AIAA-2000-4719

Algorithmic Perspectives on Problem Formulations in MDO

Natalia M. Alexandrov[1] and Robert Michael Lewis[2]

[1] Multidisciplinary Optimization Branch

[2] Institute for Computer Applications in Science and Engineering

NASA Langley Research Center

Hampton, Virginia

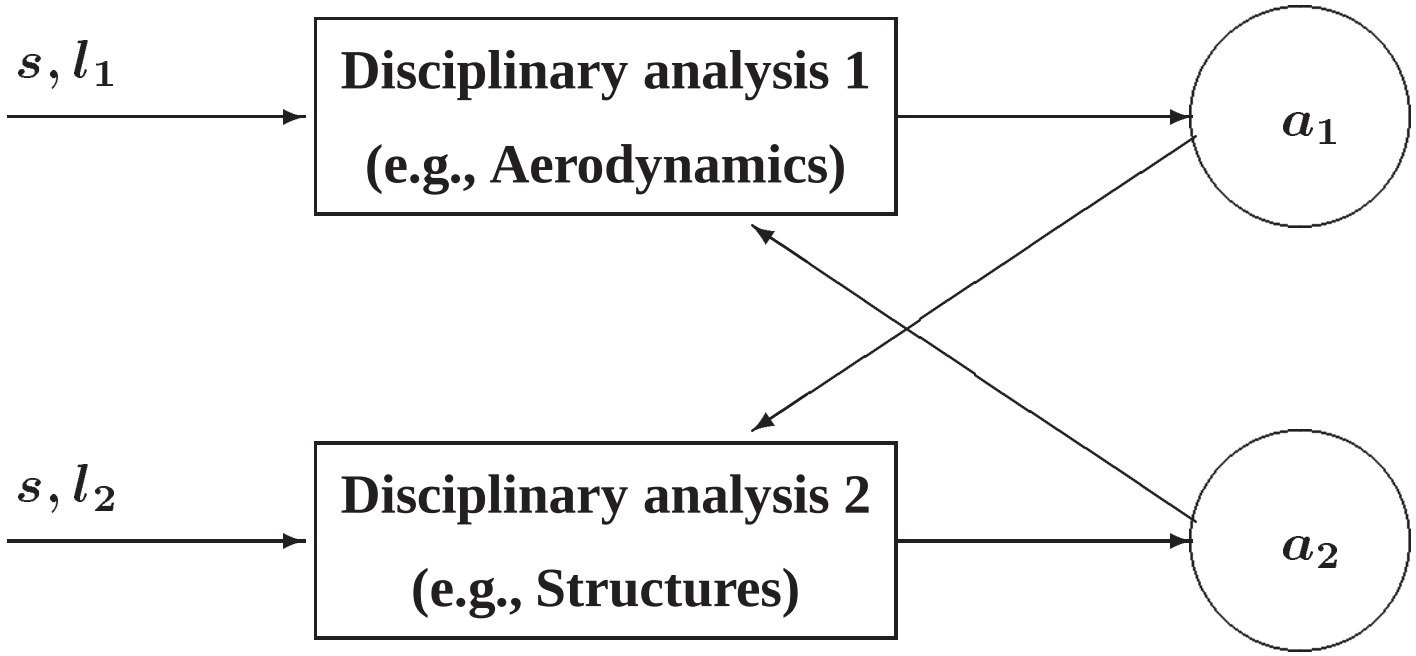
http://fmad-www.larc.nasa.gov/mdob/MDOB

- **Main theme of this and companion paper (AIAA-2000-4718)**
  - **Analytical features of MDO problem formulation strongly influence the practical ability of optimization algorithms to solve the MDO problem reliably and efficiently. For instance, enabling disciplinary autonomy may come at a price our ability to solve the resulting optimization problem reliably**
- **Subject of this paper (an algorithmic approach)**
  - **A modular approach to MDO problem formulation with focus both on problem structure and the need to implement and solve the problem reliably and efficiently**

Outline

- **A two-discipline model problem**
- **Relationship among several problem formulations**
- **Modular implementation**
- **Algorithmic interactions**
- **A comparative summary of formulation properties**
- **Concluding remarks**

The Two-Discipline Model Problem



- **Coupled MDA $\sim$ the physical requirement that a solution satisfy both analyses**
- **Given $x = (s, l_1, l_2)$, we have**

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

Some Algorithmic and Structural Considerations

- **Amenable to solution?**
- **Robust formulation?**
  - **Does the structure respect the canonical problem structure?**
  - **Do answers satisfy necessary conditions?**
  - **Is it sensitive to small changes in parameters?**
- **Efficiency of solution?**
- **Autonomy of implementation / ease of transformation**
  - **Claim: This is most labor-intensive part**
- **Autonomy of execution?**
  - **Wish to follow organizational structure for design**
  - **Wish to optimize wrt local variables only in disciplines**

Relationship among Optimization Problem Formulations

Write MDA as

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

Start with Simultaneous Analysis and Design (SAND) formulation:

$$\begin{aligned} \underset{s,a_1,a_2,l_1,l_2,t_1,t_2}{\textbf{minimize}} \quad & f_{SAND}(s, a_1, a_2) \\ \textbf{subject to} \quad & g_1(s, l_1, a_1) \geq 0 \\ & g_2(s, l_2, a_2) \geq 0 \\ & a_1 = A_1(s, l_1, t_2) \\ & a_2 = A_2(s, l_2, t_1) \\ & t_1 = a_1 \\ & t_2 = a_2 \end{aligned}$$

Relationship among Optimization Problem Formulations (cont.)

- **Eliminate subsets of variables from SAND by *closing* various subsets of constraints $\Longrightarrow$ get other formulations:**
  - **Distributed Analysis Optimization (DAO): Eliminate a_1, a_2 as independent variables by closing the disciplinary analysis constraints at every iteration of optimization**
  - **Fully Integrated Optimization (FIO): In addition, eliminate t_1, t_2 as independent variables by closing $t_1 = a_1$ and $t_2 = a_2$.**
  - **Optimization by Linear Decomposition (OLD): Eliminate l_1, l_2, t_1, t_2 as independent variables via optimization subproblems (MDA remains)**
  - **Collaborative Optimization (CO): Eliminate l_1, l_2 (but not t_1, t_2) via optimization subproblems**

Autonomy/Modularity in Implementation

- **Computational elements needed for optimization (in particular, sensitivities) can be implemented autonomously by disciplines**
- **Can reconfigure the same set of computational elements to implement one discipline or another**
- **All formulations discussed here require roughly the same amount of work to implement**

Example: Sensitivities in DAO vs FIO

Consider DAO:

$$\begin{aligned}
\underset{s,l_1,l_2,t_1,t_2}{\textbf{minimize}} \quad & f_{DAO}(s,t_1,t_2) = f(s, a_1(s,l_1,l_2,t_2), a_2(s,l_1,l_2,t_1)) \\
\textbf{subject to} \quad & g_0(s,t_1,t_2) \geq 0 \\
& g_1(s,l_1,t_1) \geq 0 \\
& g_2(s,l_2,t_2) \geq 0 \\
& t_1 = a_1(s,l_1,l_2,t_2) \\
& t_2 = a_2(s,l_2,l_2,t_1),
\end{aligned}$$

where, given (s, l_1, l_2, t_1, t_2), a_1 and a_2 are found from

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

Example: Sensitivities in DAO vs FIO, cont.

For the objective $f_{DAO}(s, t_1, t_2)$, we need

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

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

$$\frac{\partial g_1}{\partial s}, \frac{\partial g_1}{\partial l_1}, \frac{\partial g_1}{\partial t_1} \quad \text{and} \quad \frac{\partial g_2}{\partial s}, \frac{\partial g_2}{\partial l_2}, \frac{\partial g_2}{\partial t_2}.$$

For the consistency constraints $t_1 - A_1(s, l_1, t_2) = 0$ and

$t_2 - A_2(s, l_2, t_1) = 0$ we need

$$\frac{\partial A_1}{\partial s}, \frac{\partial A_1}{\partial l_1}, \frac{\partial A_1}{\partial t_2} \quad \text{and} \quad \frac{\partial A_2}{\partial s}, \frac{\partial A_2}{\partial l_2}, \frac{\partial A_2}{\partial t_1}.$$

Example: Sensitivities in DAO vs FIO, cont.

Consider FIO:

$$\begin{aligned} \underset{s,l_1,l_2}{\textbf{minimize}} \quad & f(s, a_1(s,l_1,l_2), a_2(s,l_1,l_2)) \\ \textbf{subject to} \quad & g_0(s, l_1, a_1(s,l_1,l_2), a_2(s,l_1,l_2)) \geq 0 \\ & g_1(s, l_1, a_1(s,l_1,l_2)) \geq 0 \\ & g_2(s, l_2, a_2(s,l_1,l_2)) \geq 0, \end{aligned}$$

where a_1 and a_2 are computed in MDA

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

Example: Sensitivities in DAO vs FIO, cont.

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,

$$\begin{aligned}
\frac{\partial f_{FIO}}{\partial s} &= \frac{\partial f}{\partial s} + \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial s} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial s} \\
\frac{\partial f_{FIO}}{\partial l_1} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_1} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_1} \\
\frac{\partial f_{FIO}}{\partial l_2} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_2} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_2}
\end{aligned}$$

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

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

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

and

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

to be solved for the sensitivities of a_1 and a_2 wrt (s, l_1, l_2). (Referred to as the "generalized sensitivity equations" by Sobieski, 1990)

Example: Sensitivities in DAO vs FIO, cont.

- **Observe that the same elements are needed for FIO and DAO sensitivity computations**
- **Can implement constituent elements with disciplinary autonomy if *do not integrate MDA via fixed-point iteration early***
- **The elements are integrated differently in FIO and DAO**
- **Analogous results for CO and OLD**
- **Conclusion: The same computational components are required**

Algorithmic Interactions

- **Saw how, in principle, can re-arrange computational components associated with one formulation and obtain components for another**
- **Re-arrangement may require substantial effort**
- **Now show how for some of the formulations, minor changes in an optimization algorithm may yield an algorithm for solving another formulation**
- **Straightforward to pass among some formulations $\Longrightarrow$ facilitate the use of hybrid approaches: may use one far from solution, another near solution**

Example: DAO vs FIO vs SAND (analysis and coupling constraints only)

Simplified FIO formulation:

$$\underset{x}{\text{minimize}} \quad f_{FIO}(x) \equiv f(x, a_1(x), a_2(x)),$$

where, given x, we solve the MDA

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

Simplified SAND formulation:

$$\begin{aligned} \underset{x, a_1, a_2}{\text{minimize}} \quad & f_{SAND}(x, a_1, a_2) \equiv f(x, a_1, a_2) \\ \text{subject to} \quad & \tilde{A}_1(x, a_1, a_2) = 0 \\ & \tilde{A}_2(x, a_1, a_2) = 0 \end{aligned}$$

Simplified DAO formulation:

$$\begin{aligned} \underset{x, a_1, a_2, t_1, t_2}{\text{minimize}} \quad & f_{DAO}(x, a_1, a_2) \\ \text{subject to} \quad & t_1 - a_1(x, t_1, t_2) = 0 \\ & t_2 - a_2(x, t_1, t_2) = 0 \end{aligned}$$

Example: DAO vs FIO vs SAND, cont.

W_i — basis of the null-space associated with the derivative of the block A_i. Relying on implicit differentiation and the derivations by Lewis, 1997, note the relationship among the sensitivities for the three methods:

- **Suppose, (x, a) is feasible with respect to MDA. Then the (projected) gradients at (x, a) of FIO and SAND 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 MDA. Then**

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

Can use these relationships to implement a reduced-basis optimization algorithm for the three formulations with minimal modifications.

Sketch of a 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.**

- **MDA after step 4 $\Longrightarrow$ a corresponding algorithm for FIO.**
- **Solving the disciplinary equations as in DAO $\Longrightarrow$ an algorithm for DAO.**
- **Passing between algorithms for distinct formulations is a straightforward step.**

Our Currently Favorite Formulation: Expanded DAO

$$
\begin{aligned}
\underset{s,\sigma_0,\sigma_1,\sigma_2,l_1,l_2,t_1,t_2}{\textbf{minimize}} \quad & f_{DAO}(s,t_1,t_2) \\
\textbf{subject to} \quad & g_0(\sigma_0,t_1,t_2) \geq 0 \\
& g_1(\sigma_1,l_1,t_1) \geq 0 \\
& g_2(\sigma_2,l_2,t_2) \geq 0 \\
& t_1 = a_1(\sigma_1,l_1,t_2) \\
& t_2 = a_2(\sigma_2,l_2,t_1) \\
& \sigma_0 = s \\
& \sigma_1 = s \\
& \sigma_2 = s
\end{aligned}
$$

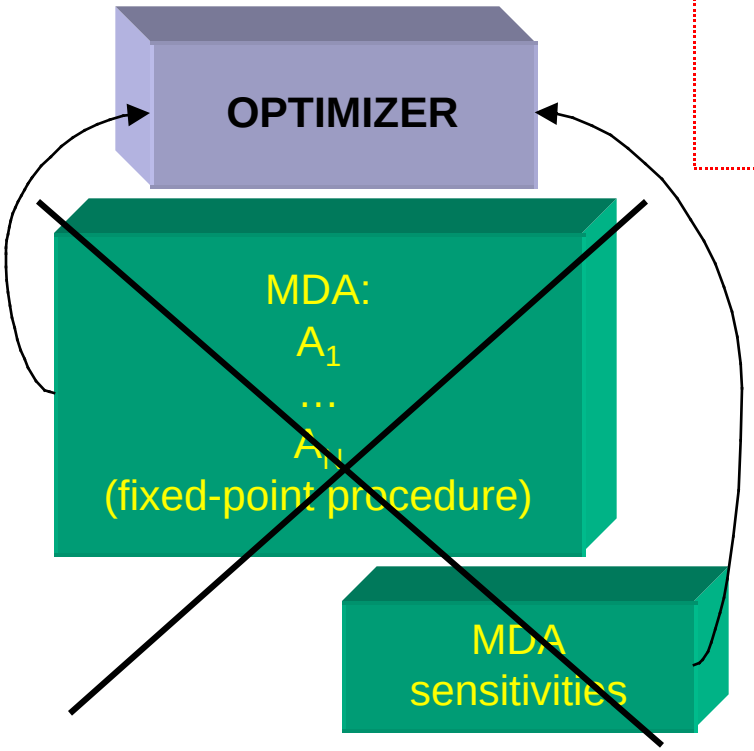
- **Expand variable space to relax the requirement that the disciplinary design constraints be satisfied with the system-level values of s**
- **Implementation autonomy, no MDA**
- **Single-level optimization problem - readily soluble**

MDO Problem Integration / Implementation

Problem: design for objective f with

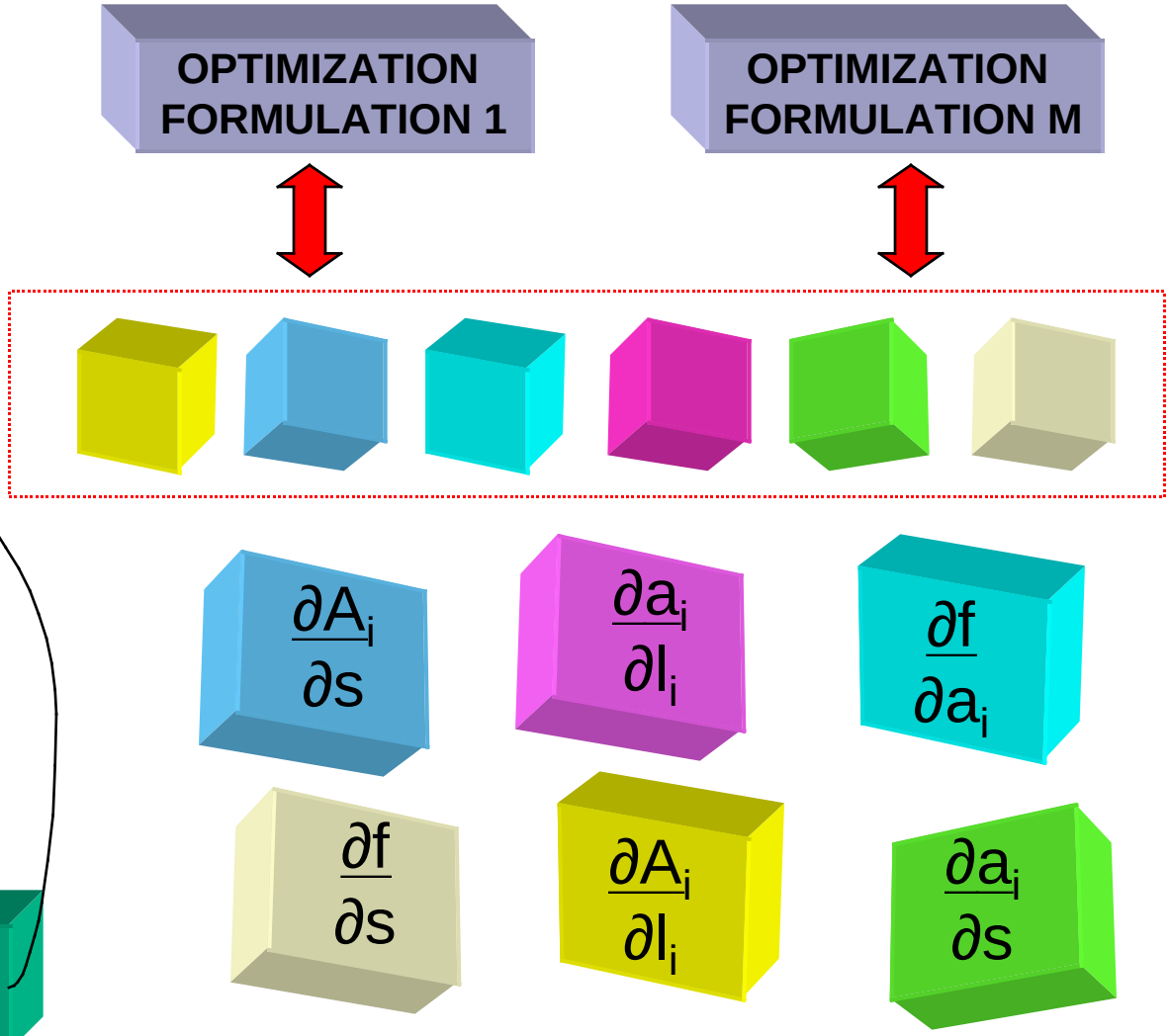
i = 1, …, N

NOW



Laborious, expensive, one-time integration, difficult to transform/ expand

FUTURE



Expend the effort at the outset to implement analysis and sensitivity modules; easy to transform and expand: an opportunity for a general framework

Appendix: Comparative Summary of Formulations

- **FIO: Single-level optimization, arbitrary coupling, some autonomy of implementation, MDA required**

- **SAND: Single-level optimization, arbitrary coupling, some autonomy of implementation, MDA not done, large optimization problem**

- **DAO: Single-level optimization, not for broadly coupled problems, autonomy of implementation, some autonomy of execution**

- **CO: Bilevel optimization, autonomy of implementation and autonomy of execution (distributed MDA), local variables handled in subproblems, no MDA, not for broadly coupled problems, not robust, can be difficult to solve**

- **OLD: Bilevel optimization, MDA required, autonomy of implementation and some autonomy of execution, not robust, can be difficult to solve**