



**AIAA 2003-3431**  
**DYNAMICALLY RECONFIGURABLE**  
**APPROACH TO**  
**MULTIDISCIPLINARY PROBLEMS**

Natalia M. Alexandrov

*NASA Langley Research Center, Hampton, Virginia*

Robert Michael Lewis

*College of William & Mary, Williamsburg, Virginia*

# DYNAMICALLY RECONFIGURABLE APPROACH TO MULTIDISCIPLINARY PROBLEMS

Natalia M. Alexandrov\*

*NASA Langley Research Center, Hampton, Virginia*

Robert Michael Lewis†

*College of William & Mary, Williamsburg, Virginia*

The complexity and autonomy of the constituent disciplines and the diversity of the disciplinary data formats make the task of integrating simulations into a multidisciplinary design optimization problem extremely time-consuming and difficult. We propose a dynamically reconfigurable approach to MDO problem formulation wherein an appropriate implementation of the disciplinary information results in basic computational components that can be combined into different MDO problem formulations and solution algorithms, including hybrid strategies, with relative ease. The ability to re-use the computational components is due to the special structure of the MDO problem. We believe that this structure can and should be used to formulate and solve optimization problems in the multidisciplinary context. The present work identifies the basic computational components in several MDO problem formulations and examines the dynamically reconfigurable approach in the context of a popular class of optimization methods. We show that if the disciplinary sensitivity information is implemented in a modular fashion, the transfer of sensitivity information among the formulations under study is straightforward. This enables not only experimentation with a variety of problem formations in a research environment, but also the flexible use of formulations in a production design environment.

## Introduction

Design optimization is not equivalent to nonlinear programming (NLP). The design problem is much more complicated and amorphous than the most difficult problems amenable to current NLP techniques. Recent years have seen the beginnings of promising alternative approaches to design optimization; see, e.g., Tatting and Gürdal.<sup>1</sup> Nonetheless, ever since its early application in structural optimization (see an extensive bibliography in Haftka and Gürdal<sup>2</sup>), NLP has been the most successful paradigm available to-date for solving simulation-based design problems, having gradually made its way into such areas as CFD-based design<sup>3,4</sup> and Multidisciplinary Design Optimization<sup>5,6</sup> (MDO). In this paper, we examine the formulation of MDO problems as optimization problems and for this purpose we view MDO as comprising a subset of the broader design problem that can be expressed in terms of NLP.

It is interesting to note that, despite great promise, state-of-the-art—or even systematic—NLP methods are not widely used for *realistic* MDO problems. In fact, relatively successful applications of systematic optimization techniques can be usually observed in highly controlled, academic environments, where researchers operate on sets of a few analyses whose readily available source code is open to modification by the participants. In contrast, most of the attempts to use

NLP to its full extent and benefit we have observed in a variety of more complex multidisciplinary settings have so far failed. In practice, heroic software integration efforts directed at implementing a version of the multidisciplinary analysis (MDA) via a fixed-point iteration usually consume most of the available project time and effort. This frequently leaves no resources for computing reliable derivatives, let alone experimenting with optimization formulations and software. The situation is exacerbated by the need to compute MDA-based objectives, constraints, and sensitivities *automatically* over relatively large regions of the design space. If solving an application problem is an end in itself, attempting the solution of a particular formulation may lead to an impasse. In this case, attempting an implementation of another formulation involves “unscrambling” the codes from the existing formulation and expending a considerable effort to implement a new one. Resource limitations usually make such an activity prohibitive. If, in addition to solving the problem, the other main project objective is to develop new design methods, again, limited resources rarely leave room for experimentation once the project time has been consumed by the conventional implementation of MDA. This leads to one-of-a-kind implementations rather than flexible applications of methodologies suitable to specific classes of problems.

Solving MDO problems is further complicated by such features as the problematic convergence of the constituent analyses, the presence of discrete variables, and the unavailability or high cost of the disciplinary derivatives, among other difficulties. But even assum-

\*Senior Member AIAA, Senior Research Scientist.

†Member AIAA, Assistant Professor. This research was supported by the National Aeronautics and Space Administration under NASA Grant NCC-1-02029.

ing sufficient smoothness and the availability of the derivative information, the synthesis of realistic MDO problems presents a challenge.

Engineering frameworks that have appeared over the past decade are meant to assist with MDO problem synthesis. Indeed, they make the task of data exchange among the components of the design problem easier once the contributing simulations are wrapped for inclusion in the framework. Computational frameworks appear to be of most benefit in a production environment, in which the properties of the design problem are relatively stable. But in a research environment, the current design frameworks usually lack the capabilities necessary to make them flexible exploratory tools. An examination of computational frameworks is outside the scope of this paper. We will say only that, in the context of the present work, design frameworks that we have observed do not provide tools for analyzing the design problem in terms of appropriate formulations, nor for matching the formulation with appropriate optimization algorithms. The task of reconfiguring problem formulations does not appear to be made easier by the existing frameworks. However, computational problem solving environments are evolving and these shortcomings may be less of an issue in the future.

With the aim of easing MDO problem synthesis and solution via NLP techniques, we have recently examined, both experimentally<sup>7</sup> and analytically,<sup>8,9,10,11</sup> the influence of a number of MDO problem formulations on the computational tractability of the resulting problems and have observed how the analytical features of MDO problem formulation, such as the degree of subsystem autonomy, directly influence the ability of numerical algorithms to solve the problem reliably and efficiently. Given the high cost of simulation-based function evaluations, it is especially important in MDO and single-discipline simulation-driven optimization to arrive at problem formulations that lead to efficient and reliable solution strategies.

Engineering practitioners of NLP have always understood the importance of optimization problem formulation – many interesting efforts over the past three decades address various ways to pose complex problems.<sup>12,13,14,15,16,17,18,19,20,21,22,23,24,25,26</sup> With some notable exceptions,<sup>13,19,22</sup> most efforts have been motivated by the philosophy we term the *structural perspective*. In the structural perspective the physical structure of the problem and organizational considerations are the main driving forces behind the choice of the mathematical problem formulation. The main drawback of the exclusive focus on physics and organization is that it may lead to nonlinear programs which are difficult to solve,<sup>9,10</sup> given the state of the art in NLP. In contrast, the *algorithmic perspective* we propose attempts to make use of the physical structure of the problem and conform to

organizational features as much as possible without sacrificing solubility by available algorithms. At times, solubility by the available algorithms may mean that we cannot satisfy all the desiderata, e.g., disciplinary autonomy, suggested by the structural perspective. In this case, the practitioner must decide between the relative merits of the resulting formulations. Of course, as optimization methods evolve, a formulation that was difficult or impossible to solve may become amenable to a newly developed algorithm. The algorithmic perspective allows us to formulate problems in the best possible way, given the state of the art in optimization algorithms, and it also points out the obstacles that must be addressed to enable solving the currently difficult formulations, such as general nonlinear multilevel problems.

Following the algorithmic perspective, we have arrived at the following conclusions:

- All MDO problem formulations are related and, in fact, share the same basic computational components comprising function and sensitivity information needed to solve the resulting optimization problems.
- Although the task of making the disciplinary analyses “speak to one another” is difficult and remains unchanged for all problem formulations, MDO problem synthesis and solution can be significantly eased with an appropriate implementation of the basic computational components.

In light of these observations, we propose a dynamically reconfigurable approach to MDO problem formulation that may be summarized as follows: an appropriate implementation of the disciplinary information results in basic computational components that can be combined into different MDO problem formulations and solution algorithms, including hybrid strategies, with relative ease. The ability to re-use the computational components is due to the special structure of the MDO problem. We believe that this structure can and should be used to formulate and solve optimization problems in multidisciplinary context.

This approach was initiated in Lewis<sup>27</sup> for unconstrained MDO problems (simulations are viewed as equality constraints in some of the problem formulations). Here we examine this approach for generally constrained MDO problems and continue the development of the ideas in application to specific algorithms. In particular, we identify the basic computational components in several MDO problem formulations and examine the dynamically reconfigurable approach in the context of barrier-SQP optimization. We show that if the disciplinary sensitivity information is implemented in a modular fashion, the transfer of sensitivity information among the formulations under study is straightforward. This enables not only experimentation with a variety of problem formations in a research

environment, but also the flexible use of formulations in a production design environment.

## The Origins of Reconfigurability

The capacity for reconfigurability stems from the relation among MDO problem formulations. We demonstrate this on a model problem.

Following the conventions of our earlier work,<sup>8,9,10</sup> we present this discussion for a two-discipline MDO problem. The two disciplines may represent, for instance, the aeroelastic interaction between aerodynamics (discipline 1) and structural response (discipline 2) for a wing in steady-state flow. We limit our discussion to three formulations: simultaneous analysis and design (SAND),<sup>2</sup> the straightforward fully integrated optimization (FIO) formulation, and distributed analysis optimization (DAO).<sup>9,10,22</sup> We refer the reader to that earlier work for the detailed development, references to the origins, and comparative properties of the three formulations.

We assume that each subsystem of the two-discipline model problem is based on a disciplinary simulation. Each disciplinary analysis  $A_i$  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 that are outputs of the other discipline. 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  $i$ . In our aeroelastic example, for instance, the input  $a_1$  from structures to aerodynamics would include the wing shape, while the input  $a_2$  from aerodynamics to structures would include the aerodynamic loads. In our notation,

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

We start with the SAND formulation. The distinctive feature of SAND is the treatment of the full MDA purely as equality constraints. The version of SAND in (1) below, also introduces auxiliary variables  $\sigma_0$ ,  $\sigma_1$ ,  $\sigma_2$ , to allow the disciplinary constraints the freedom of choosing iterates most appropriate to that discipline in the course of optimization. To ensure consistency at the solution among the shared design variables, corresponding *consistency constraints* are also introduced. Similarly, auxiliary variables  $t_1$ , and  $t_2$  ensure consistency among the analysis outputs,

Vectors  $c_i^T$  and  $c_i^E$  represent the disciplinary inequality and equality constraint systems, respectively. Vectors  $c_0^T$  and  $c_0^E$  denote system-level constraint vectors

that have contributions from both disciplines.

$$\begin{aligned} & \underset{s, \sigma_0, \sigma_1, \sigma_2, l_1, l_2, a_1, a_2, t_1, t_2}{\text{minimize}} && f(s, t_1, t_2) \\ & \text{subject to} && c_0^T(\sigma_0, t_1, t_2) \geq 0 \\ & && c_1^T(\sigma_1, l_1, t_1) \geq 0 \\ & && c_2^T(\sigma_2, l_2, t_2) \geq 0 \\ & && c_0^E(\sigma_0, t_1, t_2) = 0 \\ & && c_1^E(\sigma_1, l_1, t_1) = 0 \\ & && c_2^E(\sigma_2, l_2, t_2) = 0 \\ & && t_1 = a_1 \\ & && t_2 = a_2 \\ & && a_1 = A_1(\sigma_1, l_1, t_2) \\ & && a_2 = A_2(\sigma_2, l_2, t_1) \\ & && \sigma_0 = s \\ & && \sigma_1 = s \\ & && \sigma_2 = s. \end{aligned} \tag{1}$$

Some of the variables are unnecessary for the SAND formulation, but we introduce them to draw connections with other formulations.

Other MDO formulations may be viewed as derived from the SAND formulation by closing a particular set of constraints. In particular, if we use the equality constraints

$$\begin{aligned} a_1 - A_1(\sigma_1, l_1, t_2) &= 0 \\ a_2 - A_2(\sigma_2, l_2, t_1) &= 0, \end{aligned}$$

to eliminate the analysis outputs  $a_1, a_2$  as independent variables from the optimization problem, we obtain the DAO formulation. If, in addition, we eliminate  $\sigma_0, \sigma_1, \sigma_2, t_1, t_2$  as independent variables from SAND by always requiring the satisfaction of the multidisciplinary consistency conditions

$$\begin{aligned} t_1 &= a_1(\sigma_1, l_1, t_2) \\ t_2 &= a_2(\sigma_2, l_2, t_1) \\ \sigma_0 &= s \\ \sigma_1 &= s \\ \sigma_2 &= s, \end{aligned}$$

then we obtain the FIO approach.

We first stated SAND in the form (1) to give the reader the idea of all the degrees of freedom afforded by this formulation. Now we remove some of them, as well as some of the corresponding consistency constraints and the system-level constraints, for ease of exposition and without loss of generality. To simplify notation, we also leave the general equality constraints out of the remaining discussion – they can be handled in the same way as the simulation-based equalities. We thus restrict our attention to the following simplified SAND

formulation:

$$\begin{aligned}
& \min_{s, l_1, l_2, a_1, a_2, t_1, t_2} && f(s, t_1, t_2) \\
& \text{s.t.} && c_1(s, l_1, t_1) \geq 0 \\
& && c_2(s, l_2, t_2) \geq 0 \\
& && t_1 = a_1 \\
& && t_2 = a_2 \\
& && a_1 = A_1(s, l_1, t_2) \\
& && a_2 = A_2(s, l_2, t_1),
\end{aligned} \tag{2}$$

where we now drop the superscript denoting the inequality constraints.

A DAO formulation is

$$\begin{aligned}
& \min_{s, l_1, l_2, t_1, t_2} && f(s, t_1, t_2) \\
& \text{s.t.} && c_1(s, l_1, t_1) \geq 0 \\
& && c_2(s, l_2, t_2) \geq 0 \\
& && t_1 = a_1(s, l_1, t_2) \\
& && t_2 = a_2(s, l_2, t_1),
\end{aligned} \tag{3}$$

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

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

Because we left the last three constraints of (1) out of the formulation, we may find it more difficult to find points feasible with respect to both analysis and design constraints  $c_i$ . We comment on that benefit of introducing the auxiliary design variables  $\sigma_1$  and  $\sigma_2$  in more detail elsewhere.<sup>9,10</sup>

Finally, we consider the corresponding FIO formulation,

$$\begin{aligned}
& \min_{s, l_1, l_2} && f(s, t_1(s, l_1, l_2), t_2(s, l_1, l_2)) \\
& \text{s.t.} && c_1(s, l_1, t_1(s, l_1, l_2)) \geq 0 \\
& && c_2(s, l_2, t_2(s, l_1, l_2)) \geq 0
\end{aligned} \tag{4}$$

where we compute  $t_1(s, l_1, l_2)$  and  $t_2(s, l_1, l_2)$  by solving the multidisciplinary analysis

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

The preceding can be simplified, of course, to

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

but we will use the expanded version to emphasize the relationship to the SAND formulation.

We can view the DAO and FIO formulations (3) and (4) as having been obtained from the SAND formulation (2) by closing some of the multidisciplinary consistency (analysis) equality constraints in the SAND formulation. In both DAO and FIO this eliminates

some of the independent variables from the SAND formulation.

Other problem formulations attempt to eliminate local design variables by solving disciplinary optimization subproblems, as in Collaborative Optimization<sup>25,28,26,29</sup> and Optimization by Linear Decomposition.<sup>14,16,18</sup> Arriving at specific reconfigurable relations needed for these formulations would require more work, but the basic computational components would be the same.

The relationships (8) and (9) introduced in the next section, in turn, tell us that the sensitivities of the objective and constraints in the DAO and FIO formulations are related to the sensitivities in the SAND formulations via *variable reduction*. This point is central to our perspective, since it means that computational components implemented for the SAND formulation can be reconfigured to yield the computational components needed for other formulations. We will make this point clearer by considering specific algorithms for the solution of the SAND, DAO, and FIO formulations.

## The Reconfigurable Approach

In this section, we develop the computational components for three problem formulations and demonstrate the reconfigurable approach within the context of one class of nonlinear programming algorithms.

We emphasize that, regardless of the algorithm and formulation, the basic computational components of functions and sensitivities required by algorithms remain unchanged. One has to show how to combine these components in various algorithmic settings to allow for a straightforward conversion among formulations. We choose to demonstrate the approach on the barrier sequential quadratic programming (SQP) class of methods because of the simplicity of inequality constraint handling in barrier-SQP methods. A specific instance of the class is shown within the reconfigurable framework.

Throughout the discussion it will become apparent that the sensitivity information required by large classes of optimization algorithms can be easily obtained by exploiting the structure of simulation-based computations. The same arguments apply to general nonlinear programming programs, but those based on PDE provide the necessary information in a natural, easily available form.

## The Reduced Gradient and Hessian

We begin with some standard results. A variable appearing as a subscript denotes the partial derivative with respect to that variable. Suppose  $\phi(x, v)$  is a function and let

$$\Phi(x) = \phi(x, v(x)), \tag{5}$$

where, given  $x$ ,  $v(x)$  is computed by solving a system of equations

$$S(x, v(x)) = 0. \tag{6}$$

We assume that  $S_v$  is invertible.

The derivatives of  $\phi$  and  $\Phi$  are related as follows. Let  $W$  be the *injection operator*

$$W = W(x, v) = \begin{pmatrix} I \\ -S_v^{-1}(x, v)S_x(x, v) \end{pmatrix}.$$

We refer to  $W^T$  as the *reduction operator*.

Define  $\lambda$  by

$$\lambda = \lambda(x, v) = -(S_v(x, v))^{-T} \nabla_v \phi(x, v) \quad (7)$$

and the Lagrangian  $L(x, v; \lambda)$  by

$$L(x, v; \lambda) = \phi(x, v) + \lambda^T S(x, v).$$

Then

$$\nabla_x \Phi(x) = W^T(x, v(x)) \nabla_{(x, v)} \phi(x, v(x)). \quad (8)$$

If  $\phi$  is real valued, the quantity on the right-hand side of (8) is known as the *reduced gradient*.<sup>30</sup> The quantity on the right-hand side of the following equation is called the *reduced Hessian* of the Lagrangian:

$$\nabla_{xx}^2 \Phi(x) = W^T \left( \nabla_{(x, v)}^2 \phi + \nabla_{(x, v)}^2 S \cdot \lambda \right) W, \quad (9)$$

where

$$\begin{aligned} W &= W(x, v(x)) \\ \nabla_{(x, v)}^2 \phi &= \nabla_{(x, v)}^2 \phi(x, v(x)) \\ \nabla_{(x, v)}^2 S \cdot \lambda &= \nabla_{(x, v)}^2 S(x, v(x)) \cdot \lambda(x, v(x)) \\ &= \sum_{i=1}^n \lambda_i \nabla_{(x, v)}^2 S_i. \end{aligned}$$

A derivation of these expressions can be found in a number of sources.<sup>30, 27</sup>

### A Barrier-SQP Approach

A barrier-SQP approach to (2), (3), and (4) treats the inequality constraints by adding them in a barrier term to the objective, while retaining the equality constraints explicitly. Although our presentation has sufficient detail to convey the ideas, it is more conceptual than detailed for reasons of brevity.

For the SAND formulation (2), the algorithm proceeds by solving a sequence of equality-constrained subproblems of the form:

$$\begin{aligned} \min_{s, l_1, l_2, t_1, t_2, a_1, a_2} \quad & F_{\text{SAND}}(s, l_1, l_2, t_1, t_2) \\ \text{s.t.} \quad & t_1 = a_1 \\ & t_2 = a_2 \\ & a_1 = A_1(s, l_1, t_2) \\ & a_2 = A_2(s, l_2, t_1), \end{aligned} \quad (10)$$

where

$$\begin{aligned} F_{\text{SAND}}(s, l_1, l_2, t_1, t_2) &= f(s, t_1, t_2) \\ &- \mu \left[ \sum_i \ln c_1^i(s, l_1, t_1) + \sum_j \ln c_2^j(s, l_2, t_2) \right]. \end{aligned}$$

The barrier subproblem for the DAO formulation is

$$\begin{aligned} \min_{s, l_1, l_2, t_1, t_2} \quad & F_{\text{DAO}}(s, l_1, l_2, t_1, t_2) \\ \text{s.t.} \quad & t_1 = a_1(s, l_1, t_2) \\ & t_2 = a_2(s, l_2, t_1), \end{aligned} \quad (11)$$

where

$$\begin{aligned} F_{\text{DAO}}(s, l_1, l_2, t_1, t_2) &= f(s, t_1, t_2) \\ &- \mu \left[ \sum_i \ln c_1^i(s, l_1, t_1) + \sum_j \ln c_2^j(s, l_2, t_2) \right], \end{aligned}$$

and the disciplinary responses  $a_1(s, l_1, t_2)$  and  $a_2(s, l_2, t_1)$  are computed via the disciplinary analyses:

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

Consider the relationship between the sensitivities in the SAND and DAO formulations. In the notation of the section on reduced derivatives, for the SAND formulation we have

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

The system of equations representing the constraints that are closed is

$$\begin{aligned} S_{\text{DAO}} &= S(s, l_1, l_2, t_1, t_2, a_1, a_2) = \\ &\begin{pmatrix} A_1(s, l_1, t_2) - a_1 \\ A_2(s, l_2, t_1) - a_2 \end{pmatrix} = 0. \end{aligned} \quad (12)$$

The injection matrix  $W_{\text{DAO}}$  is

$$W_{\text{DAO}} = \begin{pmatrix} I \\ -S_v^{-1} S_x \end{pmatrix}.$$

In this case,

$$S_v = \begin{pmatrix} \partial_{a_1} A_1 - I & 0 \\ 0 & \partial_{a_2} A_2 - I \end{pmatrix}$$

and

$$S_x = \begin{pmatrix} \partial_s A_1 & \partial_{l_1} A_1 & 0 & 0 & \partial_{t_2} A_1 \\ \partial_s A_2 & 0 & \partial_{l_2} A_2 & \partial_{t_1} A_2 & 0 \end{pmatrix},$$

where now we have taken to indicating partial derivatives with  $\partial$  to avoid double subscripts.

We then have

$$\nabla_{(s, l_1, l_2, t_1, t_2)} F_{\text{DAO}} = W_{\text{DAO}}^T \nabla_{(s, l_1, l_2, t_1, t_2, a_1, a_2)} F_{\text{SAND}}$$

and

$$\nabla_{(s, l_1, l_2, t_1, t_2)}^2 F_{\text{DAO}} = W_{\text{DAO}}^T \nabla_{(s, l_1, l_2, t_1, t_2, a_1, a_2)}^2 F_{\text{SAND}} W_{\text{DAO}}.$$

These relations show that the computational components needed to implement the barrier-SQP approach to the SAND formulation are those needed to implement the barrier-SQP approach to the DAO formulation.

A similar relationship exists between the barrier-SQP subproblem for the SAND formulation and that for the FIO formulation, given by

$$\min_{s, l_1, l_2} F_{\text{FIO}}(s, l_1, l_2) \quad (13)$$

where

$$F_{\text{FIO}}(s, l_1, l_2) = f(s, t_1(s, l_1, l_2), t_2(s, l_1, l_2)) - \mu \left[ \sum_i \ln c_1^i(s, t_1(s, l_1, l_2), t_2(s, l_1, l_2)) + \sum_j \ln c_2^j(s, t_1(s, l_1, l_2), t_2(s, l_1, l_2)) \right],$$

and we compute  $t_1(s, l_1, l_2)$  and  $t_2(s, l_1, l_2)$  by solving the multidisciplinary analysis

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

Relating the FIO formulation to the SAND formulation in the notation of the section on reduced derivatives, we have

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

The system of equations representing the constraints that are closed is

$$S_{\text{FIO}} = S(s, l_1, l_2, t_1, t_2, a_1, a_2) = \begin{pmatrix} t_1 - a_1 \\ t_2 - a_2 \\ A_1(s, l_1, t_2) - a_1 \\ A_2(s, l_2, t_1) - a_2 \end{pmatrix} = 0.$$

In this case,

$$S_v = \begin{pmatrix} I & 0 & -I & 0 \\ 0 & I & 0 & -I \\ 0 & \partial_{t_2} A_1 & \partial_{a_1} A_1 - I & 0 \\ \partial_{t_1} A_2 & 0 & 0 & \partial_{a_2} A_2 - I \end{pmatrix}$$

and

$$S_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_s A_1 & \partial_{l_1} A_1 & 0 \\ \partial_s A_2 & 0 & \partial_{l_2} A_2 \end{pmatrix},$$

For the injection matrix  $W_{\text{FIO}}$

$$W_{\text{FIO}} = \begin{pmatrix} I \\ -S_v^{-1} S_x \end{pmatrix}.$$

we have

$$\nabla_{(s, l_1, l_2)} F_{\text{FIO}} = W_{\text{FIO}}^T \nabla_{(s, l_1, l_2, t_1, t_2, a_1, a_2)} F_{\text{SAND}}$$

and

$$\nabla_{(s, l_1, l_2)}^2 F_{\text{FIO}} = W_{\text{FIO}}^T \nabla_{(s, l_1, l_2, t_1, t_2, a_1, a_2)}^2 F_{\text{SAND}} W_{\text{FIO}}.$$

The same computational components from the SAND formulation used in the DAO formulation are used in the FIO formulation.

### A Reduced Basis Approach to the Barrier-SQP Subproblem

The barrier-SQP algorithm we sketch is similar to that described in, e.g., Byrd et al.<sup>31</sup> For a specific choice of algorithm to solve the barrier-SQP subproblems, we can say even more about the relationship between the computational elements needed to solve the SAND, DAO, and FIO formulations. In this section we briefly discuss a reduced basis SQP algorithm.

The solution of the SAND barrier subproblem (10) is itself an iterative process, in which we successively compute approximate solutions to subproblems of the form

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} p^T H p + g^T p \\ \text{subject to} \quad & \nabla S^T p + S = 0, \end{aligned} \quad (14)$$

where  $H$  is an approximation to the Hessian of the Lagrangian and  $g$  is the gradient of the Lagrangian. Here  $x$  is the total set of independent variables, and  $p$  the step we take in the iterative process. Specific ways of solving this problem in connection with equality constraint minimization have been addressed in numerous publications.<sup>32,33,34,35,36,37</sup>

Given a step  $p_{LF}$  that is feasible with respect to the linearized equality constraints, and a basis  $Z$  for the nullspace of the Jacobian of the equality constraints, one could express  $p$  as

$$p = p_{LF} + Zq$$

for some vector  $q$ , and the SQP subproblem could be reduced to an unconstrained problem:

$$\text{minimize} \quad \frac{1}{2} q^T Z^T H Z q + (g + H p_{LF})^T Z^T q. \quad (15)$$

In a *reduced basis* approach to an equality constrained optimization problem we take advantage of a certain form of basis for the nullspace of the linearized constraints.<sup>30</sup> At each iteration of the optimization algorithm, we use the fact that

$$Z = \begin{pmatrix} I \\ -S_v^{-1} S_x \end{pmatrix}$$

is a basis for the nullspace of the Jacobian of  $S$ . As a linearly feasible point  $p_{LF}$  we may simply take

$$p_{LF} = \begin{pmatrix} 0 \\ -B^{-1} S \end{pmatrix}. \quad (16)$$

This corresponds to solving the linearized multidisciplinary analysis equations.

In terms of  $W$ , then, the model SQP subproblem (14) can be transformed to the unconstrained problem

$$\text{minimize } \frac{1}{2}q^T W^T H W q + (g + H p_{LF})^T W^T q.$$

We then set  $p = Wq$ .

#### **Implementation of the Reduced Basis Algorithm for the Different Formulations**

The relationship between the different formulations discussed in section on sensitivities means that it is possible to implement an optimization algorithm for the SAND formulation so that with a single modification we obtain an algorithm for either the FIO or DAO formulations.

In the discussion that follows, the subscript and superscript “ $c$ ” on the iterates denote quantities associated with the current iterate, and the subscript and superscript “ $+$ ” denote quantities associated with the next iterate. At each iteration, we approximately solve the subproblem

$$\begin{aligned} &\text{minimize} && \frac{1}{2}p^T H p + g^T p \\ &\text{subject to} && \|\nabla S^T p + S\| \leq \theta \\ &&& \|p\| \leq r, \end{aligned} \quad (17)$$

where  $H$  is an approximation to the current Hessian of the Lagrangian,  $g$  is the gradient of the Lagrangian,  $r$  is the current trust radius, and  $\theta$  is chosen to enforce sufficient decrease on  $\|\nabla S^T p + S\|^2$  over a trust-region of radius  $\beta r$ , where  $0 < \beta < 1$  is fixed. We use Fraction of Cauchy Decrease (FCD) for sufficient decrease here. FCD is a mild condition meaning that the model predicts at least a fraction of the decrease that would have been predicted by the steepest descent step within the trust region.

Algorithm 1 contains an outline of the solution procedure for the SAND formulation. The approximate solution of the subproblem is effected in two stages. The first stage, Step 4, is to take a step in the basic variables to improve linear feasibility, as in (16). The second stage, Step 5, is then to improve optimality subject to the constraint of not degrading the improved linear feasibility achieved by Step 4.

#### **Algorithm 1:** Reduced-basis algorithm for SAND

Initialization: Choose an initial  $(x_c, v_c)$ .

Until convergence, do {

1. Compute the multiplier

$$\lambda_{SAND} = -S_v^{-1} \nabla_v F_{SAND}.$$

2. Test for convergence.
3. Construct a local model of  $L$  about  $(x_c, v_c)$ .

4. Take a step  $p^{LF}$  to improve linear feasibility:

$$p^{LF} = \alpha \begin{pmatrix} 0 \\ -S_v^{-1} S \end{pmatrix}.$$

5. Subject to the improved linear feasibility, take a step to improve optimality:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}q^T W^T H W q + (g + H p_{LF})^T W^T q \\ &\text{subject to} && \|p_{LF} + Wq\| \leq r. \end{aligned}$$

6. Set  $p = (p_x, p_v) = p_{LF} + Wq$ .
7. Evaluate  $(x_+, v_+) = (x_c, v_c) + (p_x, p_v)$ .
8. Update  $(x_c, v_c)$ ,  $r$ , and other parameters.

Now suppose this algorithm is modified by adding an analysis step in which given design variables  $x_c$ , the multidisciplinary analysis problem  $S(x_c, v_c) = 0$  is solved.

#### **Algorithm 2:** Reduced basis algorithm for SAND formulation with a multidisciplinary analysis step

Initialization: Choose an initial  $x_c$ .

**Analysis:** Solve  $S_{FIO}(x_c, v_c(x_c)) = 0$  for  $v_c(x_c)$ .

Until convergence, do {

- 1-6. These steps remain unchanged.
7. **Analysis:** Solve  $S_{FIO}(x_+, v_+) = 0$  for  $v_+(x_+)$  and evaluate  $(x_+, v_+)$ .
8. This step remains unchanged.

Then it can be shown that this single modification, illustrated in Algorithm 2, yields a trust region algorithm for the FIO formulation.<sup>27</sup> Thus, simply by introducing an analysis step, the reduced basis barrier-SQP algorithm for the SAND formulation becomes a reduced basis barrier-SQP algorithm for the FIO formulation, without any re-implementation of sensitivities or other procedures.

If we add a partial analysis step to the algorithm for the SAND formulation, an analysis that enforces feasibility with respect to the block of constraints  $S_{DAO}$  from (13), as in Algorithm 3, then we obtain a reduced basis algorithm for the DAO formulation.

#### **Algorithm 3:** Reduced-basis algorithm for SAND formulation with an analysis step that eliminates $v = (a_1, a_2)$ .

Initialization: Choose an initial  $(x_c, v_c)$ .

**Analysis:** Solve  $S_{DAO}(x_c, v_c(x_c)) = 0$  for  $v_c(x_c)$ .

Until convergence, do {

- 1-6. These steps remain unchanged.



7. **Analysis:** Solve  $S_{\text{DAO}}(x_+, v_+) = 0$   
for  $v_+(x_+)$  and evaluate  $(x_+, v_+)$ .
  8. This step remains unchanged.
- }

Thus, by adding appropriate analysis steps to the reduced basis Algorithm 1 for the SAND formulation, we obtain reduced basis algorithms for the FIO and DAO formulations. This means that a properly implemented SAND barrier-SQP approach can be reconfigured to become a barrier-SQP approach for other formulations.

### Other Algorithms

The reconfigurable scheme outlined in the previous sections would work for other methods that handle the inequality constraints by adding a penalty term to the objective. The augmented Lagrangian method is an example of such an algorithm. We believe the approach will be valid for active set methods; however, proving this assertion will require further work because of the complexity of active set methods.

### Concluding Remarks

Experience with multidisciplinary problems points to the strong influence of the analytical features of MDO problem formulation on the ability of nonlinear programming algorithms to solve the problem reliably and efficiently. In our studies we maintain the algorithmic perspective, which takes into account the problem structure and the organizational requirements to the greatest possible extent without sacrificing solubility by optimization algorithms. At the same time, we realize that the organizational requirements may bring a practitioner of MDO to use methods that emphasize the structural perspective over the algorithmic one. Regardless of the approach, there is a clear need for flexible MDO problem implementation that combines the capability to formulate and re-formulate problems as needed with the choice of optimization algorithms appropriate for specific MDO formulations.

We believe that such a capability flows naturally from the fact that all MDO formulations need the same basic computational components and are related through the specific closure of constraints. Some combinations of MDO formulations and optimization algorithms contain an even richer structure that results in a user's ability to transform one formulation into another in the context of a specific algorithm with very little effort. We demonstrated this reconfigurable approach on the barrier-SQP class of methods and three MDO problem formulation.

Other problem formulations in other algorithmic contexts may require considerably more work to develop the easy reconfigurability. However, regardless of the formulation and algorithm, the computational components remain unchanged. Our current and future plans include developing an analysis of most

major classes of algorithms in combination with the promising MDO problem formulations that, we hope, will result in a set of guidelines for choosing appropriate problem formulations and the attendant optimization methods, given specific problem characteristics.

### References

- <sup>1</sup>Tatting, B. and Z. Gürdal, "Cellular Automata for Design of Two-Dimensional Continuum Structures," AIAA-2000-4832, 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2000.
- <sup>2</sup>Haftka, R. T. and Gürdal, Z., *Elements of Structural Optimization*, Kluwer Academic Publishers, Boston, 1993, 3rd edn.
- <sup>3</sup>Jameson, A., "Essential Elements of Computational Algorithms for Aerodynamic Analysis and Design," Tech. Rep. 97-68, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia, December 1997.
- <sup>4</sup>Newman, III, J. C., Taylor, III, A. C., Barnwell, R. W., Newman, P. A., and Hou, G. J.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87-96.
- <sup>5</sup>Sobieszcanski-Sobieski, J. and Haftka, R. T., "Multidisciplinary aerospace design optimization: Survey of recent developments," *Structural Optimization*, Vol. 14, No. 1, August 1997, pp. 1-23.
- <sup>6</sup>Alexandrov, N. M. and Hussaini, M. Y., editors, *Multidisciplinary Design Optimization: State of the Art*, Philadelphia, 1997, Society for Industrial and Applied Mathematics.
- <sup>7</sup>Alexandrov, N. M. and Kodiyalam, S., "Initial Results of an MDO Method Evaluation Study," 1998, AIAA Paper 98-4884.
- <sup>8</sup>Alexandrov, N. M. and Lewis, R. M., "Analytical and Computational Aspects of Collaborative Optimization," National Aeronautics and Space Administration, NASA/TM-210104-2000, April 2000.
- <sup>9</sup>Alexandrov, N. M. and Lewis, R. M., "Analytical and computational properties of distributed approaches to MDO," AIAA Paper 2000-4718, September 2000.
- <sup>10</sup>Alexandrov, N. M. and Lewis, R. M., "Algorithmic perspectives on problem formulations in MDO," AIAA Paper 2000-4719, 2000.
- <sup>11</sup>Alexandrov, N. M. and Lewis, R. M., "Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design," *AIAA Journal*, Vol. 40, No. 2, February 2002, pp. 301-309.
- <sup>12</sup>Schmit, Jr., L. A. and Ramanathan, R. K., "Multilevel approach to minimum weight design including buckling constraints," *AIAA Journal*, Vol. 16, No. 2, February 1978, pp. 97-104.
- <sup>13</sup>Thareja, R. and Haftka, R. T., "Numerical difficulties associated with using equality constraints to achieve multi-level decomposition in structural optimization," AIAA Paper 86-0854, 1986.
- <sup>14</sup>Sobieszcanski-Sobieski, J., "A linear decomposition method for large optimization problems—blueprint for development," Tech. Rep. TM 83248, NASA, 1982.
- <sup>15</sup>Schmit, Jr., L. A. and Mehrotra, M., "Multilevel optimum design of structures with fiber-composite stiffened-panel components," *AIAA Journal*, Vol. 20, No. 1, January 1982, pp. 138-147.
- <sup>16</sup>Barthelemy, J.-F. M., "Development of a Multilevel Optimization Approach to the Design of Modern Engineering Systems," NASA Contractor Report 172184, August 1983, Virginia Polytechnic Institute and State University, Blacksburg, VA.
- <sup>17</sup>Schmit, L. A. and Chang, K. J., "A multilevel method for structural synthesis," *A collection of technical papers* :

AIAA/ASME/ASCE/AHS 25th Structures, Structural Dynamics and Materials Conference, American Institute of Aeronautics and Astronautics, 1984.

<sup>18</sup>Sobieszcanski-Sobieski, J., James, B. B., and Riley, M. F., "Structural Sizing by Generalized, Multilevel Optimization," *AIAA Journal*, Vol. 25, No. 1, January 1987, pp. 139-145.

<sup>19</sup>Frank, P. D. and Shubin, G. R., "A comparison of optimization-based approaches for a model computational aerodynamics design problem," *Journal of Computational Physics*, Vol. 98, No. 1, January 1992, pp. 74-89.

<sup>20</sup>Adelman, H. M., Walsh, J. L., and Pritchard, J. I., "Recent advances in integrated multidisciplinary optimization of rotorcraft, AIAA Paper 92-4777-CP," *Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, September 1992, pp. 710-721.

<sup>21</sup>Sobieszcanski-Sobieski, J., "Two alternative ways for solving the coordination problem in multilevel optimization," *Structural Optimization*, Vol. 6, 1993, pp. 205-215.

<sup>22</sup>Cramer, E., Dennis, Jr., J. E., Frank, P., Lewis, R., and Shubin, G., "Problem formulation for multidisciplinary design optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, November 1994, pp. 754-776.

<sup>23</sup>Balling, R. J. and Sobieszcanski-Sobieski, S., "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.

<sup>24</sup>Balling, R. J. and Sobieszcanski-Sobieski, S., "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.

<sup>25</sup>Braun, R., *Collaborative Optimization: An architecture for large-scale distributed design*, Ph.D. thesis, Stanford University, May 1996, Department of Aeronautics and Astronautics.

<sup>26</sup>Braun, R. D. and Kroo, I. M., "Development and application of the collaborative optimization architecture in a multidisciplinary design environment," *Multidisciplinary Design Optimization: State of the Art*, edited by N. M. Alexandrov and M. Y. Hussaini, SIAM, 1997, pp. 98-116.

<sup>27</sup>Lewis, R. M., "Practical Aspects of Variable Reduction Formulations and Reduced Basis Algorithms in Multidisciplinary Design Optimization," *Multidisciplinary Design Optimization: State-of-the-Art*, edited by N. Alexandrov and M. Y. Hussaini, SIAM, Philadelphia, 1997.

<sup>28</sup>Braun, R. D., Moore, A. A., and Kroo, I. M., "Collaborative approach to launch vehicle design," *Journal of Spacecraft and Rockets*, Vol. 34, July 1997, pp. 478-486.

<sup>29</sup>Sobieski, I. and Kroo, I., "Aircraft design using collaborative optimization," AIAA paper 96-0715, presented at the 34th AIAA Aerospace Sciences Meeting, Reno, Nevada, Jan. 15-18, 1996, 1996.

<sup>30</sup>Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, London, 1981.

<sup>31</sup>Byrd, R. H., Hribar, M. E., and Nocedal, J., "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM J. Optim.*, Vol. 9, No. 4, 1999, pp. 877-900.

<sup>32</sup>Omojokun, E. O., *Trust region algorithms for optimization with nonlinear equality and inequality constraints*, Ph.D. thesis, University of Colorado, Department of Computer Science, Boulder, Colorado, 1989.

<sup>33</sup>Byrd, R. H., Schnabel, R. B., and Shultz, G. A., "A trust region algorithm for nonlinearly constrained optimization," *SIAM Journal on Numerical Analysis*, Vol. 24, No. 5, 1987, pp. 1152-1170.

<sup>34</sup>El-Alem, M., *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*, Ph.D.

thesis, Rice University, Department of Mathematical Sciences, 1988.

<sup>35</sup>El-Alem, M., "A global convergence theory for the Celis-Dennis-Tapia trust-region algorithm for constrained minimization," *SIAM Journal on Numerical Analysis*, Vol. 28, No. 1, February 1991, pp. 266-290.

<sup>36</sup>Dennis, Jr., J. E., El-Alem, M., and Maciel, M. C., "A global convergence theory for general trust-region-based algorithms for equality constrained optimization," *SIAM Journal Optim.*, Vol. 7, No. 1, 1997, pp. 177-207.

<sup>37</sup>Alexandrov, N. M., "Multilevel Methods for MDO," *Multidisciplinary Design Optimization: State of the Art*, edited by N. M. Alexandrov and M. Y. Hussaini, Society for Industrial and Applied Mathematics, Philadelphia, 1997, pp. 79-89.