AIAA 2000-4719

Algorithmic Perspectives on Problem Formulations in 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

ALGORITHMIC PERSPECTIVES ON PROBLEM FORMULATIONS IN MDO

NATALIA M. ALEXANDROV* AND ROBERT MICHAEL LEWIS†

Abstract

This work is concerned with an approach to formulating the multidisciplinary optimization (MDO) problem that reflects an algorithmic perspective on MDO problem solution. The algorithmic perspective focuses on formulating the problem in light of the abilities and inabilities of optimization algorithms, so that the resulting nonlinear programming problem can be solved reliably and efficiently by conventional optimization techniques. We propose a modular approach to formulating MDO problems that takes advantage of the problem structure, maximizes the autonomy of implementation, and allows for multiple easily interchangeable problem statements to be used depending on the available resources and the characteristics of the application problem.

Key Words: Autonomy, complex system design, distributed analysis optimization, multidisciplinary analysis, multidisciplinary optimization, nonlinear programming, sensitivities, system synthesis

Introduction

The underlying theme of this and related papers [1–3] is the strong influence of the analytical features of multidisciplinary optimization (MDO) problem formulation on the ability of nonlinear programming (NLP) algorithms to solve the problem reliably and efficiently. The companion paper [3] examines several distributed formulations for MDO and concludes that the desirable trait of autonomy for the subsystems that comprise the MDO problem may come at a price: In some of the approaches considered, the system-level constraints introduced to relax the interdisciplinary coupling and enable disciplinary autonomy can cause analytical and computational difficulties for optimization algorithms.

The premise of the present paper is that algorithmic considerations should enter strongly into decisions about problem formulation. We focus both on the MDO problem structure—in particular, the underlying multidisciplinary analysis—and the need to solve the problem reliably and efficiently. We propose a modular approach to MDO problem formulation that reflects both concerns. We examine several MDO problem formulations and show how the formulations are related to one another. We call the approach modular because, given the appropriate disciplinary sensitivity information, it is straightforward to pass from one formulation to another in the context of optimization. Similar techniques were used in [4] to derive the flexible algorithms for different classes of formulations in the framework of reduced basis algorithms.

The conclusion (and contribution) of this paper may be summarized as follows: an appropriate implementation of the disciplinary information allows one to combine the disciplinary "pieces" into several MDO problem formulations and solution algorithms with minimal extra effort. The re-use of information is facilitated by the structure of the problem due to disciplinary analyses and interdisciplinary coupling. That is, the disciplinary analyses can be viewed as nonlinear equality constraints. Using this fact, function and derivative information, if implemented appropriately, can be easily re-used in the context of reduced-basis optimization algorithms. Moreover, the function and derivative information required for optimization is similar for all derivative-based algorithms. We argue that the structure is not only a convenient device, but should be exploited in order to formulate and solve the optimization problem efficiently in the framework of robust nonlinear programming algorithms.

Multidisciplinary analysis

For clarity, we present our discussion of formulations for a two-discipline MDO problem. Discipline 1 and Discipline 2 may represent, for instance, the aeroelastic interaction between aerodynamics and structural response for a wing in steady-state flow. In this section, we briefly present the multidisciplinary analysis and the motivation

*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.

for the present work.

We assume that each constituent subsystem of the two-discipline model problem is based on a disciplinary analysis. Each discipline takes as its input a set of disciplinary design variables l_i, a set of shared or system-level design variables s, and some parameters. The variables l_1 and l_2 are local to Disciplines 1 and 2, respectively. Let the vector a_i represent the totality of outputs from a given discipline, including all data passed to the other discipline as parameters and quantities passed to design constraints and objectives, if any. In the context of MDO, the parameters fed into Discipline 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. We denote the totality of design variables by x, i.e., $x = (s, l_1, l_2)$.

The coupled multidisciplinary analysis system (MDA) reflects the physical requirement that a solution simultaneously satisfy the two disciplinary analyses. The input parameters to each discipline are required to correspond to some (or all) of the outputs a_j from the other disciplinary analysis. We write the multidisciplinary analysis as a simultaneous system of equations. Given $x = (s, l_1, l_2)$, we have a block system

$$
\begin{aligned}
a_1 &= A_1(s, l_1, a_2) & (1)\\
a_2 &= A_2(s, l_2, a_1). & (2)
\end{aligned}
$$

In the context of MDO, the blocks of the system represent the disciplinary analyses together with the necessary interdisciplinary couplings.

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 outputs are also known as disciplinary responses. The MDA implicitly defines a_1 and a_2 as functions of (s, l_1, l_2):

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

In analysis mode or at each iteration of a conventional optimization procedure, the design variable vector x is passed to the MDA system. The system is then solved for the state vector $a = (a_1, a_2)$. The solution is accomplished via an iterative procedure such as Gauss-Seidel or an inexact Newton method.

The implementation of MDA, arguably, takes most of the integration effort in solving MDO problems. It may require a degree of assembly of the MDA and extensive interaction among disciplinary experts. For instance, for design one needs to develop a sensitivity capability that involves the MDA. Because MDA is not usually converged to a high degree of accuracy, the computation of finite-difference derivatives that involve MDA is difficult. In addition, from the nonlinear programming perspective, far from solutions, it is preferable not to follow the complicated and expensive constraint manifold defined by solutions of (1)–(2). Instead, the fastest (Newton-based) optimization algorithms rely on the so-called infeasible principle that calls for attaining feasibility with respect to all constraints only at solutions. For these reasons, dispensing with MDA is one of the main motivations of distributed optimization approaches [5].

However, completely avoiding MDA may be premature. First, solving the coupled equations (1)–(2) leads to designs in which the coupled disciplines give a physically consistent (and thus meaningful) result. Therefore, the MDA is generally performed at least to obtain a baseline design. Second, when optimization algorithms are applied to formulations that decompose the MDA, the intermediate iterates or designs will not necessarily satisfy the MDA (1)–(2) until a solution is reached. Thus, if an optimization process has to be stopped due to exhausted resources, the final design may not be physically realizable. And, finally, if the bandwidth of coupling among the disciplines is great, decomposing MDA may not be a viable option, due to the need for introducing a large number of auxiliary variables that assist in decoupling the MDA.

The approach proposed here attempts to reconcile the apparent contradiction between the need to establish the autonomy of integration—the ability to implement the requisite computational modules independently along the disciplinary lines—and the current need to keep MDA and MDA-based optimization methods in the arsenal of MDO. This is accomplished by considering the modularity (i.e., re-usability) of disciplinary information in the context of MDO formulations and optimization algorithms.

In summary, we do not reject the MDA and MDA-based optimization problem formulations. Instead, we propose a computational "recipe" for implementing MDO problems in a way that facilitates experimentation with MDO problem formulations.

Formulations of a two-discipline model problem

This section describes several optimization problem formulations, with a focus on rigorous notation and the relationship among the described formulations. Having stated the formulations in detail, we later examine them to see how we can re-use the disciplinary function and sensitivity information in one formulation in the implementation of other formulations. Obviously, the perspective advocated in this work is not limited to the formulations under discussion.

Fully integrated optimization formulation

The conventional approach to MDO problem formulation is to impose an optimizer over the MDA. Given the need to satisfy the MDA at a solution, this formulation is, arguably, natural. We call it the fully integrated optimization (FIO) formulation and use it to represent the original problem, i.e., the problem one ideally wishes to solve. Its mathematical statement is

$$\begin{array}{ll} \min\limits_{x} & f_{FIO}(x) \equiv f(x, a_1(x), a_2(x)) \\ \textbf{s. t.} & g_0(s, l_1, a_1(x), a_2(x)) \geq 0 \\ & g_1(s, l_1, a_1(x)) \geq 0 \\ & g_2(s, l_2, a_2(x)) \geq 0, \end{array} \tag{3}$$

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

The FIO formulation is also known as Multidisciplinary Feasible [6] in the nonlinear programming community and sometimes as "All-in-One" in the engineering community. The first term refers to maintaining feasibility with respect to the multidisciplinary analysis, but is somewhat misleading because it also appears to imply feasibility with respect to disciplinary constraints. The second term resembles the term "All-at-Once" which sometimes refers to a method at the opposite end of the formulation spectrum (see the next subsection). The term "FIO" is proposed to avoid these ambiguities in the naming convention.

The fully integrated formulation reflects a variable reduction technique. At each iteration of an optimization procedure applied to (3), the design variable vector x is passed to the MDA system. Solving the system for the state vector a reduces the dimension of the optimization problem (3) by making it a problem in x only; $a = a(x)$ is a dependent variable.

The drawback of this formulation is the need to solve the MDA at each iteration. The drawback is also its benefit, in that all optimization iterates are feasible with respect to the MDA, even far from solution. Thus, even if optimization cannot continue to a solution, the intermediate designs will be physically realizable. Another benefit of (3) is that is has the smallest possible number of optimization variables.

SAND formulation

A formulation class at the opposite end of the spectrum views the full MDA purely as equality constraints:

$$\begin{array}{ll} \min\limits_{x, a_1, a_2} & f_{SAND}(x, a_1, a_2) \equiv f(x, a_1, a_2) \\ \textbf{s. t.} & a_1 - A_1(s, l_1, a_2) = 0 \\ & a_2 - A_2(s, l_2, a_1) = 0 \\ & g_0(s, l_1, a_1, a_2) \geq 0 \\ & g_1(s, l_1, a_1) \geq 0 \\ & g_2(s, l_2, a_2) \geq 0. \end{array} \tag{4}$$

In structural optimization, this approach is known as Simultaneous Analysis and Design (SAD or SAND) [7], a convention we adopt. It has also been called the All-at-Once approach (e.g., [6]).

SAND is motivated by the nonlinear programming experience which suggests that allowing optimization algorithms the freedom not to follow the MDA constraint manifold (i.e., not performing MDA) far from solutions facilitates finding solutions at a lower expense. Optimality is attained together with feasibility only at solutions.

Distributed analysis optimization

The class of distributed analysis optimization (DAO) formulations relies on the introduction of auxiliary variables and consistency constraints that decouple the MDA, thus imparting some degree of autonomy to the disciplinary computations.

We first rewrite the MDA (1)–(2) as the equivalent system

$$\begin{aligned} a_1 - A_1(s, l_1, t_2) &= 0 \\ a_2 - A_2(s, l_2, t_1) &= 0 \\ t_1 - a_1 &= 0 \qquad (5) \\ t_2 - a_2 &= 0. \qquad (6) \end{aligned}$$

The auxiliary variables t_1 and t_2 stand in for the disciplinary responses a_1 and a_2, respectively. They serve to decouple the MDA equations (1)–(2).

We can now re-write the fully integrated formulation (3) as an equivalent formulation

$$\begin{array}{ll} \min\limits_{s, l_1, l_2, t_1, t_2} & f_{DAO}(s, t_1, t_2) = f(s, t_1, t_2) \\ \textbf{s. t.} & g_0(s, t_1, t_2) \geq 0 \\ & g_1(s, l_1, t_1) \geq 0 \\ & g_2(s, l_2, t_2) \geq 0 \\ & t_1 = a_1(s, l_1, t_2) \\ & t_2 = a_2(s, l_2, t_1), \end{array} \tag{7}$$

where, given (s, l_1, l_2, t_1, t_2), the disciplinary responses $a_1(s, l_1, t_2)$ and $a_2(s, l_2, t_1)$ are found by solving the disciplinary analysis equations

$$\begin{aligned} a_1 - A_1(s, l_1, t_2) &= 0 \\ a_2 - A_2(s, l_2, t_1) &= 0. \end{aligned}$$

Equations (5) and (6) are examples of interdisciplinary consistency constraints. The degrees of freedom introduced by expanding the set of optimization variables to include t_1, t_2 are removed by the consistency constraints.

The DAO approach introduces a degree of disciplinary autonomy but respects the requirements of conventional nonlinear programming analysis and algorithms and avoids the analytical difficulties of the bilevel approaches discussed shortly.

Other members of the DAO class appeared earlier in [4,6,8] under the names of "in-between" or "individual discipline feasible" (IDF) approaches. The latter name is unfortunate since it suggests that the formulation maintains 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 DAO to refer to a general class of methods that includes the IDF approach from [4,6,8]. In this formulation, we treat the implicit interdisciplinary consistency constraints in the multidisciplinary analysis as explicit equality constraints in the optimization problem.

In the DAO approach, further closure with respect to disciplinary design constraints or system level constraints is determined by the kind of optimization algorithm used. Ideally, one would be able to start with design variables (s, l_1, l_2) for which the disciplinary design constraints defined by the g_i are satisfied. One could then apply an optimization algorithm that maintained feasibility with respect to these constraints so that all subsequent designs obtained in the course of the optimization satisfied the disciplinary design constraints, thereby accomplishing the same end that other problem formulations (e.g., Collaborative Optimization) achieve through the definition of its disciplinary optimization problems.

On the other hand, it might be difficult to find initial design variables (s, l_1, l_2) for which the disciplinary design constraints are satisfied. To address this difficulty, we can expand the space of variables as follows:

$$\begin{array}{ll} \min\limits_{s,\sigma_1,\sigma_2,l_1,l_2} & f_{DAO}(s, t_1, t_2) \\ \textbf{s. t.} & g_0(\sigma_0, t_1, t_2) \geq 0 \\ & g_1(\sigma_1, l_1, t_1) \geq 0 \\ & g_2(\sigma_2, l_2, t_2) \geq 0 \\ & t_1 = a_1(\sigma_1, l_1, t_2) \\ & t_2 = a_2(\sigma_2, l_2, t_1) \\ & \sigma_0 = s \\ & \sigma_1 = s \\ & \sigma_2 = s. \end{array} \tag{8}$$

This relaxes the requirement that the disciplinary design constraints be satisfied with the system-level values of s. In particular, we now have the flexibility to select the initial σ_i in a way that the disciplinary design constraints are satisfied, exactly as in Collaborative Optimization. One can then apply an optimization algorithm that enforces feasibility with respect to the disciplinary design constraints.

It is straightforward to verify that DAO is equivalent to the original FIO. This makes DAO easy to analyze; for instance, if standard constraint qualifications are satisfied by the original problem, then they also hold for the DAO formulation. The convergence properties of optimization algorithms applied to DAO are those of the algorithms applied to conventional FIO. Given a good solver for equality constrained optimization problems, the method is expected to be efficient. We note that nonlinear equality constraints are usually considered undesirable in engineering optimization, possibly due to the preference for feasible iteration methods in engineering optimization software (e.g., [?]). However, most state-of-the-art methods in nonlinear programming do not maintain feasibility with respect to nonlinear equality constraints at every iteration. Instead, feasibility is attained only at solutions. Thus, a good equality constrained solver can generally handle additional nonlinear equality constraints with relative ease.

Collaborative Optimization

Collaborative Optimization (CO) [9–20] is a bilevel approach with a system-level problem of the following form:

$$\begin{array}{ll} \min\limits_{s,t_1,t_2} & f(s, t_1, t_2) \\ \textbf{s. t.} & C(s, t_1, t_2) = 0. \end{array} \tag{9}$$

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

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.

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):

$$\begin{array}{ll} \min\limits_{\sigma_1,l_1} & \frac{1}{2}\left[\| \sigma_1 - s \|^2 + \| a_1(\sigma_1, l_1, t_2) - t_1 \|^2\right] \\ \textbf{s. t.} & g_1(\sigma_1, l_1, a_1(\sigma_1, l_1, t_2)) \geq 0, \end{array} \tag{10}$$

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

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

Overbars (e.g., $\bar{\sigma}_1, \bar{l}_1$) indicate optimal solutions of subsystem problems as a function of system-level variables. In the disciplinary subproblem (10), the system-level variables (s, t_1, t_2) serve either as parameters or targets that we seek 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

$$\begin{array}{ll} \min\limits_{\sigma_2, l_2} & \frac{1}{2}\left[\| \sigma_2 - s \|^2 + \| a_2(\sigma_2, l_2, t_1) - t_2 \|^2\right] \\ \textbf{s. t.} & g_2(\sigma_2, l_2, a_2(\sigma_2, l_2, t_1)) \geq 0. \end{array} \tag{11}$$

Again, a_2 is computed via the disciplinary analysis

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

Reformulating the FIO formulation along the lines of CO also requires the introduction of the third set of system-level constraints and a "Discipline 0" subproblem that represents the system-level design constraint. Optimization of "Discipline 0" treats the system-level design constraints to obtain $\bar{\sigma}_0(s, t_1, t_2)$, $\bar{\zeta}_1(s, t_1, t_2)$, and $\bar{\zeta}_2(s, t_1, t_2)$:

$$\begin{array}{ll} \min\limits_{\sigma_0, \zeta_1, \zeta_2} & \frac{1}{2}\left[\| \sigma_0 - s \|^2 + \| \zeta_1 - t_1 \|^2 + \| \zeta_2 - t_2 \|^2\right] \\ \textbf{s. t.} & g_0(\sigma_0, \zeta_1, \zeta_2) \geq 0. \end{array} \tag{12}$$

Disciplinary minimization subproblems of the form (10)–(12) distinguish 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 (10)–(11) is used to define the system-level consistency constraints c_i.

We discuss the version of CO in which the consistency condition is to drive to zero the value of the target mismatch objective in subproblems (10)–(11). At the system-level, the interdisciplinary consistency constraints are simply the optimal values of the objectives in (10)–(11). That is, the consistency constraints $C = (c_1, c_2)$ are defined as

$$\begin{aligned} c_0(s, t_1, t_2) = \frac{1}{2}\Big[&\| \bar{\sigma}_0(s, t_1, t_2) - s \|^2 \\ &+ \| \bar{\zeta}_1(s, t_1, t_2) - t_1 \|^2 \\ &+ \| \bar{\zeta}_2(s, t_1, t_2) - t_2 \|^2\Big] \end{aligned} \tag{13}$$

$$\begin{aligned} c_1(s, t_1, t_2) = \frac{1}{2}\Big[&\| \bar{\sigma}_1(s, t_1, t_2) - s \|^2 + \\ &\| a_1(\sigma_1(s, t_1, t_2), l_1(s, t_1, t_2), t_1) - t_1 \|^2\Big] \end{aligned} \tag{14}$$

$$\begin{aligned} c_2(s, t_1, t_2) = \frac{1}{2}\Big[&\| \bar{\sigma}_2(s, t_1, t_2) - s \|^2 + \\ &\| a_2(\sigma_2(s, t_1, t_2), l_2(s, t_1, t_2), t_2) - t_2 \|^2\Big] \end{aligned} \tag{15}$$

We call this version CO_2, where the subscript "2" refers to the fact that the c_i are sums of squares.

Simple examples of optimization problems formulated as Collaboration Optimization can be found in [1].

Optimization by Linear Decomposition

Optimization by Linear Decomposition (OLD) [21–24], maintains interdisciplinary consistency at the system level while seeking to minimize the violation of the disciplinary design constraints at the subsystem level.

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. 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:

$$\begin{array}{ll} \min\limits_{l_1} & c_1(s, l_1, t_1, t_2) \\ \textbf{s. t.} & t_1 = a_1(s, l_1, t_2), \end{array} \tag{16}$$

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

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

In the disciplinary subproblem (16), 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 $a_1(s, l_1, t_2) - t_1 = 0$.

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:

$$\begin{array}{ll} \min\limits_{l_2} & c_2(s, l_2, t_1, t_2) \\ \textbf{s. t.} & t_2 = a_2(\sigma_2, l_2, t_1). \end{array} \tag{17}$$

Again, a_2 is computed via the disciplinary analysis

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

The subproblems (16)–(17) 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 (16)–(17) defines the system-level consistency constraints c_i. The resulting system-level problem is

$$\begin{array}{ll} \min\limits_{s,t_1,t_2} & f(s, a_1(s, \bar{l}_1(s,t_1,t_2), t_2), a_2(s, \bar{l}_2(s,t_1,t_2), t_1)) \\ \textbf{s. t.} & g_0(s, l_1, a_1(s, \bar{l}_1(s,t_1,t_2), a_2(s, \bar{l}_2(s,t_1,t_2), t_1)) \geq 0 \\ & c_1(s, \bar{l}_1(s,t_1,t_2), t_1, t_2) \leq 0 \\ & c_2(s, \bar{l}_2(s,t_1,t_2), t_1, t_2) \leq 0. \end{array} \tag{18}$$

One choice of discrepancy function is

$$c_1(s,l_1,t_1,t_2) = \sum_j (\min(0, g_1^j(s,l_1,a_1(s,l_1,t_2))))^2 \tag{19}$$

$$c_2(s,l_2,t_1,t_2) = \sum_j (\min(0, g_2^j(s,l_2,a_2(s,l_2,t_1))))^2. \tag{20}$$

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 $c_i = 0$.

OLD has been proposed in connection with an algorithm for its solution (e.g., [?, 23]). 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, [?, 23] 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. This algorithm will be subject to the analytical features of OLD as would be all algorithms that rely on linearlization of the constraints, such SLP or SQP.

Relationship among the formulations

Approaches to MDO problems are generally based on techniques for eliminating variables from the original problem. The variables are eliminated by enforcing various subsets of the constraints in different ways. In [25], we say that an MDO formulation is *closed* with respect to a given set of constraints if the formulation—rather than an optimization algorithm for its solution—assumes that these constraints are satisfied at every iteration of the optimization. If the formulation does not necessarily assume that a set of constraints is satisfied, we say that that formulation is *open* with respect to the set of constraints.

For instance, consider conventional optimization applied to the FIO formulation. We perform a multidisciplinary analysis at each step. This corresponds to maintaining closure of all the disciplinary analysis constraints in (3).

All the approaches we discuss here can be viewed from the perspective of eliminating various subsets of variables from the SAND formulation. For instance, if we begin with a SAND formulation but require that our designs satisfy the equality constraints representing the multidisciplinary analysis. Begin with the following SAND formulation,

$$\begin{array}{ll} \min\limits_{s,l_1,l_2,t_1,t_2} & f(s,t_1,t_2) \\ \textbf{s. t.} & g_1(s,l_1,t_1) \geq 0 \\ & g_2(s,l_2,t_2) \geq 0 \\ & t_1 = a_1 \\ & t_2 = a_2 \\ & a_1 = A_1(s,l_1,t_2) \\ & a_2 = A_2(s,l_2,t_1). \end{array} \tag{21}$$

If we require that at every iteration a_1 and a_2 satisfy the last two equality constraints,

$$\begin{aligned} a_1 - A_1(s,l_1,t_2) &= 0 \\ a_2 - A_2(s,l_2,t_1) &= 0, \end{aligned}$$

the we obtain the DAO formulation (7). In this reduction we are using these equality constraints to eliminate the analysis outputs a_1, a_2 as independent variables from the optimization problem.

If, in addition, we eliminate t_1, t_2 as independent variables from (21) by always requiring that

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

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 (16)–(17). Thus, in OLD, multidisciplinary analysis is performed at each iteration.

CO, on the other hand, eliminates the disciplinary design variables l_1, l_2 in DAO via (10)–(11), but does not eliminate the coupling variables t_1, t_2. Like DAO, the multidisciplinary analysis is enforced via the system-level constraints.

Modularity in implementation

In this section we illustrate how the computational modules needed for optimization algorithms, particularly the elements needed for sensitivity calculations, can be implemented autonomously by discipline. Moreover, one can reconfigure the same set of computational elements to implement one or other of the formulations discussed previously. This suggests that, if properly implemented, all of the formulations we have discussed require roughly the same amount of work to implement, and much of the implementation can be done autonomously in the individual disciplines.

For instance, consider the sensitivity information required to apply an optimization algorithm to DAO. We will show that the same pieces are needed to implement optimization for the fully integrated approach. Optimization of DAO requires the following sensitivities. For the objective $f_{DAO}(s, t_1, t_2)$, we need

$$\frac{\partial f}{\partial s}, \frac{\partial f}{\partial t_1}, \frac{\partial f}{\partial t_1} \tag{22}$$

For the design constraints $g_1(s, l_1, t_1)$ and $g_2(s, l_2, t_2)$ we need

$$\frac{\partial g_1}{\partial s}, \frac{\partial g_1}{\partial l_1}, \frac{\partial g_1}{\partial t_1} \tag{23}$$

and

$$\frac{\partial g_2}{\partial s}, \frac{\partial g_2}{\partial l_2}, \frac{\partial g_2}{\partial t_2}. \tag{24}$$

Finally, for the consistency constraints $t_1 - A_1(s, l_1, t_2)$ and $t_2 - A_2(s, l_2, t_1)$ we need

$$\frac{\partial A_1}{\partial s}, \frac{\partial A_1}{\partial l_1}, \frac{\partial A_1}{\partial t_2} \tag{25}$$

and

$$\frac{\partial A_2}{\partial s}, \frac{\partial A_2}{\partial l_2}, \frac{\partial A_2}{\partial t_1}. \tag{26}$$

In FIO approach, we need to compute the sensitivities of the objective

$$f_{FIO}(s, l_1, l_2) = f(s, a_1(s, l_1, l_2), a_2(s, l_1, l_2)).$$

By the chain rule,

$$\begin{aligned}
\frac{\partial f_{FIO}}{\partial s} &= \frac{\partial f}{\partial s} + \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial s} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial s} && (27)\\
\frac{\partial f_{FIO}}{\partial l_1} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_1} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_1} && (28)\\
\frac{\partial f_{FIO}}{\partial l_2} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_2} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_2}. && (29)
\end{aligned}$$

We compute the derivatives of a_1 and a_2 by implicit differentiation of the multidisciplinary analysis equations

$$\begin{aligned}
a_1 - A_1(s, l_1, a_2) &= 0\\
a_2 - A_2(s, l_2, a_1) &= 0.
\end{aligned}$$

This yields

$$\begin{pmatrix} I & -\frac{\partial A_1}{\partial a_2} \\ -\frac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \frac{\partial a_1}{\partial s} \\ \frac{\partial a_2}{\partial s} \end{pmatrix} = - \begin{pmatrix} \frac{\partial A_1}{\partial s} \\ \frac{\partial A_2}{\partial s} \end{pmatrix}, \tag{30}$$

$$\begin{pmatrix} I & -\frac{\partial A_1}{\partial a_2} \\ -\frac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \frac{\partial a_1}{\partial l_1} \\ \frac{\partial a_2}{\partial l_1} \end{pmatrix} = - \begin{pmatrix} \frac{\partial A_1}{\partial l_1} \\ 0 \end{pmatrix}, \tag{31}$$

and

$$\begin{pmatrix} I & -\frac{\partial A_1}{\partial a_2} \\ -\frac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \frac{\partial a_1}{\partial l_2} \\ \frac{\partial a_2}{\partial l_2} \end{pmatrix} = - \begin{pmatrix} 0 \\ \frac{\partial A_2}{\partial l_2} \end{pmatrix} \tag{32}$$

which must be solved for the sensitivities of a_1 and a_2 with respect to the design variables (s, l_1, l_2). (These are discussed as the "generalized sensitivity equations" in [26]).

Now compare the quantities required for the FIO sensitivity calculations (27)–(32) with those listed in (22)–(26) as required for the DAO sensitivity calculations. We see that these are the same. Moreover, these constituent pieces can be implemented for both approaches in a manner that respects disciplinary autonomy, *if* one starts with the non-integrated pieces of the multidisciplinary analysis as in the DAO formulation. These pieces are then integrated differently in DAO and FIO; however, the disciplinary building blocks are the same.

A similar examination of the sensitivities of the design constraints g_1 and g_2 in the FIO approach leads to the same conclusion. The computational pieces needed to implement optimization for the FIO can be assembled from those needed to implement optimization for the DAO formulation.

Likewise, if one examines the sensitivities that must be computed for the disciplinary subproblems in either CO or OLD, one finds that they involve the pieces in (22)–(26). The sensitivities of the system-level constraints are then computed via post-optimality sensitivity analysis of solutions of the disciplinary subproblems. This requires only computational components that are present in the DAO formulation, as well.

Algorithmic interactions

In the previous section we saw how one could, in principle, rearrange the computational components associated with the optimization of one formulation to obtain the pieces need to implement another. Of course, this may require significant effort to do. In this section we discuss how for some of the formulations, minor changes in an optimization algorithm designed for its solution may yield an optimization algorithm for the solution of an alternative formulation. This can make it straightforward to pass between formulations and facilitates the use of hybrid approaches, e.g., approaches that may use one formulation far from solutions and another close to solutions.

As an example, we examine the DAO, FIO, and SAND formulations, and for simplicity, consider only the constraints formed by the disciplinary analyses and interdisciplinary couplings. Thus, the simplified FIO formulation is

$$\min_{x} \quad f_{FIO}(x) \equiv f(x, a_1(x), a_2(x)), \qquad (33)$$

where, given x, we solve the multidisciplinary analysis system

$$\begin{pmatrix} \tilde{A}_1(x) \\ \tilde{A}_2(x) \end{pmatrix} = \begin{pmatrix} a_1 - A_1(x, a_1(x), a_2(x)) \\ a_2 - A_2(x, a_1(x), a_2(x)) \end{pmatrix} = 0. \qquad (34)$$

The simplified SAND formulation is

$$\begin{array}{ll} \min\limits_{x,a_1,a_2} & f_{SAND}(x, a_1, a_2) \equiv f(x, a_1, a_2) \\ \textbf{s. t.} & \tilde{A}_1(x, a_1, a_2) = 0 \\ & \tilde{A}_2(x, a_1, a_2) = 0. \end{array} \qquad (35)$$

Finally, the simplified DAO formulation is

$$\begin{array}{ll} \min\limits_{x,a_1,a_2,t_1,t_2} & f_{DAO}(x, a_1, a_2) \\ \textbf{s. t.} & t_1 - a_1(x, t_1, t_2) = 0 \\ & t_2 - a_2(x, t_1, t_2) = 0. \end{array} \qquad (36)$$

Let W_i be the matrix representing the basis of the null-space associated with the derivative of the block A_i. Then, relying on implicit differentiation and the derivations in [4], we note the following relationship among the sensitivities required for the three methods:

- Suppose, (x, a) is feasible with respect to (34). Then the (projected) gradients at (x, a) of the FIO and SAND formulations are related by

  $$\nabla_x f_{FIO}(x) = W_{SAND}^T(x, a) \nabla_{x,a} f_{SAND}(x, a),$$

  where W_{SAND} denotes a particular basis for the null-space of $\nabla \tilde{A}^T$ in the SAND approach.

- Suppose that (x, a) is feasible with respect to (36). Then

  $$W_{DAO}^T \nabla_{x,a} f_{DAO}(x, a) = W_{SAND}^T(x, a) \nabla_{x,a} f_{SAND}(x, a).$$

One can then use these relationships to implement a reduced-basis optimization algorithm for the three formulations with minimal modifications. Here we only sketch the conceptual algorithm.

Consider one step of a reduced-basis algorithm for the SAND formulation:

1. Construct a local model of the Lagrangian about the current design.
2. Take a substep to improve feasibility.
3. Subject to improved feasibility, take a substep to improve optimality.
4. Set the total step to the sum of the substeps, evaluate and update.

Performing the multidisciplinary analysis (34) after step 4 yields a corresponding algorithm for FIO. Solving the disciplinary equations as in DAO, we obtain an algorithm for DAO. Thus, passing between algorithms for distinct formulations is a straightforward step.

Incorporating the inequality constraints into this procedure is more involved and will be considered elsewhere. Here we wish to emphasize the general principle: by judicious implementation of the disciplinary sensitivity components, one can impart a large degree of flexibility to portions of computation.

Concluding remarks

In this paper, we discussed the requisite computational components (the function and derivative information) for the implementation of optimization for several MDO formulations. The amount of direct and auxiliary information that has to be implemented differs from one formulation to another. For instance, distributed optimization approaches optimize with respect to local variables at the subproblem level and with respect to the global variables and interdisciplinary couplings at the system level, while distributed analysis approaches optimize with respect to the complete set of variables at the system level. However, the problem structure of MDO ensures that similar sensitivity information—and hence the implementation effort—will be required for all formulations.

Moreover, if the disciplinary sensitivity information is implemented appropriately, one can transfer sensitivity information among several of the formulations in a straightforward manner and with minimal expense, thus enabling experimentation with a variety of problem formations. We gave an example of such re-use of information with minimal expense for the DAO and FIO formulations. While DAO formulation does not exhibit the same degree of execution autonomy as do the bilevel optimization methods we discussed (CO and OLD), it does exhibit an autonomy of problem integration or implementation and can be solved by conventional optimization algorithms robustly and efficiently.

As a matter of methodology, we caution against implementing the multidisciplinary analysis in an obvious manner (i.e., simply via a fixed-point iteration), without consideration of later integration into an optimization problem. The multidisciplinary analysis, if implemented in a modular fashion, can be an end in itself, but can also be used in a variety of optimization problem formulations.

In summary, when examining candidate problem formulations and optimization algorithms, re-use of disciplinary information and the flexibility in formulation and solution should be considered.

References

[1] 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.

[2] Natalia M. Alexandrov and Robert Michael Lewis. An analysis of some bilevel approaches to multidisciplinary design optimization. Technical report, ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia 23681-2199, June 2000.

[3] Natalia M. Alexandrov and Robert Michael Lewis. Analytical and computational properties of distributed approaches to MDO. AIAA-2000-4718, 2000. Presented at the Eighth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

[4] R. M. Lewis. Practical aspects of variable reduction formulations and reduced basis algorithms in multidisciplinary design optimization. In N.M. Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State-of-the-Art*, Philadelphia, 1997. SIAM.

[5] J. Sobieszczanski-Sobieski and R. T. Haftka. Multidisciplinary aerospace design optimization: Survey of recent developments. *Structural Optimization*, 14(1):1–23, August 1997.

[6] 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.

[7] Raphael T. Haftka, Zafer Gürdal, and Manohar P. Kamat. *Elements of structural optimization*. Kluwer Academic Publishers, Dordrecht, 1990.

[8] 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.

[9] R.D. Braun. *Collaborative Optimization: An architecture for large-scale distributed design*. PhD thesis, Stanford University, May 1996. Department of Aeronautics and Astronautics.

[10] R. D. Braun, A. A. Moore, and I. M. Kroo. Collaborative approach to launch vehicle design. *Journal of Spacecraft and Rockets*, 34:478–486, July 1997.

[11] 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.

[12] Ian Sobieski and Ilan Kroo. Aircraft design using collaborative optimization. AIAA paper 96-0715, presented at the 34th AIAA Aerospace Sciences Meeting, Reno, Nevada, Jan. 15-18, 1996, 1996.

[13] 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.

[14] 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.

[15] 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.

[16] H. M. Adelman, J. L. Walsh, and J. I. Pritchard. Recent advances in integrated multidisciplinary optimization of rotorcraft, AIAA paper AIAA-92-4777-CP. In *Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, pages 710–721, September 1992.

[17] Jaroslaw Sobieszczanski-Sobieski. Two alternative ways for solving the coordination problem in multilevel optimization. *Structural Optimization*, 6:205–215, 1993.

[18] 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.

[19] 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.

[20] J.L. Walsh, K.C. Young, J.I. Pritchard, H.M. Adelman, and W.R. Mantay. Integrated aerodynamic/dynamic/structural optimization of helicopter rotor blades using multilevel decomposition. Technical Report NASA TP 3465, NASA Langley Research Center, January 1995.

[21] Jaroslaw Sobieszczanski-Sobieski. A linear decomposition method for large optimization problems—blueprint for development. Technical Report TM 83248, NASA, 1982.

[22] Jaroslaw Sobieszczanski-Sobieski, Benjamin James, and Augustine Dovi. Structural optimization by multilevel decomposition. In *Proceedings of the AIAA/ASME/ASCE/AHS 24th Structures, Structural Dynamics, and Materials Conference*, May 1983. AIAA Paper 83-0832-CP.

[23] Jaroslaw Sobieszczanski-Sobieski, Benjamin B. James, and Michael F. Riley. Structural sizing by generalized, multilevel optimization. *AIAA Journal*, 25(1):139–145, January 1987.

[24] Jaroslaw Sobieszczanski-Sobieski. Optimization by decomposition: A step from hierarchic to non-hierarchic systems. Technical Report TM 101494, NASA, Hampton, Virginia, September 1988.

[25] N. M. Alexandrov and R. M. Lewis. Comparative properties of collaborative optimization and other approaches to MDO. In *Engineering Design Optimization*. MCB University Press, 1999.

[26] Jaroslaw Sobieszczanski-Sobieski. Sensitivity of complex, internally coupled systems. *American Institute of Aeronautics and Astronautics (AIAA) Journal*, 28(1):153–160, 1990.