Multidisciplinary System
Design Optimization (MSDO)

Decomposition and Coupling
Lecture 4
17 February 2004

Olivier de Weck

© Massachusetts Institute of Technology - Prof. de Weck and Prof. Willcox
Today’s Topics

Last time discussed standard approach: Sequential modular analysis (Lecture 3).

• Modules are executed sequentially with or without feedback loops.

Other Approaches:
  – MDO frameworks
  – Distributed analysis
  – Distributed design
Fundamentally different approaches in MDO

Distributed Analysis
- disciplinary models provide analysis
- all optimization done at system level

Distributed Design
- provide disciplinary models with design tasks
- optimization at subsystem and system levels

non-hierarchical decomposition

hierarchical decomposition

CSSO
CO
BLISS

© Massachusetts Institute of Technology - Prof. de Weck and Prof. Willcox
Given
\[ x \in \mathbb{R}^n \]
\[ J : \mathbb{R}^n \to \mathbb{R} \]
\[ g : \mathbb{R}^n \to \mathbb{R}^m \]

Solve the problem
\[ \min J(x) \]
\[ \text{s.t. } g(x) \geq 0 \]

That is, find \( x^* \) s.t. \( J(x^*) \leq f(x), \forall x \in \text{dom}(J) \cap \text{dom}(g) \)
Distributed Analysis

• Disciplinary models provide analysis
• Optimization is controlled by some overseeing code or database
  
  e.g. GenIE database system (Stanford)
  iSight (Optimizer)
During the optimization, the overseeing code keeps track of the values of the design variables and objective. The values of the design variables are changed according to the optimization algorithm. Disciplinary models are asked to evaluate constraints/objective.
Distributed Design

System level optimizer

SS1 optimizer
SS1 analyzer

SS2 optimizer
SS2 analyzer

SSN optimizer
SSN analyzer

Subsystem black box (BB)

command/result
command/result
command/result
Advantages of Decoupling

Computation of $g(x)$ can be very time consuming, want to divide the work and compute in parallel. For example, if $x = (x_1, x_2)$, where $x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}$ and $g(x) = (g_1(x_1), g_2(x_2))$.

Then $g_1$ and $g_2$ can be computed in parallel. Graphically,
The decoupled constraints assumption is not general. Subsystems can be coupled and loops can arise. For example,

Computation of $w_1$ and $w_2$ requires an iterative method.
• An example where such a loop happens is as follows:

\[
\begin{align*}
\min J(x_1, x_2) \\
& w_1 = g_1(x_1, g_2(x_2, w_1)) \geq 0 \\
\text{s.t.} & \quad w_2 = g_2(x_2, g_1(x_1, w_2)) \geq 0 \\
\end{align*}
\]

where \( x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}, g_i : x_i \times i \mapsto w_i, i = 1, 2 \)

• \( w_1 \) and \( w_2 \) satisfy coupled relations at each optimization iteration. At each constraint evaluation, nonlinear equations must be solved (e.g. by Newton’s method) in order to obtain \( w_1 \) and \( w_2 \), which can be time consuming.

Want a way to return to the situation of decoupled constraints.
Surrogate Variables (“Tearing”)

Information loop can be broken by introducing surrogate variables.

\[
\min J(x_1, x_2) \\
\text{s.t.} \quad w_1 = g_1(x_1, g_2(x_2, w_1)) \geq 0 \\
\text{s.t.} \quad w_2 = g_2(x_2, g_1(x_1, w_2)) \geq 0
\]

\[
\min J(x_1, x_2) \\
\text{s.t.} \quad g_1(x_1, u_1) \geq 0 \\
\text{s.t.} \quad g_2(x_2, u_2) \geq 0 \\
\text{s.t.} \quad u_2 - g_1(x_1, u_1) = 0 \\
\text{s.t.} \quad u_1 - g_2(x_2, u_2) = 0
\]

• \(u_1\) and \(u_2\) are decision variables acting as the inputs to \(g_1(\text{SS1})\) and \(g_2\) (SS2). Introducing surrogate variables breaks information loop but increases the number of decision variables.
min $J_1 + J_2$

s.t. $w_1 \geq 0$

$w_2 \geq 0$

where $J_1 = x_1^2 + x_2^2$

$$J_2 = (x_3 - 3)^2 + (x_4 - 4)^2$$

$$w_1 = x_1^3 - x_2^3 + 2w_2$$

$$w_2 = x_3^3 - x_4^3 + 2w_1$$

min $x_1^2 + x_2^2 + (x_3 - 3)^2 + (x_4 - 4)^2$

s.t. $w_1 = g_1(x_1, x_2, x_3, x_4) \geq 0$

$w_2 = g_2(x_1, x_2, x_3, x_4) \geq 0$

Solution:

$x = (0, 0, 4, 3, 12 \frac{1}{3}, 24 \frac{2}{3})$

MATLAB 5.3

coupled: 356,423 FLOPS 4.844s

uncoupled: 281,379 FLOPS 0.453s
Distributed Design Methods

- Disciplinary models are provided with design tasks
- Optimization is performed at a subsystem level in addition to the system level

**Concurrent Subspace Optimization (CSSO)**

- divide the design problem into several discipline-related subspaces
- each subspace shares responsibility for satisfying constraints while trying to reduce a global objective

**Collaborative Optimization (CO)**

- disciplinary teams satisfy local constraints while trying to match target values specified by a system coordinator
- preserves disciplinary-level design freedom
Collaborative Optimization

Coupled

Uncoupled

OPTIMIZER

TARGET STATE
Two levels of optimization:

- A system-level optimizer provides a set of targets.
  - These targets are chosen to optimize the system-level objective function

- A subsystem optimizer finds a design that minimizes the difference between current states and the targets.
  - Subject to local constraints
Collaborative Optimization

\[
\begin{align*}
\text{min } J_{\text{sys}} \\
\text{wrt: } x_0 &= \{\text{target variables}\} \\
\text{s.t. } J_k &= 0 \quad \forall \text{ subproblems } k
\end{align*}
\]

\[
\begin{align*}
\text{min } J_1 &= \left\{ \text{target variables} - \text{local variables} \right\}^2 \\
\text{x} &= \{\text{local variables}\} \\
\text{s.t. } \{\text{local constraints}\}
\end{align*}
\]

\[
\begin{align*}
\text{min } J_k &= \left\{ \text{target variables} - \text{local variables} \right\}^2 \\
\text{x} &= \{\text{local variables}\} \\
\text{s.t. } \{\text{local constraints}\}
\end{align*}
\]
The subsystem optimizer modifies local variables to achieve the best design for which the set of local variables and computed results most nearly matches the system targets.

The local constraints must also be satisfied.

\[
\min J_1 = \left\{ \frac{\text{target variables} - \text{local variables}}{2} \right\}^2
\]

\[
x = \{\text{local variables}\}
\]

\[
\text{s.t. } \{\text{local constraints}\}
\]
min $J_{sys}$

wrt: $x_0 = \{\text{target variables}\}$

s.t. $J_k = 0 \quad \forall \text{subproblems}_k$

- System-level optimizer changes target variables to improve objective and reduce differences $J_k$
  - $J_k=0$ are called compatibility constraints
  - compatibility constraints are driven to zero, but may be violated during the optimization
  - CO may therefore discover parts of the design space that cannot be reached by sequential optimization
Consider a simple aircraft design problem: maximize range for a given take-off weight by choosing wing area, aspect ratio, twist angle, L/D, and wing weight.

\[ \text{wing area, } S \]
\[ \text{aspect ratio, } AR \]

\[ \text{aero} \]

\[ \text{twist angle, } \theta \]

\[ \text{struct} \]

\[ \text{wing weight, } W \]

\[ \text{perf} \]

\[ \text{range, } R \]

modified from Kroo et al. AIAA 94-4325
CO Example: Aircraft Design

\[
\begin{align*}
\text{max } & R_0 \\
x_0 & = [R_0 \ S_0 \ AR_0 \ \theta_0 \ L/D_0 \ W_0]^T \\
\text{s.t. } & J_1 = 0, \ J_2 = 0, \ J_3 = 0
\end{align*}
\]

\[
\begin{align*}
\text{min } & J_1 \\
J_1 & = (AR-AR_0)^2 + (\theta-\theta_0)^2 + (L/D-L/D_0)^2 + (S-S_0)^2 \\
x & = [AR \ \theta]^T
\end{align*}
\]

\[
\begin{align*}
\text{min } & J_2 \\
J_2 & = (AR-AR_0)^2 + (\theta-\theta_0)^2 + (S-S_0)^2 + (W-W_0)^2 \\
x & = [S \ AR]^T
\end{align*}
\]

\[
\begin{align*}
\text{min } & J_3 \\
J_3 & = (R-R_0)^2 + (L/D-L/D_0)^2 + (W-W_0)^2 \\
x & = [L/D \ W]^T
\end{align*}
\]

© Massachusetts Institute of Technology - Prof. de Weck and Prof. Willcox
Collaborative Optimization

\[ \min J_{sys} \]

wrt: \( x_0 = \{ \text{target variables} \} \)

s.t. \( J_k = 0 \quad \forall \text{ subproblems} \)

\[ \{x_0\} \quad \text{performance analysis} \]

\( J_1 \)

\[ \min J_1 = \left\{ \text{target variables} - \text{local variables} \right\}^2 + \left\{ \text{coupling variables} - \text{local variables} \right\}^2 \]

\[ x = \{ \text{local variables} \} \]

s.t. \( \{ \text{local constraints} \} \)

\( \{x\} \quad \text{computed results} \)

analysis for subsystem 1

\( J_k \)

\[ \min J_k = \left\{ \text{target variables} - \text{local variables} \right\}^2 + \left\{ \text{coupling variables} - \text{local variables} \right\}^2 \]

\[ x = \{ \text{local variables} \} \]

s.t. \( \{ \text{local constraints} \} \)

\( \{x\} \quad \text{computed results} \)

analysis for subsystem \( k \)
$x_0 = \text{system-level target variable values}$

$x = \text{subsystem local variables}$

$y_{ij} = \text{coupling functions}$

- $y_{ij} = \text{outputs of subsystem } j \text{ which are needed as inputs to subsystem } i.$
- Coupling equations must also be satisfied, so coupling variables are included in subsystem objective.
- Used to reduce the number of system-level parameters.
\[ Q_i = \left\{ X_{sh}, Y_{BBi}^*, w_{BBi} \right\} \]
A black box has the following properties:

1. BB has its own local variables (Xloc) and has the exclusive right to determine Xloc. Xloc is a subset of decision variables that can appear explicitly only in the associated BB.
2. BB must satisfy its constraints at each system level iteration.
3. BB operates independently of other BB’s. Neither its inputs nor its outputs are directly communicated between other BB’s. Also, BB assumes no knowledge (e.g. Xloc) of other BB’s. Instead, BB connection is done implicitly via the system optimizer, by the use of Y*.
4. Computation methods within a BB are not restricted by BLISS. (It can be simulation or just an intelligent guess.)
• BLISS is a bi-level optimization algorithm. The subsystem optimization formulation is as follows:

Given: \( Q = \{ X_{sh}, Y^*, w \} \)

variables: \( U = \{ X_{loc}, Y^* \} \)

min : \( f(U) = \sum_i w_i Y_i^* \)

s.t. \( g(U) \leq 0, \) for each BB

\( h(U) = 0, \) for each BB

\( U_l \leq U \leq U_u \)

output: \( Y^* \)

keep: \( X_{loc} \)

\( X_{sh} \) : share decision variable

\( Y^* \) : input to BB from other BB (surrogate var)

\( w \) : weight used in BB optimization

\( X_{loc} \) : local decision variable

\( Y^* \) : output of BB (to system and/or other BB)

\( g(\cdot) \) : BB inequality constraints

\( h(\cdot) \) : BB equality constraints

\( U_{lower} \) : lower bound on local variables

\( U_{upper} \) : upper bound on local variables
Insert – Slides by Dr. Sobieski
Wing drag and weight both influence the flight range $R$.

$R$ is the system objective

- Structure influences $R$ by
  - directly by weight
  - indirectly by stiffness that affect displacements that affect drag

$$R = \frac{k}{\text{Drag}} \log \left[ \frac{W_o + W_s + W_f}{W_o + W_s} \right]$$

- Dilemma: What to optimize the structure for? Lightness? Displacements = $1/\text{Stiffness}$? An optimal mix of the two?

Courtesy of Jaroslaw Sobieski. Used with permission.
Trade-off between opposing objectives of lightness and stiffness

- What to optimize for?
  - Answer: minimum of $f = w_1 \text{Weight} + w_2 \text{Displacement}$
  - vary $w_1, w_2$ to generate a population of wings of diverse Weight/Displacement ratios
  - Let system choose $w_1, w_2$.

Courtesy of Jaroslaw Sobieski. Used with permission.
Approximations

• a.k.a. Surrogate Models

• Why Approximations:
  - OK for small problems
  - Now-standard practice for large problems to reduce and control cost

$$\text{cents}$$

Analyzer → Human judgment ← Approximate Model

Analyzer → Optimizer ← Human judgment

Courtesy of Jaroslaw Sobieski. Used with permission.
Design of Experiments (DOE) & Response Surfaces (RS)

• RS provides a “domain guidance”, rather than local guidance, to system optimizer

DOE

• Placing design points in design space in a pattern

Example: Star pattern (shown incomplete)

RS

\[ F(X) = a + \{b\}'X + \{X\}'[c]X \]

• quadratic polynomial
• hundreds of variables

Courtesy of Jaroslaw Sobieski. Used with permission.
BLISS 2000: MDO Massive Computational Problem Solved by RS (or alternative approximations)

System optimization

Instantaneous response

Precompute off-line in parallel

Optimization of subsystem or discipline

MC cloud

Analysis of subsystem or discipline

Optimization of subsystem or discipline

- Radical conceptual simplification at the price of a lot more computing. Concurrent processing exploited.

Courtesy of Jaroslaw Sobieski. Used with permission.
Coupled System Sensitivity

- Consider a multidisciplinary system with two subsystems A and B (e.g. Aero. & Struct.)
  - system equations can be written in symbolic form as
    \[ A[(X_A, Y_B), Y_A] = 0 \]
    \[ B[(X_B, Y_A), Y_B] = 0 \]
  - rewrite these as follows
    \[ Y_A = Y_A(X_A, Y_B) \]
    \[ Y_B = Y_B(X_B, Y_A) \]

these governing equations define as implicit functions. **Implicit Function Theorem** applies.

Courtesy of Jaroslaw Sobieski. Used with permission.
Coupled System Sensitivity - Equations

- These equations can be represented in matrix notation as

\[
\begin{bmatrix}
I & -\frac{\partial Y_A}{\partial Y_B} \\
-\frac{\partial Y_B}{\partial Y_A} & I
\end{bmatrix}
\begin{bmatrix}
\frac{dY_A}{dX_A} \\
\frac{dY_B}{dX_A}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial Y_A}{\partial X_A} \\
0
\end{bmatrix}
\]

- Total derivatives can be computed if partial sensitivities computed in each subsystem are known
  Linear, algebraical equations with multiple RHS

Courtesy of Jaroslaw Sobieski. Used with permission.
Flowchart of the System Optimization Process

System Analysis

System Sensitivity Analysis

Sensitivity solution

Optimizer

Approximate Analysis

Start

Stop

Courtesy of Jaroslaw Sobieski. Used with permission.
System Internal Couplings Quantified

- Strength: relatively large $\partial YO/ \partial YI$
- Breadth:
  - $\{YO\}$ and $\{YI\}$ are long
  - $[\partial YO/ \partial YI]$ large and full

Courtesy of Jaroslaw Sobieski. Used with permission.
Supersonic Business Jet Test Case

- Structures (ELAPS)
- Aerodynamics (lift, drag, trim supersonic wave drag by A - Wave)
- Propulsion (look-up tables)
- Performance (Breguet equation for Range)

Examples: Xsh - wing aspect ratio, Engine scale factor
Xloc - wing cover thickness, throttle setting
Y - aerodynamic loads, wing deformation.

Some stats:
- Xlocal: struct. 18 aero 3 propuls. 1
- X shared: 9
- Y coupl.: 9

Courtesy of Jaroslaw Sobieski. Used with permission.
System of Modules (Black Boxes) for Supersonic Business Jet Test Case

- Data Dependence Graph
- RS - quadratic polynomials, adjusted for error control

Courtesy of Jaroslaw Sobieski. Used with permission.
• Histogram of RS predictions and actual analysis for Range

Courtesy of Jaroslaw Sobieski. Used with permission.

Updated Journal Article (handout):


References (II)


