take place at problem solutions. That is, either the system-level problem is smooth at a solution, as in the case of CO₂, in which case it does not have Lagrange multipliers, or the system-level problem is not smooth at a solution, as in the case of CO₁. The only exception to this situation occurs in the rare event that none of the design constraints are binding at the solution. It is precisely the feature of CO₂ that smoothes out the problem—the vanishing of the system-level constraint Jacobian—that assures that the Lagrange multipliers do not exist. In CO₁, the Lagrange multipliers do exist, but at the expense of the constraint derivative discontinuity.

These analytical features mean that conventional, smooth optimization algorithms will generally have increasing difficulty as they approach the solution to the problem. Worse yet, optimization algorithms may give bogus answers that are difficult to identify as such since the usual first-order conditions that characterize stationary points do not hold and computing system-level constraint derivatives is prone to large error near the solution.

Numerical experiments on simple (convex) problems confirm that these analytical features can make the system-level problem in collaborative optimization difficult to solve [2]. Numerical difficulties with CO were previously noted in [27] and later in [28, 29]; we believe these analytical features may explain these earlier observations, as well.

Analytical properties of OLD. If one chooses the discrepancy function (12)–(13) in OLD, one obtains a system-level problem for which Lagrange multipliers do not exist, as in CO₂. The reason is the same: the system-level constraints vanish on open sets, in general, and this means that the constraint Jacobians vanish.

One can use nonsmooth discrepancy functions, as in (16). This leads to nonsmooth constraints in the system-level problem, as in CO₁, which necessitates specialized optimization techniques, as discussed in [12].

Algorithmic consequences

The nonexistence of Lagrange multipliers for the system-level problem can cause trouble for methods that

rely on the KKT conditions, such as sequential quadratic programming or feasible directions methods (see, for instance, the discussion in [2]). Moreover, the nonexistence of multipliers is caused by the fact that the constraint Jacobians vanish on the interior of the realizable sets. The fact that the constraint Jacobians vanish can cause additional numerical problems in the solution of the system-level problem. Unfortunately, these difficulties arise at realizable points, which are the designs that make the most sense for the underlying engineering problem. In order to be assured that conventional optimization methods will work at all reliably to approach a solution of the system-level problem, one needs to introduce mechanisms to stay safely away from the realizable sets of designs. This seems counterintuitive, but is a consequence of the consistency constraints that result from the attempts to remove the disciplinary design variables from the system-level problem, and the system-level constraints that ensue.

In the case of nonsmooth system-level constraints, the discontinuities in the constraint Jacobians commonly occur at solutions. This also causes problems for smooth optimization techniques applied to the system-level problems. In [12] the authors sketch an algorithm for handling the nonsmoothness that arises in the hybrid OLD approach. In effect, their algorithm approximates the generalized derivative of the nonsmooth (but Lipschitz) equality constraints. The authors report satisfactory results with this approach, though they note that there is a potential for failure.

We note that OLD has been proposed in connection with an algorithm for its solution (e.g., [7, 8]). The algorithm attempts to avoid the expense of performing subsystem optimization problems every time a constraint or a constraint derivative is required for the solution of the system-level optimization problem. Instead, [7, 8] propose an approach that could be viewed as analogous to sequential quadratic or linear programming (SQP or SLP, respectively). In this approach, the system-level problem is solved with linearized constraints; i.e., for each design cycle, the constraint value and constraint derivative are held constant. The distinction from SQP or SLP is that the objective is used instead of its quadratic or linear model. However, this algorithm will still suffer from the analytical features of OLD discussed previously.

Another peculiarity of the system-level problems arises in connection with the system-level equality constraints that vanish identically on the interior of the feasible region. If one starts an equality constrained optimization algorithm at a feasible point, even very near an optimal solution, one will frequently immediately leave the feasible region, and the remaining iterates will work their way back towards the solution through a sequence of infeasible designs. This behavior was observed in [2, 12], for instance. This occurs because the system-level equal-

ity constraints provide no hint that one is near the boundary of the feasible region because the constraint Jacobian is zero there. In the absence of any information about the boundary of the feasible region, the problem appears unconstrained and one ends up leaving the feasible region. The system-level problem is an odd equality constrained optimization problem. The feasible region defined by the equality constraints is generally an open set. Usually if one has only equality constraints, the feasible region is a hypersurface, and all the constraints are binding at a solution. This is not the case with the system-level constraints that vanish on the interior of realizable sets.

A formulation reflecting the algorithmic perspective

Starting with the fully integrated approach (3), we can formulate the MDO problem in a way that respects the requirements of conventional nonlinear programming analysis and algorithms and avoids the analytical difficulties of the bilevel approaches we have discussed. Hence this formulation represents the algorithmic perspective.

One such class of approaches appears in [23, 24, 30], where it is called either the “in-between” or “individual discipline feasible” (or IDF) approach. The latter name is unfortunate since it suggests that the formulation insures designs that satisfy the disciplinary design constraints, whereas it really refers to the fact that the analysis outputs are consistent with (“feasible with respect to”) the disciplinary analyses, though not the multidisciplinary analysis.

To avoid this confusion, we use the term Distributed Analysis Optimization to refer to a general class of methods that includes the IDF approach from [23, 24, 30]. In this formulation, we treat the implicit interdisciplinary consistency constraints in the multidisciplinary analysis as explicit equality constraints in the optimization problem. The fully integrated approach (3) becomes

$$\begin{aligned}
 \min_{s, l_1, l_2, t_1, t_2} \quad & f(s, t_1, t_2) \\
 \mathbf{s. t.} \quad & g_1(s, l_1, t_1) \geq 0 \\
 & g_2(s, l_2, t_2) \geq 0 \\
 & t_1 = a_1(s, l_1, t_2) \\
 & t_2 = a_2(s, l_1, t_1),
 \end{aligned} \tag{18}$$

where

$$\begin{aligned}
 a_1(s, l_1, t_2) &= A_1(s, l_1, t_2) \\
 a_2(s, l_2, t_1) &= A_2(s, l_2, t_1).
 \end{aligned}$$

Details of the formulation may be found in the companion paper [4].

Analytical properties of DAO. The DAO formulations enjoy the same smoothness and stability properties as the fully integrated formulation (3). There is no difficulty with Lagrange multipliers or nonsmoothness.

Algorithmic consequences of DAO. The DAO formulations have a larger number of optimization variables than does the fully integrated formulation (3). The increase in the number of optimization variables depends on the bandwidth of the interdisciplinary coupling. Because the analytical properties of DAO are the same as those of the fully integrated approach, performance of optimization algorithms applied to DAO will not suffer as a consequence of reformulation.

DAO formulations possess another attractive algorithmic feature. Optimization algorithms are frequently sensitive to the quality of derivative information. Because the multidisciplinary analyses are not usually computed to a high degree of accuracy, computing finite difference derivatives of an integrated multidisciplinary analysis is an error prone procedure. Therefore, methods that rely on MDA for function evaluations may not be able to obtain sufficiently accurate derivatives. Because DAO formulations do not require multidisciplinary analyses to be performed, the sensitivity of optimization algorithms with respect to the convergence of MDA is not an issue for DAO. Thus, the numerical performance of optimization algorithms may improve compared to solving MDA-based formulations.

Relationship among the methods

All the approaches we discuss here can be viewed from the perspective of eliminating various subsets of variables from the DAO formulation. Begin with the DAO approach:

$$\begin{aligned}
 \min_{s, l_1, l_2, t_1, t_2} \quad & f(s, t_1, t_2) \\
 \text{s. t.} \quad & g_1(s, l_1, t_1) \geq 0 \\
 & g_2(s, l_2, t_2) \geq 0 \\
 & t_1 = a_1(s, l_1, t_2) \\
 & t_2 = a_2(s, l_2, t_1).
 \end{aligned} \tag{19}$$

If we eliminate t_1, t_2 as independent variables from (19) by always requiring

$$\begin{aligned}
 t_1 &= a_1(s, l_1, t_2) \\
 t_2 &= a_2(s, l_2, t_1),
 \end{aligned}$$

then we obtain the fully integrated approach (3), since we are requiring our designs to satisfy the multidisciplinary analysis consistency equations (1)–(2).

OLD can be viewed as taking the further step of eliminating the disciplinary design variables l_1, l_2 as independent variables from the optimization problem, in addition to eliminating t_1, t_2 . This elimination is accomplished via the subsystem problems (9)–(10). Thus, in OLD, multidisciplinary analysis is performed at each iteration. The hybrid methods can also be viewed as trying to eliminate

both the coupling variables t_1, t_2 and the disciplinary design variables l_1, l_2 , as in OLD. However, the elimination of t_1, t_2 is accomplished via the objective in the subsystem optimization problems, rather than through equality constraints in the subsystem optimization problems, a relaxation of the approach in OLD.

CO, on the other hand, eliminates the disciplinary design variables l_1, l_2 in DAO via (5)–(6), but does not eliminate the coupling variables t_1, t_2 . Like DAO, the multidisciplinary analysis is enforced via the system-level constraints.

Consequences of disciplinary autonomy

As we mentioned, disciplinary autonomy is generally desirable in an approach to formulating and solving MDO problems. It is useful to distinguish between two types of disciplinary autonomy in the context of MDO. First, autonomy of integration or implementation is the extent to which the problem can be implemented in a decentralized manner. Second, autonomy of execution is the extent to which the calculations that arise in the optimization are decoupled.

Autonomy of integration—the ability to implement the requisite computational modules independently along the lines of the disciplines—is an attractive feature in a formulation. However, if one begins with a multidisciplinary analysis, the implementation of the fully integrated formulation (3) may require extensive interaction among disciplinary experts, since the underlying analysis may have been put together in a way that might not be quite appropriate for the purposes of optimization, and may require a degree of re-assembly. For instance, one needs to develop a sensitivity capability that involves the MDA.

The bilevel methods we have discussed exhibit autonomy of integration. Moreover, single-level optimization formulations can also exhibit autonomy of integration if properly implemented. This is the case of the DAO approach, as discussed in the companion paper [4].

One goal of autonomy of execution may be to simplify the computational process by distributing the disciplinary computations. However, the main aim of this autonomy is usually expressed as the goal of conforming to disciplinary organizational procedures. In other words, it would be useful if the disciplinary organizations could perform their disciplinary design activities independently, with the help of some targets passed from the system coordinator.

Autonomy of execution is a question of the interdisciplinary coupling in the problem. There is the degree to which the calculations can be performed independently across the disciplines. There is also the question of whether the disciplinary design variables can be elimi-

nated from the system-level problem.

The bilevel approaches we have discussed possess a certain degree of execution autonomy, because the system-level optimizer does provide the disciplinary problems with targets for disciplinary outputs and, possibly, for the shared variables, while allowing each disciplinary optimization problem to manipulate its set of local design variables. However, the disciplinary subproblems are not disciplinary design optimization in the single-discipline sense. That is, the subproblem objective functions are not disciplinary objectives, such as lift, drag, or weight. Instead, the subsystem objectives serve to minimize the inconsistency among the disciplinary analyses (as in CO), or, disciplinary design infeasibility (as in OLD).

Elimination of the disciplinary design variables from the system-level problem is the most attractive feature of the bilevel formulations. However, as we have discussed, the elimination of the disciplinary design variables leads to system-level problems that have analytical features that can cause trouble for conventional numerical algorithms. In contrast, the single-level formulations (3) and (18) do not eliminate the local variables from the system-level optimization problem. This feature is the price single-level optimization formulations pay for robust solution by conventional nonlinear programming algorithms.

Formulations that decompose the MDA, as do CO and DAO, have two significant drawbacks. First, the intermediate iterates or designs will not necessarily satisfy the multidisciplinary analysis (1)–(2) until a solution is reached. In contrast, the solution process of the fully integrated formulation (3) and OLD will always have iterates that satisfy the multidisciplinary analysis, and thus are physically realizable, even if the optimization has to be stopped without having attained an optimum.

Second, distributed formulations are difficult to treat in optimization under uncertainty—the area that is becoming increasingly important in engineering design. This is because one introduces independent auxiliary variables that may need to match random variables whose statistical properties are not known *a priori*. The fully integrated formulation can incorporate uncertainty in a natural manner. This feature is discussed in more detail elsewhere.

Despite the drawbacks of the distributed formulations outlined above, autonomy is an attractive feature. When considering a problem formulation that aims for disciplinary autonomy, one should consider the following technical challenges:

- Variables local to disciplinary subsystems frequently enter into the disciplinary problem description as a part of an inequality constraint system. Eliminating inequalities from the system-level problem in a robust manner is difficult.

- MDO systems are inherently coupled. Decoupling is performed by introducing local copies of the coupling variables at the subsystem level. However, the coupling must be restored, at least at the solution of the overall MDO problem. Some of the constraints that restore coupling have a form that will cause difficulties for conventional optimization algorithms.

As we have discussed, the techniques by which the bilevel methods described here decompose MDA can cause practical difficulties for computational optimization. For this reason it is preferable to decompose MDA by means of a DAO formulation, if possible. This is especially so because true disciplinary design autonomy is not achieved by the bilevel methods in question, i.e., the subproblem objectives are not disciplinary objectives, but serve to attain MDA at solutions. On the other hand, DAO formulations do not eliminate the local disciplinary variables from the system-level optimization problem, and, in fact, increase the number of design variables. Whether this is a difficulty depends on the specific problem and the optimization method used. Recent advances in large-scale optimization algorithms lead us to believe that the presence of the local variables in the system-level problem may not present a lasting difficulty.

Concluding remarks

Bilevel optimization formulations arise naturally in an attempt to treat problem complexity, size, and expense by allowing for disciplinary autonomy of implementation and execution. Any method that yields a system-level problem with equality constraints that represent the disciplinary feasible regions will suffer from the difficulties we have discussed. If the constraints are smooth, then Lagrange multipliers will not exist. On the other hand, the system-level constraints may be nonsmooth. Moreover, the system-level constraints, if they are explicitly equalities (as in CO) or implicitly equalities (as in OLD with the discrepancy function (12)–(13)), do not reveal the presence of the boundary of the feasible region. These features will confuse constrained optimization algorithms, as we have discussed.

Thus, elimination of the disciplinary design variables comes at the price of a system-level problem with undesirable analytical features that may cause trouble for computation. The distributed formulation we currently consider as a viable alternative decomposes MDA but does not eliminate local variables from the system-level problem. On a positive side, DAO does not suffer from the analytical and computational drawbacks of the bilevel optimization formulations. We conjecture that the most viable approaches to solving MDO problems will combine conventional problem formulations with specialized opti-

mization algorithms. Several alternatives are being currently considered.

References

- [1] Natalia M. Alexandrov and Robert Michael Lewis. Comparative properties of collaborative optimization and other approaches to MDO. In *Proceedings of the ASMO UK/ISSMO Conference on Engineering Design Optimization*, 1999. Available as ICASE technical report 99-24.
- [2] Natalia M. Alexandrov and Robert Michael Lewis. Analytical and computational aspects of collaborative optimization. National Aeronautics and Space Administration, NASA/TM-210104-2000, April 2000.
- [3] Natalia M. Alexandrov and Robert Michael Lewis. An analysis of some bilevel approaches to multidisciplinary design optimization. Technical report, Institute for Computer Applications in Science and Engineering, Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia 23681-2199, June 2000.
- [4] Natalia M. Alexandrov and Robert Michael Lewis. Algorithmic perspectives on problem formulations in MDO. AIAA Paper 2000-4719, 2000. Presented at the Eighth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.
- [5] Jaroslaw Sobieszczanski-Sobieski. A linear decomposition method for large optimization problems—blueprint for development. National Aeronautics and Space Administration, NASA/TM-83248-1982, 1982.
- [6] Jean-Francois Marie Barthelemy. Development of a multilevel optimization approach to the design of modern engineering systems. NASA/CR-172184-1983, August 1983.
- [7] Jaroslaw Sobieszczanski-Sobieski, Benjamin James, and Augustine Dovi. Structural optimization by multilevel decomposition. *AIAA Journal*, 23:1775–1782, 1985.
- [8] Jaroslaw Sobieszczanski-Sobieski, Benjamin B. James, and Michael F. Riley. Structural sizing by generalized, multilevel optimization. *AIAA Journal*, 25(1):139–145, January 1987.
- [9] G. A. Wrenn and A. R. Dovi. Multilevel decomposition approach to the preliminary sizing of a transport aircraft wing. *AIAA Journal*, 25:632–638, 1988.
- [10] Jaroslaw Sobieszczanski-Sobieski. Two alternative ways for solving the coordination problem in multi-level optimization. *Structural Optimization*, 6:205–215, 1993.
- [11] R. J. Balling and S. Sobieszczanski-Sobieski. Optimization of coupled systems: A critical overview of approaches. AIAA Paper 94-4330, September 1994. Presented at the Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Florida, September 7–9, 1994.
- [12] R. J. Balling and S. Sobieszczanski-Sobieski. An algorithm for solving the system-level problem in multilevel optimization. AIAA Paper 94-4333, September 1994. Presented at the Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Florida, September 7–9, 1994.
- [13] Robert D. Braun. *Collaborative Optimization: An architecture for large-scale distributed design*. PhD thesis, Stanford University, May 1996. Department of Aeronautics and Astronautics.
- [14] R. D. Braun, A. A. Moore, and I. M. Kroo. Collaborative approach to launch vehicle design. *Journal of Spacecraft and Rockets*, 34(4):478–486, July 1997.
- [15] R. D. Braun and I. M. Kroo. Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In N. M. Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, pages 98–116. SIAM, 1997.
- [16] Ian Sobieski and Ilan Kroo. Aircraft design using collaborative optimization. AIAA Paper 96-0715, 1996. Presented at the 34th AIAA Aerospace Sciences Meeting, Reno, Nevada, Jan. 15-18, 1996.
- [17] L. A. Schmit, Jr. and R. K. Ramanathan. Multilevel approach to minimum weight design including buckling constraints. *AIAA Journal*, 16(2):97–104, February 1978.
- [18] Lucien A. Schmit, Jr. and Massood Mehrinfar. Multilevel optimum design of structures with fiber-composite stiffened-panel components. *AIAA Journal*, 20(1):138–147, January 1982.
- [19] L. A. Schmit and K. J. Chang. A multilevel method for structural synthesis. In *A collection of technical papers : AIAA/ASME/ASCE/AHS 25th Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics and Astronautics, 1984.

- [20] Howard M. Adelman, Joanne L. Walsh, and Jocelyn I. Pritchard. Recent advances in integrated multidisciplinary optimization of rotorcraft. In *Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, pages 710–721, September 1992. AIAA paper 92-4777.
- [21] Joanne L. Walsh, Katherine C. Young, Jocelyn I. Pritchard, Howard M. Adelman, and Wayne R. Mantay. Integrated aerodynamic/dynamic/structural optimization of helicopter rotor blades using multilevel decomposition. NASA Technical Paper 3465, January 1995.
- [22] A. Migdalas, P.M. Pardalos, and P. Värbrand, editors. *Multilevel Optimization: Algorithms and Applications*. Kluwer, 1998.
- [23] Evin J. Cramer, J. E. Dennis, Jr., Paul D. Frank, Robert M. Lewis, and Gregory R. Shubin. On alternative problem formulations for multidisciplinary design optimization. In *Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, pages 518–530, 1992. AIAA Paper 92-4752.
- [24] Robert Michael Lewis. Practical aspects of variable reduction formulations and reduced basis algorithms in multidisciplinary design optimization. In Natalia Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State-of-the-Art*, Philadelphia, 1997. SIAM.
- [25] G. Kreisselmeier and R. Steinhauser. Systematic control design by optimizing a vector performance index. In M. A. Cuenod, editor, *Computer Aided Design of Control Systems*, pages 113–117. Pergamon Press, 1980. Proceedings of the IFAC Symposium on Computer Aided Design of Control Systems, 29-31 August 1979, Zurich, Switzerland.
- [26] Anthony V. Fiacco and Garth P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Classics in Applied Mathematics. SIAM, Philadelphia, 1990.
- [27] Rajiv Thareja and Raphael T. Haftka. Numerical difficulties associated with using equality constraints to achieve multi-level decomposition in structural optimization. AIAA Paper 86-0854, 1986.
- [28] N. M. Alexandrov and S. Kodiyalam. Initial results of an MDO method evaluation study. In *Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998. AIAA Paper 98-4884.
- [29] S. Kodiyalam. Evaluation of methods for multidisciplinary design optimization, Phase I. Technical Report NASA/CR-1998-208716, National Aeronautics and Space Administration, September 1998.
- [30] J. E. Dennis, Jr. and Robert Michael Lewis. A comparison of nonlinear programming approaches to an elliptic inverse problem and a new domain decomposition approach. Technical Report TR94-33, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005-1892, August 1994.