6^{th} **World Congress on Structural and Multidisciplinary Optimization**
Rio de Janeiro, 30 May–03 June, 2005 Brazil

On the Use of Analytical Target Cascading and Collaborative Optimization for Complex System Design

James Allison, Michael Kokkolaras, Marc Zawislak, Panos Y. Papalambros

Department of Mechanical Engineering, The University of Michigan
2350 Hayward, Ann Arbor, Michigan 48109-2125
{optimize,mk,mzawisla,pyp}@umich.edu

1 Abstract

The methods of analytical target cascading (ATC) and collaborative optimization (CO) are studied with respect to their intended use and applicability. ATC was initially developed as a product development tool, while CO evolved from efforts to coordinate multidisciplinary analyses using a multidisciplinary design optimization (MDO) formulation. ATC and CO are used typically to solve object-based and discipline-based decomposed system optimization problems, respectively. Their mathematical formulations appear to be similar although they were developed with different motivations. The article defines and compares terminologies for each approach and shows that each has unique applicability and solution process. Two new analytical example problems are employed to illustrate the distinctions between ATC and CO. The first example elucidates how each method can be used to solve the same design problem. The second example uses a new optimization formulation, nested ATC-MDO, to illustrate their complementary nature. In this formulation a system is partitioned by object, and objects are partitioned by discipline. ATC coordinates the design of system elements and an MDO method, such as CO, coordinates the multidisciplinary design of an element. The formulation maps well to organizational matrix structures. The overall study demonstrates the benefits of using complementary solution strategies in solving complex system optimization problems.

2 Keywords: Analytical Target Cascading, Collaborative Optimization, Complex System Design, Matrix Organizations

3 Introduction

Design of modern engineering products can be a complex task, and numerous methodologies have been developed to support this endeavor. One approach is to partition the original design problem into smaller and easier to solve subproblems, and then coordinate these problems toward a consistent and optimal system solution. A key motivation for decomposition methods is the nature of design organizations. No single group or computing facility is typically capable of executing all design activities associated with a complex product, requiring distributed analysis and design. Existing organizational structures provide a natural means for partitioning a design problem.

One category of decomposition methods is multidisciplinary design optimization (MDO), a field of engineering based on the needs of multidisciplinary analysis (MDA) and on exploiting the synergy that exists between constituent components of a system [1]. Single-level MDO methods, such as the individual disciplinary feasible and all-at-once approaches [2, 3], utilize a single optimizer to make all design decisions. Although typically efficient, these centralized approaches have heavy communication requirements, and may not map well to many existing design organizations and analysis tools. In response, multi-level MDO methods have been developed that distribute decision making throughout the system (providing design groups with some autonomy and ability to utilize existing design tools), and reduce communication requirements.

This article investigates the distinctions between Collaborative Optimization (CO), a bi-level MDO method, and Analytical Target Cascading (ATC), a formal product development tool. CO was first presented by Braun [4] in 1996, and ATC was formally presented by Michelena *et al.* [5] in 1999. Both methods have been actively researched through the present, producing numerical refinements and extensions to practical applicability. Recent questions have arisen regarding the distinctions between CO and ATC, owing to similar mathematical formulations and overlapping terminology. This article examines the formulation and solution process of each method, and clarifies terminology. Two illustrative examples provide a venue for clarifying and reinforcing arguments made. The first example demonstrates the solution of a structural design problem first with CO, and then with ATC, utilizing the same partitioning structure in each case. This provides a clear comparison of solution process, communication patterns, and formulation. The second example introduces a new formulation for solving complex design problems: nested ATC-MDO. In addition to supplying a means for observing ATC and CO within the same formulation, nested ATC-MDO exploits the complementary strengths of ATC and MDO. This approach applies to design problems that can first be partitioned by object, and then by discipline, and maps well to design organizations with a matrix structure [6]. This article does not attempt to make performance evaluations or comparisons, but rather focuses on process and paradigm distinctions. It is suggested that flexibility in choosing the methodology or combination of methodologies best suited for a task is an important approach to solving complex system optimization problems.

4 Review of Collaborative Optimization

This section briefly reviews the background, formulation, and recent refinements of collaborative optimization (CO), an MDO strategy that facilitates the integration of analyses required for the design of a complex product. Factors motivating the development of CO include reduced communication requirements, provision for distributed design authority, and flexibility to accommodate analysis changes. CO was developed under the MDO paradigm, where systems are typically partitioned along discipline boundaries, which are often dictated by the design organization structure or available analysis tools. Separate disciplinary analyses, or subspaces, are coupled by interactions, that increase design problem difficulty. MDO strategies aim at exploiting interactions synergistically to produce superior results.

An implicit emphasis of MDO methods is analysis and input/output relations. An important question pertinent to structuring an MDO design problem is how to integrate the various analyses best. The CO architecture allows legacy analysis tools to be used without special modifications, and accommodates specialized optimization techniques for particular disciplines, such as optimal control or structural optimization.

Before decomposition, quantities of interest are the design variables $\mathbf{x}$, the objective function $f(\mathbf{x})$, and the design constraint functions $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$. After decomposition, other values of interest emerge. The vector of coupling variables $\mathbf{y}_{ij}$ is the set of values computed by subspace j required as inputs to subspace i. The collection of all coupling variables $\mathbf{y}$ has no common components with $\mathbf{x}$. In addition, $\mathbf{x}$ can be partitioned into local variables $\mathbf{x}_{\ell i}$ that are pertinent only to subspace i, and shared variables $\mathbf{x}_{si}$ that are inputs to subspace i and at least one other subspace. The vector $\mathbf{x}_i$ contains local and shared variables required for subspace i.

A result of CO development under the MDO paradigm is suitability for design problems with a particular structure. A collection of analyses does not usually have a natural hierarchical ordering—all subspaces are of similar importance. There is no system level analysis that integrates lower level analyses. In other words, MDO problem structures have shared variables, but normally lack shared functions. Using the terminology of Chen [7], the associated functional dependency table is column based. This structure and lack of system-level analysis reinforces emphasis on analysis and analysis integration. An implication of this structure is the sufficiency of a bi-level problem structure for CO. The top level performs the coordination, and all subspaces are located at the lower level. Since analyses are of equal importance, it is not necessary to extend the CO hierarchy to multiple levels. In cases where an analysis hierarchy does exist or system analysis is present, compatibility constraints, which will be explained shortly, can bring all subspaces onto the same level.

Figure 1 illustrates the architecture of CO and the associated hierarchic communication channels. A system optimizer coordinates the activities of all subspace design problems, guiding the system toward optimality and consistency. The system optimizer seeks system optimality by minimizing the system objective function f_s, and requires system consistency via auxiliary constraints ($\mathbf{f}^* = [f_1^*, f_2^* \dots f_N^*]^{\mathrm{T}} = \mathbf{0}$), which require consistency among all coupling and shared variables. The auxiliary constraints decouple the subspaces, such that communication is only required between the subspaces and the system optimizer. This also allows parallel execution of analyses and transforms the problem structure from non-hierarchic to hierarchic. System optimization is performed with respect to the system targets $\hat{\mathbf{z}}$, which are the vehicle for negotiation between subspaces. Subspace optimizer i receives the targets $\hat{\mathbf{z}}_i$ pertinent to subspace i, and seeks to minimize the discrepancy between these targets and the corresponding computed quantities $\mathbf{z}_i$, subject to local design constraints $\mathbf{g}_i(\mathbf{x}_i)$ and $\mathbf{h}_i(\mathbf{x}_i)$. The discrepancy between the system targets and their corresponding subspace quantities is captured by the i^{th} subspace objective function, f_i. The system objective function $f_s(\hat{\mathbf{z}})$ is a scalar function of the system targets, and is computed directly by one of the subspaces.

A target in the vector $\hat{\mathbf{z}}_i$ exists for every shared variable $\mathbf{x}_{si}$ used in subspace i and for every input $\mathbf{y}_{ij}$ and output $\mathbf{y}_{ji}$ coupling variable, i.e., $\mathbf{z}_i = [\mathbf{x}_{si}^{\mathrm{T}}, \mathbf{y}_{ij}^{\mathrm{T}}, \mathbf{y}_{ji}^{\mathrm{T}}]^{\mathrm{T}}$. No targets exist for local design variables $\mathbf{x}_i$, allowing subspaces direct control over them and providing a degree of subspace autonomy.

The subspace objectives f_i measure discrepancy between subspace targets and the corresponding responses: $f_i(\mathbf{x}_{\ell i}, \mathbf{y}_{ij}) = \|\mathbf{z}_i - \hat{\mathbf{z}}_i\|_2^2$. The local targets $\hat{\mathbf{z}}_i$ are fixed parameters in the subspace optimizer, which seeks to match these targets by varying the local design variables and input coupling variables. The output coupling variables $\mathbf{y}_{ij}$ are computed based on these decision variables, and are incorporated into f_i. At every system level iteration, the optimal value of the subspace objective function f_i^* is passed to the system optimizer and used as a system-level auxiliary constraint. Thus, the CO process consists of nested optimization. The original CO formulation is:

System Level Formulation

$$\begin{aligned} \min_{\hat{\mathbf{z}}} \quad & f_s(\hat{\mathbf{z}}) \\ \text{subject to} \quad & \mathbf{f}^*(\hat{\mathbf{z}}) = \mathbf{0} \end{aligned} \tag{1}$$

Subspace Formulation

$$\begin{aligned} \min_{\mathbf{x}_{\ell i}, \mathbf{y}_{ij}} \quad & f_i(\mathbf{x}_{\ell i}, \mathbf{y}_{ij}) = \|\mathbf{z}_i - \hat{\mathbf{z}}_i\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_i(\mathbf{x}_{\ell i}, \mathbf{y}_{ij}) \leq \mathbf{0} \\ & \mathbf{h}_i(\mathbf{x}_{\ell i}, \mathbf{y}_{ij}) = \mathbf{0} \end{aligned} \tag{2}$$

Some researchers have identified difficulties with the original CO formulation, particularly when gradient-based algorithms are employed [8, 9]. At the solution, the consistency constraint gradient vanishes, Lagrange multipliers approach zero, and constraint qualification is not met. A practical solution is to loosen algorithm tolerances, as was done in the first example of this article. However, this limits numerical accuracy. Another issue is that the subspace objective function is generally not differentiable. DeMiguel and Murray proposed a modified CO formulation (MCO) in 2000 [9]. The compatibility constraints are relaxed and moved to the system objective as an exact penalty function, and a smoothing algorithm is used to ameliorate other remaining difficulties.

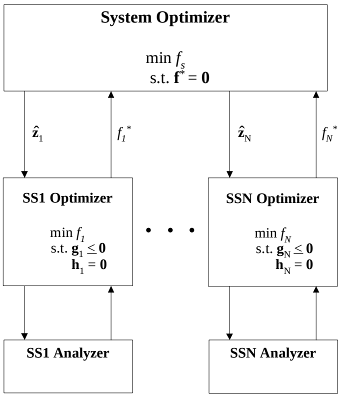

Figure 1: Collaborative Optimization architecture

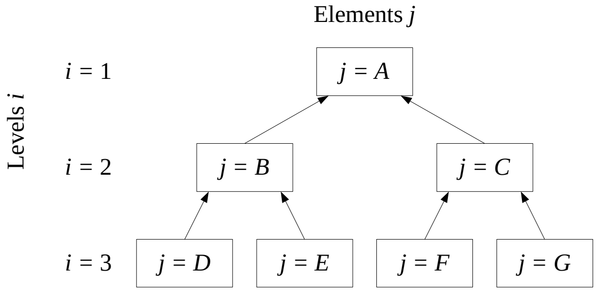

Figure 2: ATC hierarchy notation conventions

5 Review of Analytical Target Cascading

Analytical Target Cascading (ATC) is the result of efforts to formalize activities early in the product development process. It was originally intended as a tool to cascade system-level product targets through a hierarchy of design groups. The ATC architecture was originally based on hierarchical analysis structures with unidirectional functional dependencies. This maps well to many design organizations and can help avoid the need to restructure communication channels for the solution process. Recent developments allow application of ATC to more general problems that possess feedback coupling, such as the first example problem of this article.

The ATC paradigm is based on hierarchical organizational and analysis structures, which are typically partitioned by object. A typical ATC approach is to take a high-level system analysis and use more detailed subsystem analyses at the lower levels. This is a different approach than CO, which focuses on discipline-based analysis integration. An example hierarchy with standardized notation is shown in Figure 2. ATC typically involves system-integration analysis, and so it is a natural fit to a very general class of problem structures with shared functions, i.e., problems with a hybrid functional dependency table [7]. This also results in reduced emphasis on analysis integration. Additionally, systems modeled under the ATC paradigm can 'absorb' some coupling variables into a system analysis that in an MDO approach would be explicitly handled by the system optimizer.

The discussion here focuses on the target propagation process, the nexus of ATC. When used for product development, ATC propagates product targets using a model-based solution process, and targets are provided to design groups to work towards. If the design groups cannot meet the targets or if there are consistency problems, the target propagation is revisited with new system targets and possibly more accurate analysis models.

The original ATC formulation described in [5, 10] includes inequality compatibility constraints that enforce consistency within some tolerance ε. These tolerances are decision variables, and are driven to zero by placing them in the objective function of the parent element formulation. Monotonicity analysis [11] reveals the compatibility constraints to be always active, allowing for substitution for ε in the objective function. This penalty formulation was introduced by Michalek and Papalambros [12, 13], reducing the dimension of the optimization problem and facilitating the use of efficient weighting update methods [12, 14]. In addition, the penalty formulation is an intuitive basis to explain ATC. Shared values (shared variables or coupling variables) are identified. Copies of these values are made in the appropriate elements, and penalties are assigned in the element objective functions for inconsistencies among shared values.

The generalized penalty formulation for the optimization problem P_{ij} associated with element j at level i is given in Equation (3). This formulation has been extended from what was presented in [12, 13] to accommodate systems with feedback coupling, i.e., systems with responses generated by parent element analyses that are required as inputs to the analysis of child elements. This extends ATC's applicability beyond product development to MDO type problems, and is helpful for this article's focus. The responses $\mathbf{R}_{ij}^{i}$ are outputs of the analysis $\mathbf{r}_{ij}$ of element j at level i required by the parent element as analysis inputs. $\mathbf{R}_{ik}^{i}$ are also outputs of the analysis $\mathbf{r}_{ij}$, but are required by child element k. $\mathbf{R}_{ij}^{i-1}$ are targets set by the parent element p at level $i-1$ for $\mathbf{R}_{ij}^{i}$. $\mathbf{R}_{(i-1)p}^{i}$ are targets set by element j at level i for $\mathbf{R}_{(i-1)p}^{i-1}$, which are responses generated by the parent element p at level $i-1$ that are inputs to the analysis of element j. The linking variables $\mathbf{y}_{ij}^{i}$ (equivalent to shared variables in MDO terminology) are the design variables required at element j that are shared with other elements, as determined by element j at level i. The vector $\mathbf{y}_{ip}^{i-1}$ is comprised of linking variable targets set by the parent element p at level $i-1$ for child elements at level i. The binary valued selection matrix $\mathbf{S}_j$ is multiplied by the aggregate vector $\mathbf{y}_{ip}^{i-1}$ to choose the targets that correspond to $\mathbf{y}_{ij}^{i}$. $\mathbf{S}_j$ is also used to form the vector of corresponding linking variable penalty weights from $\mathbf{w}_{ip}^{y}$. The $\circ$ operator denotes term by term vector multiplication, such that each term in a weighting vector is multiplied by the term in the deviation vector with the same index. This allows every shared value to have assigned to it a weight expressing the relative importance of consistency for that shared value. $\mathbf{S}_p$ and $\mathbf{S}_k$ are the matrices that select what outputs of analysis $\mathbf{r}_{ij}$ are to be passed to the parent element and child element k, respectively. Terms 4 and 6 of the objective enforce penalties for deviation between targets set by element j for all elements k that belong to the set of all children $\mathcal{C}_{ij}$ of element j and the corresponding responses and linking variables determined at level $i+1$. Term 5 of the objective enforces consistency between targets set by child elements and the corresponding responses from element j.

Optimization is performed with respect to all inputs to the analysis for element j ($\bar{\mathbf{x}}_{ij}$) and the targets set for linking variables pertaining to child elements ($\mathbf{y}_{(i+1)j}^{i}$). Analysis inputs include local design variables $\mathbf{x}_{ij}$, linking variables $\mathbf{y}_{ij}$, responses from below $\mathbf{R}_{(i+1)k}^{i}$, and targets set for responses from above $\mathbf{R}_{(i-1)p}^{i}$. P_{ij} must also satisfy design constraints $\mathbf{g}_{ij}$ and $\mathbf{h}_{ij}$. The generalized formulation for problem P_{ij} is:

$$
\begin{aligned}
\min_{\bar{\mathbf{x}}_{ij},\mathbf{y}_{(i+1)j}^{i}} \quad & \left\|\mathbf{w}_{ij}^{R}\circ\left(\mathbf{R}_{ij}^{i}-\mathbf{R}_{ij}^{i-1}\right)\right\|_2^2+\left\|\mathbf{w}_{(i-1)p}^{R}\circ\left(\mathbf{R}_{(i-1)p}^{i}-\mathbf{R}_{(i-1)p}^{i-1}\right)\right\|_2^2+\left\|\mathbf{S}_j\mathbf{w}_{ip}^{y}\circ\left(\mathbf{S}_j\mathbf{y}_{ip}^{i-1}-\mathbf{y}_{ij}^{i}\right)\right\|_2^2 \qquad (3)\\
& +\sum_{k\in\mathcal{C}_{ij}}\left\|\mathbf{w}_{(i+1)k}^{R}\circ\left(\mathbf{R}_{(i+1)k}^{i}-\mathbf{R}_{(i+1)k}^{i+1}\right)\right\|_2^2+\sum_{k\in\mathcal{C}_{ij}}\left\|\mathbf{w}_{ik}^{R}\circ\left(\mathbf{R}_{ik}^{i}-\mathbf{R}_{ik}^{i+1}\right)\right\|_2^2\\
& +\sum_{k\in\mathcal{C}_{ij}}\left\|\mathbf{S}_k\mathbf{w}_{(i+1)j}^{y}\circ\left(\mathbf{S}_k\mathbf{y}_{(i+1)j}^{i}-\mathbf{y}_{(i+1)k}^{i+1}\right)\right\|_2^2\\
\text{subject to} \quad & \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij})\leq\mathbf{0},\ \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij})=\mathbf{0}\\
\text{where} \quad & \mathbf{R}_{ij}^{i}=\mathbf{S}_p\mathbf{r}_{ij}(\bar{\mathbf{x}}_{ij}),\ \mathbf{R}_{ik}^{i}=\mathbf{S}_k\mathbf{r}_{ij}(\bar{\mathbf{x}}_{ij}) \quad k\in\mathcal{C}_{ij}\\
& \bar{\mathbf{x}}_{ij}=\left[\mathbf{x}_{ij}^{\mathrm{T}},\mathbf{y}_{ij}^{\mathrm{T}},\mathbf{R}_{(i+1)k}^{i\,\mathrm{T}},\mathbf{R}_{(i-1)p}^{i\,\mathrm{T}}\right]^{\mathrm{T}} \quad k\in\mathcal{C}_{ij}
\end{aligned}
$$

If an element is at the top level of a system, terms 2 and 3 of the objective do not exist, and the targets $\mathbf{R}_{ij}^{i-1}$ are external, fixed targets. If an element has no child elements, terms 4–6 of the objective do not exist. To clarify the relationship between ATC and CO terminology, shared variables (denoted by $\mathbf{x}_s$ in CO) are equivalent to linking variables (denoted by $\mathbf{y}$ in ATC), and all coupling variables are considered responses in ATC. However, responses that correspond to external fixed targets are not coupling variables.

Parent elements coordinate interaction between child elements either explicitly through shared values, or implicitly through integration analysis at the parent element. The ATC process is a sharp distinction from the nested optimization of CO. In ATC, each problem P_{ij} is executed independently, i.e., updated responses or targets from other elements are not required during execution. P_{ij} is solved completely while holding all input parameters fixed. An element's input parameters include both targets and responses sent to an element from a parent, termed upper parameters ($\mathbf{R}_{ij}^{i-1}$, $\mathbf{R}_{(i-1)p}^{i-1}$, $\mathbf{S}_j\mathbf{y}_{ip}^{i-1}$), and targets and responses sent from a child to an element, termed lower parameters ($\mathbf{R}_{ik}^{i+1}$, $\mathbf{R}_{(i+1)k}^{i+1}$, $\mathbf{y}_{(i+1)k}^{i+1}$). An external coordinating algorithm is required to execute the ATC process, and can begin by using guesses for input parameters to the top-level problem. This produces upper parameters for child elements below, which require guesses for their lower parameters (assuming more than two levels exist).

One effective coordination strategy is a bi-level nested approach—solve the top level first, and then completely solve all lower levels in a nested fashion before updating the lower parameters for the top level problem. This process is repeated until the top level penalty terms stop changing. For example, consider a three-element, three level-system. P_{1A} is solved and sends upper parameters to P_{2B}. The solution to P_{2B} generates upper parameters used to solve P_{3C}, which then returns lower parameters to P_{2B}, which is solved again. The coordination algorithm alternates between the solution of P_{2B} and P_{3C}, updating upper and lower parameters, respectively, until the penalty terms in P_{2B} stop changing. The pertinent results from P_{2B} are then passed to P_{1A} as lower parameters. This is repeated until convergence of P_{1A}. This nested coordination process requires several executions of the top level problem, and progressively more executions of problems lower on the hierarchy. Other potentially more efficient coordination strategies exist, but the ATC convergence proof [15] is based on nested coordination.

If the external, fixed, top-level targets are unattainable, the penalty terms will not approach zero unless the weights are dynamically adjusted and approach infinity [12], making it impractical to achieve a perfectly consistent system. Weighting update methods newly applied to ATC allow the practitioner to achieve solutions with an acceptably low level of inconsistency when targets are unattainable [12, 14].

6 Example Problem 1

This section details the anchor design problem first presented in [16]. The analysis is partitioned by physical object, but has an MDO flavor due to feedback coupling. A baseline design solution is presented, followed by CO and ATC formulations and solutions. This exercise provides a means to communicate the nuances of each method, and to directly compare solution processes, terminology, and communication patterns.

6.1 Anchor Design Problem

Consider an anchoring system with n_b cantilever beams of length L with solid circular cross sections of diameter d_i interconnected with $n_r = n_b - 1$ solid cylindrical rods of diameter d_{rj} with unique length l_{rj}, attached to the beams with pin joints (Figure 3). A downward force F_1 is applied at the pin joint of beam 1. Some anchoring systems (such as multiple tent stakes) require several beams to distribute the load and prevent damage to the foundation.

The system is statically indeterminate because the number of unknown forces and moments exceeds the number of equilibrium equations. Additional compatibility constraints are used to facilitate solution. It is assumed that linearity holds, materials are homogeneous, and body forces, buckling, and stress concentrations may be neglected. The anchor design problem is posed as a mass allocation problem, i.e., for a maximum allowed system mass m_{allow}, find beam and rod diameters that minimize the deflection at the point of force application, subject to bending and axial stress constraints, and transmitted force constraints. The last set of constraints approximately ensure integrity of the foundation. Table 1 summarizes design variables and parameters.

Table 1: Anchor design problem variables and parameters

local design variables	x_i:	d_i	$i = 1 \ldots n_b$
shared design variables	x_{si}:	d_{rj}	$j = 1 \ldots n_r$
design parameters	$\mathbf{p}$:	$L, l_{rj}, E, \rho, \sigma_{allow}, m_{allow}, Ft_{allow}$	$j = 1 \ldots n_r$

Ft_{allow} is the allowable transmitted force, ρ is the material density, σ_{allow} is the allowable stress, and E is the modulus of elasticity. The design optimization problem in negative-null form is presented in Equation (4).

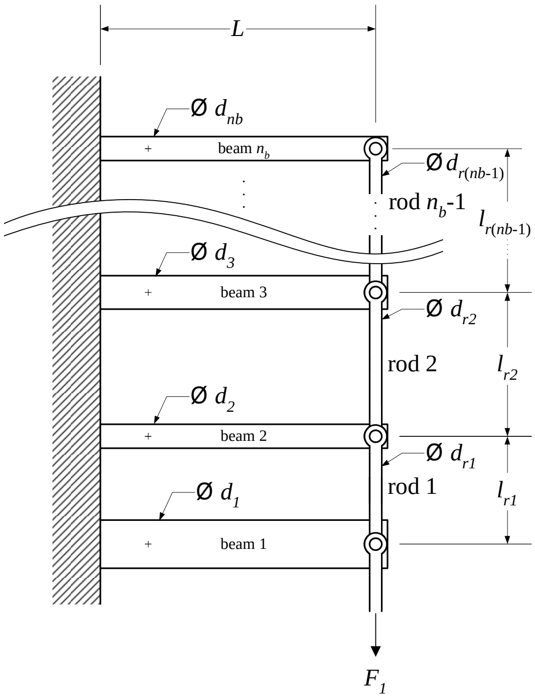

Figure 3: Schematic of the anchor design problem

$$
\begin{aligned}
\min_{\mathbf{x}=[\mathbf{x}_\ell^{\mathbf{T}},\mathbf{x}_\mathbf{s}^{\mathbf{T}}]^{\mathbf{T}}} \quad f(\mathbf{x}) &= \delta_1(\mathbf{x}) \\
\text{subject to} \quad g_{1i}(\mathbf{x}) &= \sigma_{bi}(\mathbf{x}) - \sigma_{allow} \leq 0 \quad i = 1 \ldots n_b \\
g_{2j}(\mathbf{x}) &= \sigma_{aj}(\mathbf{x}) - \sigma_{allow} \leq 0 \quad j = 1 \ldots n_r \\
g_3(\mathbf{x}) &= \sum_{i=1}^{n_b} m_{bi}(\mathbf{x}) + \sum_{j=1}^{n_r} m_{rj}(\mathbf{x}) - m_{allow} \leq 0 \\
g_{4i}(\mathbf{x}) &= Ft_i(\mathbf{x}) - Ft_{allow} \leq 0 \quad i = 1 \ldots n_b
\end{aligned} \tag{4}
$$

Solving the design problem requires prediction of each beam end deflection (δ_i), the extension of each rod (δ_{rj}), the bending stress in each beam (σ_{bi}), the axial stress in each rod (σ_{aj}), the transmitted force at each beam (Ft_i), and the beam and rod masses (m_{bi} & m_{ri}), for given beam and rod diameters. The following relations from fundamental solid mechanics theory were used in the development of the analysis model:

$$\sigma_b = \frac{Mc}{I}, \quad I = \frac{\pi}{64}d^4, \quad \sigma_a = \frac{P}{A_c}, \quad \delta_b = \frac{PL^3}{3EI}, \quad \delta_r = \frac{PL}{EA_c}$$

M is the bending moment in a beam at the base, I is the beam area moment of inertia, P is the applied load, and A_c is the beam cross-sectional area. F_i is the downward force exerted at the end of beam i, and F_j is the axial load present in rod j. The responses of interest may be generalized and grouped into the three categories introduced below.

Intermediate or Bottom Beam

$$\delta_i = \frac{64L^3(F_i - F_{i+1})}{3\pi E d_i^4} \qquad (5a)$$

$$\sigma_{bi} = \frac{32L(F_i - F_{i+1})}{\pi d_i^3} \qquad (5b)$$

$$m_{bi} = \frac{\pi}{4} d_i^2 L \rho \qquad (5c)$$

Arbitrary Rod

$$\delta_{rj} = \frac{4F_{j+1} l_{rj}}{\pi E d_{rj}^2} \qquad (6a)$$

$$\sigma_{aj} = \frac{4F_{j+1}}{\pi d_{rj}^2} \qquad (6b)$$

$$m_{rj} = \frac{\pi}{4} d_{rj}^2 l_{rj} \rho \qquad (6c)$$

Top Beam

$$\delta_{n_b} = \frac{64L^3 F_{n_b}}{3\pi E d_{n_b}^4} \qquad (7a)$$

$$\sigma_{bn_b} = \frac{32LF_{n_b}}{\pi d_{n_b}^3} \qquad (7b)$$

$$m_{bn_b} = \frac{\pi}{4} d_{n_b}^2 L \rho \qquad (7c)$$

The transmitted force Ft_i for each beam is $F_i - F_{i+1}$. Compatibility conditions require that the deflection of the end of beam i is equal to the deflection of beam $i+1$ plus the extension of the connecting rod ($j = i$), i.e., $\delta_i = \delta_{i+1} + \delta_{ri}$. The deflection of a beam is dependent upon the deflection of the surrounding beams, introducing coupling into the system. A three-beam anchor analysis may be partitioned into three subspaces, where subspace i contains the analysis of beam i and rod i. Figure 4 illustrates this partitioning, along with communication paths and functional relationships. The quantity m_i is the sum of all masses in subspace i. The functional relationshsips can be derived from Equations (5)–(7). The resulting shared, local, and coupling variables are identified in Table 2. The total design vector is $\mathbf{x} = [\mathbf{x}_\ell^{\mathrm{T}} \mathbf{x}_s^{\mathrm{T}}]^{\mathrm{T}}$, where $\mathbf{x}_\ell = [d_1, d_2, d_3]^{\mathrm{T}}$ and $\mathbf{x}_s = [d_{r1}, d_{r2}]^{\mathrm{T}}$.

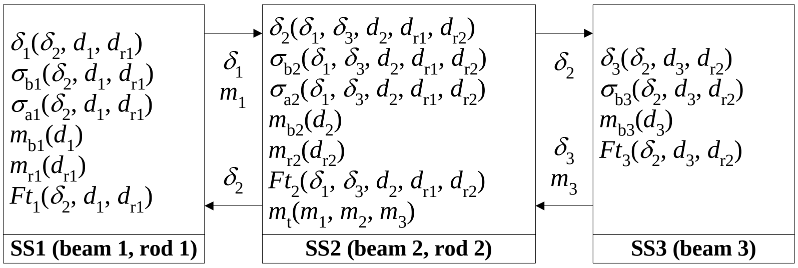

Figure 4: Suggested partitioning of the three-beam anchor design problem

Table 2: Anchor design problem variable designations

$SS1$	$SS2$	$SS3$
$\mathbf{y}_{21} = [\delta_1,\ m_1]^T$	$\mathbf{y}_{21} = [\delta_1,\ m_1]^T$	$y_{32} = \delta_2$
$y_{12} = \delta_2$	$\mathbf{y}_{23} = [\delta_3,\ m_3]^T$	$\mathbf{y}_{23} = [\delta_3,\ m_3]^T$
$x_{\ell 1} = d_1$	$y_{12} = \delta_2$	$x_{\ell 3} = d_3$
$x_{s1} = d_{r1}$	$y_{32} = \delta_2$	$x_{s3} = d_{r2}$
	$x_{\ell 2} = d_2$	
	$\mathbf{x}_{s2} = [d_{r1},\ d_{r2}]^T$	

Table 3: Design variable and parameter values for anchor problem analysis

design variable	value	units	parameter	value	units
d_1	0.05	meters	F_1	1000	Newtons
d_2	0.05	meters	L	1.000	meters
d_3	0.05	meters	l_{r1}	1.000	meters
d_{r1}	0.005	meters	l_{r2}	1.000	meters
d_{r2}	0.005	meters	E	70	GPa
			ρ	2700	kg/meter^3

Stress, mass and transmitted force limits were set to $\sigma_{allow} = 127\ MPa$, $m_{allow} = 7\ kg$, and $Ft_{allow} = 400\ N$. A baseline solution to Equation (4) was found using standard optimization, i.e., performing a complete analysis at every optimization iteration. Using the parameter values and the starting design point from Table 3, the solution was found to be $\mathbf{x}_* = [d_1\ d_2\ d_3\ d_{r1}\ d_{r2}]^{\mathrm{T}} = [.035\ .035\ .029\ .005\ .003]^{\mathrm{T}}$. The minimum deflection at beam 1 was $\delta_1^* = 27.0$ millimeters. Increasing d_1 or d_2 would have violated the corresponding transmitted force constraint. The mass constraint was also active, but no stress constraints were active. If the applied force F_1 was instead less than twice the transmitted force limit, zero mass would be allocated to the third beam. Additional beams are required only if the existing beams cannot distribute the force well enough to prevent failure of the foundation. It was observed that three local optima exist. Which optimum is found depends on the starting point. The design point presented above corresponds to the global optimum.

6.2 CO Implementation

This section presents the CO formulation of the anchor design problem and the corresponding results. The partitioning scheme described in the previous section is utilized. Each subspace is given the required system targets $\hat{\mathbf{z}}_i$ by the system optimizer, and returns to the system optimizer the minimum deviation from the system targets (f_i^*) with respect to local design variables and input coupling variables. Subspace 1 also returns the system objective function $f_s = \delta_1$. Each subspace is responsible to satisfy its own local bending and axial stress constraints (g_{1i} and g_{2i} respectively), as well as its maximum transmitted force constraint (g_{4i}). Subspace 2 also works to satisfy the mass constraint g_3. The system and subspace targets are defined below, followed by the CO formulation of the anchor problem.

$$\hat{\mathbf{z}} = [x_{s1},\ \mathbf{x}_{s2}^{\mathrm{T}},\ x_{s3},\ y_{12},\ \mathbf{y}_{21}^{\mathrm{T}},\ \mathbf{y}_{23}^{\mathrm{T}},\ y_{32}]^{\mathrm{T}}, \hat{\mathbf{z}}_1 = [x_{s1},\ y_{12},\ \mathbf{y}_{21}^{\mathrm{T}}]^{\mathrm{T}}, \hat{\mathbf{z}}_2 = [\mathbf{x}_{s2}^{\mathrm{T}},\ y_{12},\ \mathbf{y}_{21}^{\mathrm{T}},\ \mathbf{y}_{23}^{\mathrm{T}}\ y_{32}]^{\mathrm{T}}, \hat{\mathbf{z}}_3 = [x_{s3},\ \mathbf{y}_{23}^{\mathrm{T}},\ y_{32}]^{\mathrm{T}}$$

System Optimizer

$$\begin{aligned} \min_{\hat{\mathbf{z}}} \quad & f_s(\hat{\mathbf{z}}) \\ \text{subject to} \quad & \mathbf{f}^*(\hat{\mathbf{z}}) = \mathbf{0} \end{aligned} \tag{8}$$

Subspace 1 Optimizer

$$\begin{aligned} \min_{\mathbf{x}_{\ell 1}, \mathbf{y}_{1j}} \quad & f_1(\mathbf{x}_{\ell 1}, \mathbf{y}_{1j}) = ||\mathbf{z}_1 - \hat{\mathbf{z}}_1||_2^2 \\ \text{subject to} \quad & \mathbf{g}_{11}(\mathbf{x}_{\ell 1}, \mathbf{y}_{1j}) \le \mathbf{0} \\ & \mathbf{g}_{21}(\mathbf{x}_{\ell 1}, \mathbf{y}_{1j}) \le \mathbf{0} \\ & \mathbf{g}_{41}(\mathbf{x}_{\ell 1}, \mathbf{y}_{1j}) \le \mathbf{0} \end{aligned} \tag{9}$$

Subspace 2 Optimizer

$$\begin{aligned} \min_{\mathbf{x}_{\ell 2}, \mathbf{y}_{2j}} \quad & f_2(\mathbf{x}_{\ell 2}, \mathbf{y}_{2j}) = ||\mathbf{z}_2 - \hat{\mathbf{z}}_2||_2^2 \\ \text{subject to} \quad & \mathbf{g}_{12}(\mathbf{x}_{\ell 2}, \mathbf{y}_{2j}) \le \mathbf{0} \\ & \mathbf{g}_{22}(\mathbf{x}_{\ell 2}, \mathbf{y}_{2j}) \le \mathbf{0} \\ & \mathbf{g}_{42}(\mathbf{x}_{\ell 2}, \mathbf{y}_{2j}) \le \mathbf{0} \\ & \mathbf{g}_{3}(\mathbf{x}_{\ell 2}, \mathbf{y}_{2j}) \le \mathbf{0} \end{aligned} \tag{10}$$

Subspace 3 Optimizer

$$\begin{aligned} \min_{\mathbf{x}_{\ell 3}, \mathbf{y}_{3j}} \quad & f_3(\mathbf{x}_{\ell 3}, \mathbf{y}_{3j}) = ||\mathbf{z}_3 - \hat{\mathbf{z}}_3||_2^2 \\ \text{subject to} \quad & \mathbf{g}_{13}(\mathbf{x}_{\ell 3}, \mathbf{y}_{3j}) \le \mathbf{0} \\ & \mathbf{g}_{43}(\mathbf{x}_{\ell 3}, \mathbf{y}_{3j}) \le \mathbf{0} \end{aligned} \tag{11}$$

The same parameters and starting point were used as in the baseline solution, and all values were scaled appropriately. The resulting solution, $\mathbf{x}_* = [d_1\ d_2\ d_3\ d_{r1}\ d_{r2}]^{\mathrm{T}} = [.033\ .025\ .029\ .007\ .008]^{\mathrm{T}}$ with an objective value of $\delta_1 = 42.0$ millimeters, is relatively close to the solution, considering the expected numerical accuracy for algorithm parameters used. Improved accuracy can be obtained by tightening convergence tolerances at the cost of increased computation time. To overcome problems meeting KKT conditions at convergence, the SQP constraint tolerances were loosened to 10^{-4}. The necessity of this relaxation places a limitation on the attainable accuracy using the basic CO formulation.

6.3 ATC Implementation

To effectively demonstrate distinctions between CO and ATC, we will use the exact partitioning scheme from the CO implementation, illustrated in Figure 4. In the CO case we had one option for this partitioning—all subspaces existed at the lower of the two levels in the architecture. ATC allows for multiple hierarchical configurations. We can choose a three level hierarchy with either subspace 1 or 3 at the top level, or a two level hierarchy with subspace 2 chosen as the top level, and the remaining subspaces at the lower level. We choose the latter for this implementation.

To begin creating the ATC formulation, we first identify which values are shared between the top level and the lower level. Subspace 2 (top level, element A) shares d_{r1}, δ_1, δ_2, and m_1 with subspace 1 (bottom level, element B), and shares d_{r2}, δ_2, δ_3, and m_3 with subspace 3 (bottom level, element C). Using ATC terminology, d_{r1} and d_{r2} are linking variables, and the other shared values are responses. Recall that according to MDO terminology these values are shared variables and coupling variables, respectively. We create copies of all pertinent shared values in each element, and penalties for inconsistencies in these shared values. Each element retains local design constraints, and is optimized with respect to local variables and any input shared values. The top level target for the system objective function δ_1 in element A is set as δ_1^*, the baseline solution, ensuring target attainability. It is important to emphasize that *a priori* knowledge of δ_1^* is not required for solution via ATC, but facilitates efficiency. The fully expanded ATC formulation, without weights, is presented in Equations (12)–(14). ATC indices are enclosed in parentheses for clarity.

Problem P_{1A}, Element A

$$\begin{aligned} \min_{\bar{\mathbf{x}}_{1A}=[d_2,\delta_1,\delta_3,m_1,m_3]^{\mathrm{T}}, y_{2B}^1=d_{r1}, y_{2C}^1=d_{r2}} \quad & \left(\delta_{1(1A)}^{(1)} - \delta_1^*\right)^2 + \left(d_{r1(2B)}^{(1)} - d_{r1(2B)}^{(2)}\right)^2 + \left(d_{r2(2C)}^{(1)} - d_{r2(2C)}^{(2)}\right)^2 \\ & + \left(\delta_{1(2B)}^{(1)} - \delta_{1(2B)}^{(2)}\right)^2 + \left(\delta_{2(1B)}^{(1)} - \delta_{2(1B)}^{(2)}\right)^2 + \left(\delta_{2(1C)}^{(1)} - \delta_{2(1C)}^{(2)}\right)^2 \\ & + \left(\delta_{3(2C)}^{(1)} - \delta_{3(2C)}^{(2)}\right)^2 + \left(m_{1(2B)}^{(1)} - m_{1(2B)}^{(2)}\right)^2 + \left(m_{3(2C)}^{(1)} - m_{3(2C)}^{(2)}\right)^2 \\ \text{subject to} \quad & \mathbf{g}_{12}(\bar{\mathbf{x}}_{1A}) \le \mathbf{0} \\ & \mathbf{g}_{22}(\bar{\mathbf{x}}_{1A}) \le \mathbf{0} \\ & \mathbf{g}_{42}(\bar{\mathbf{x}}_{1A}) \le \mathbf{0} \\ & \mathbf{g}_{3}(\bar{\mathbf{x}}_{1A}) \le \mathbf{0} \end{aligned} \tag{12}$$

Problem P_{2B}, Element B

$$\begin{aligned} \min_{\bar{\mathbf{x}}_{2B}=[d_1,d_{r1},\delta_2]^{\mathrm{T}}} \quad & \left(d_{r1(2B)}^{(1)} - d_{r1(2B)}^{(2)}\right)^2 + \left(\delta_{1(2B)}^{(1)} - \delta_{1(2B)}^{(2)}\right)^2 \\ & + \left(\delta_{2(1A)}^{(1)} - \delta_{2(1A)}^{(2)}\right)^2 + \left(m_{1(2B)}^{(1)} - m_{1(2B)}^{(2)}\right)^2 \\ \text{subject to} \quad & \mathbf{g}_{11}(\bar{\mathbf{x}}_{2B}) \le \mathbf{0} \\ & \mathbf{g}_{21}(\bar{\mathbf{x}}_{2B}) \le \mathbf{0} \\ & \mathbf{g}_{41}(\bar{\mathbf{x}}_{2B}) \le \mathbf{0} \end{aligned} \tag{13}$$

Problem P_{2C}, Element C

$$\begin{aligned} \min_{\bar{\mathbf{x}}_{2C}=[d_3,d_{r2},\delta_2]^{\mathrm{T}}} \quad & \left(d_{r2(2C)}^{(1)} - d_{r2(2C)}^{(2)}\right)^2 + \left(\delta_{2(1A)}^{(1)} - \delta_{2(1A)}^{(2)}\right)^2 \\ & + \left(\delta_{3(2C)}^{(1)} - \delta_{3(2C)}^{(2)}\right)^2 + \left(m_{3(2C)}^{(1)} - m_{3(2C)}^{(2)}\right)^2 \\ \text{subject to} \quad & \mathbf{g}_{13}(\bar{\mathbf{x}}_{2C}) \le \mathbf{0} \\ & \mathbf{g}_{43}(\bar{\mathbf{x}}_{2C}) \le \mathbf{0} \end{aligned} \tag{14}$$

Again the parameters from Table 3 were used, and each scaled penalty term was given a weight of 10. Convergence criteria required that the objective function from problem P_{1A} (dominated by penalty terms) stopped changing within a tolerance of 0.001. The resulting solution was $\mathbf{x}_* = [d_1\ d_2\ d_3\ d_{r1}\ d_{r2}]^T = [.040\ .030\ .026\ .006\ .003]^T$ with an objective value of $\delta_1 = 25.4$ millimeters, also relatively close to the baseline solution. Tighter convergence criteria would effect a better result, at the expense of additional ATC iterations.

7 Example Problem 2

This section introduces an example problem that can be partitioned first by object, and then by discipline, in order to illustrate nested ATC-MDO, a newly developed formulation, and to elucidate the unique strengths of ATC and CO. A brief conceptual description of the design problem is provided, followed by the formulation for nesting CO within ATC.

7.1 Electric Water Pump Design Problem

Consider a sump pump used to remove water from a basement where an electric motor drives a centrifugal pump via a v-belt. The sump pump must supply a minimum required pressure head and volumetric flow rate, while occupying no more than a prescribed geometric space and avoiding failure of the belt. The design objective is to minimize electrical power requirements. The system analysis may be separated into two objects—motor and pump—and the motor analysis can be separated into two disciplines—electromechanical and thermal.

The system analysis depends on functions supplied by the motor and pump simulations, represented by surrogate models. The motor model predicts the torque-speed curve and electrical power requirements for a given motor design. Within the motor model the electromechanical analysis calculates the motor current I, torque τ_m, and electric input power P_e at k_1 distinct rotational speeds ω, given values for geometric motor design variables $\mathbf{x}_m$ and the system temperature $\mathbf{T} = [T_{\omega 1}, T_{\omega 2} \dots T_{\omega k_1}]^{\mathrm{T}}$ at each rotational speed, evaluated by the thermal discipline. The thermal discipline requires values for $\mathbf{x}_m$ and $\mathbf{I}$ (the current for each rotational speed) in order to predict $\mathbf{T}$. Feedback coupling exists between these two disciplines due to the two-way communication of $\mathbf{I}$ and $\mathbf{T}$. Intuitively, the coupling exists because an increase in current results in higher temperatures and resistivity, which in turn impacts motor current. The coupled analysis of these two disciplines can be performed using fixed point iteration [3]. The resulting k_1 values of τ_m are used to construct a linear approximation for $\tau_m(\omega)$. The coefficients for this surrogate model ($\mathbf{c}_m$) are communicated to the system analysis.

The centrifugal pump model predicts the volumetric flow rate Q, output pressure H, and the required drive torque τ_p for a given pump design $\mathbf{x}_m$ and rotational speed. These responses are evaluated at k_2 distinct rotational speeds, which are then used to create polynomial approximations of $Q(\omega)$, $H(\omega)$, and $\tau_p(\omega)$. Again, the corresponding coefficients ($\mathbf{c}_p$) are provided to the system analysis.

The v-belt couples the motor and pump, and provides a means of adjusting the available motor torque curve. Figure 5 illustrates how a belt speed ratio of 1.8 modifies a motor torque curve. The design problem requires quantification of the maximum flow rate, which can be determined by finding the maximum speed ω_{max} the motor is capable of driving the pump at, which is located where the pump torque curve and the modified motor torque curve intersect. Since the belt integrates the system, it is natural to incorporate the belt analysis with the system analysis. The belt analysis considers mechanical efficiency and mechanical advantage to generate a modified $\tau_m(\omega)$ curve, and also evaluates stress in the belt to predict failure.

In summary, motor design variables are provided to the electromechanical and thermal disciplines. Fixed point iteration is used to evaluate the motor performance, and the coefficients representing $\tau_m(\omega)$ are supplied to the system analysis. The pump analysis finds the flow rate, required torque, and pressure for given values of the pump design variables, and the corresponding polynomial coefficients are passed to the system analysis. The system analysis uses the surrogate performance curves and the belt design variable and analysis to determine ω_{max} and the corresponding values for P_e (derived from $\tau_m(\omega)$), H, and Q.

The electric water pump design problem is formally presented in Equation (15). The complete design vector is $\mathbf{x} = [\mathbf{x}_p^{\mathrm{T}}, \mathbf{x}_m^{\mathrm{T}}, x_b]^{\mathrm{T}}$. The first four constraints have already been described. The last two constraints require that the wire windings do not interfere with the motor poles and that the inner pump radius is smaller than the outer radius.

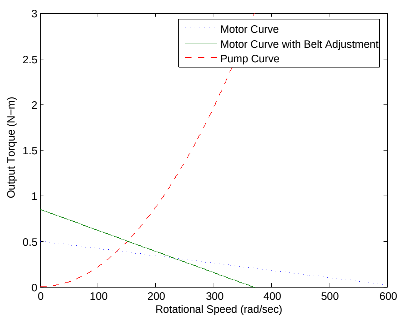

Figure 5: Surrogate torque curves for motor, motor with belt adjustment, and pump

$$\begin{aligned} \min_{\mathbf{x}} \quad & P_e \\ \text{subject to} \quad & g_1(\mathbf{x}) = Q_{min} - Q \leq 0 \\ & g_2(\mathbf{x}) = \sigma - \sigma_{max} \leq 0 \\ & g_3(\mathbf{x}) = H_{min} - H \leq 0 \\ & g_4(\mathbf{x}) = V - V_{max} \leq 0 \\ & g_5(\mathbf{x}) = (D_o + d) - D \leq 0 \\ & g_6(\mathbf{x}) = r_1 - r_2 \leq 0 \end{aligned} \tag{15}$$

7.2 Nested ATC-MDO Implementation

Many design organizations and analysis structures posses a matrix structure, i.e., aligned by both discipline and object [6]. This allows for both depth of knowledge or analysis, and responsiveness to requirements based on an object perspective. A new formulation exploits the natural strengths of ATC and CO by combining them in a nested formulation that maps directly to a matrix organization or design problem. ATC is used to coordinate the design of the overall system, and an MDO method (CO in this case) is used to solve any tightly coupled design problems within ATC elements. In example problem 2, the motor subsystem is further decomposed into two disciplines. All design variables are shared between them, resulting in a particularly simple nested ATC-CO formulation, shown in Equations (16)–(20). Problems P_{1A}, P_{2B}, and P_{2C} are coordinated and solved as usual for ATC, with the exception that for every iteration of the P_{2B} optimization, both $SS1$ and $SS2$ problems must be completed, supplying updated values of $\mathbf{f}^*(\hat{\mathbf{z}})$ and $f_s(\hat{\mathbf{z}})$ to P_{2B}. Surrogate models are fit in the appropriate subspaces, allowing evaluation of $f_s(\hat{\mathbf{z}})$ at the subspace level. An open question is whether nested ATC-MDO has advantages for matrix decomposed problems over application of the extended ATC formulation that allows for feedback coupling. Results and further investigation of nested ATC-MDO are part of future work.

Problem P_{1A}

$$\begin{aligned} \min_{\bar{\mathbf{x}}_{1A}=\left[\mathbf{x}_b^{\mathrm{T}},\mathbf{c}_m^{\mathrm{T}},\mathbf{c}_p^{\mathrm{T}}\right]^{\mathrm{T}}} \quad & \left(P_{e(1A)}^{(1)} - 0\right)^2 + \left\|\mathbf{c}_{m(2B)}^{(1)} - \mathbf{c}_{m(2B)}^{(2)}\right\|_2^2 + \left\|\mathbf{c}_{p(2B)}^{(1)} - \mathbf{c}_{p(2B)}^{(2)}\right\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_i(\bar{\mathbf{x}}_{1A}) \leq \mathbf{0} \quad i = 1\ldots 3 \end{aligned} \tag{16}$$

Problem P_{2B}, Motor Subsystem

$$\begin{aligned} \min_{\hat{\mathbf{z}}=\left[\mathbf{x}_m^{\mathrm{T}},\mathbf{I}^{\mathrm{T}},\mathbf{T}^{\mathrm{T}}\right]^{\mathrm{T}}} \quad & f_s(\hat{\mathbf{z}}) = \left\|\mathbf{c}_{m(2B)}^{(1)} - \mathbf{c}_{m(2B)}^{(2)}\right\|_2^2 \\ \text{subject to} \quad & \mathbf{f}^*(\hat{\mathbf{z}}) = \mathbf{0} \end{aligned} \tag{17}$$

Problem P_{2C}, Pump Subsystem

$$\min_{\bar{\mathbf{x}}_{2C}=\mathbf{x}_p} \quad \left\|\mathbf{c}_{p(2B)}^{(1)} - \mathbf{c}_{p(2B)}^{(2)}\right\|_2^2 \tag{18}$$

$SS1$, Electromechanical Subspace

$$\begin{aligned} \min_{\mathbf{x}_m,\mathbf{T}} \quad & \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_i(\mathbf{x}_m, \mathbf{T}) \leq \mathbf{0} \quad i = 4\ldots 6 \\ \text{where} \quad & \mathbf{z} = \left[\mathbf{x}_m^{\mathrm{T}}, \mathbf{I}(\mathbf{x}_m, \mathbf{T})^{\mathrm{T}}, \mathbf{T}^{\mathrm{T}}\right]^{\mathrm{T}} \end{aligned} \tag{19}$$

$SS2$, Thermal Subspace

$$\begin{aligned} \min_{\mathbf{x}_m,\mathbf{I}} \quad & \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2 \\ \text{subject to} \quad & \mathbf{g}_i(\mathbf{x}_m, \mathbf{I}) \leq \mathbf{0} \quad i = 4\ldots 6 \\ \text{where} \quad & \mathbf{z} = \left[\mathbf{x}_m^{\mathrm{T}}, \mathbf{I}^{\mathrm{T}}, \mathbf{T}(\mathbf{x}_m, \mathbf{T})^{\mathrm{T}}\right]^{\mathrm{T}} \end{aligned} \tag{20}$$

8 Concluding Remarks

The mathematical formulations of ATC and CO appear to be similar, particularly when considering MCO and the ATC penalty formulation. Both use targets and penalty functions to enforce system consistency, and correlation exists between ATC weighting update methods and the MCO smoothing algorithm. Nevertheless, four key distinctions exist: solution process, targets and communication patterns, intended structure of corresponding design problems, and paradigm.

It was illustrated that CO and ATC utilize nested optimization and coordination solution processes, respectively. The CO system optimizer is executed once, and performs the system coordination. Because subspace convergence is required at every iteration, this single execution can be time consuming. ATC utilizes an external coordination algorithm, and the top–most optimization problem is executed several times. However, each individual execution is independent of other elements, resulting in relatively fast solution time for the top element. Another process difference is that CO relies on the optimization algorithm to tolerate compatibility constraint violation before convergence, while ATC (whether in its original form or penalty formulation) explicitly relaxes compatibility constraints and does not require the algorithm to tolerate compatibility constraint violation.

Targets are set for all shared and coupling variables in each methodology, but where in the structure and when in the process these targets are set differs. The CO system optimizer sets targets for all shared and input and output coupling variables for each subspace, and subspaces set no targets for the system optimizer (they cannot, due to the nested optimization process). An ATC parent element sets targets for all shared and output coupling variables for each of its child elements, while the child elements set targets for coupling variables input to the child element from its parent element, and for any coupling variables and responses input to the child element from below. Child elements can set targets for parent elements because of the nested coordination process structure. Another important communication difference concerns the values passed up to the top level. In CO the subspace optimizers return their optimal objective function values and the system objective. In contrast, ATC child elements return only optimal decision variable values and pertinent analysis responses.

CO is intended for integration and optimization of discipline-based analyses with a non-hierarchical structure and a column based functional dependency table. ATC is intended for solving hierarchical target setting problems, normally partitioned by object, with a hybrid functional dependency table. Either method can be manipulated to fit unintended problem types, but the resulting formulation may exhibit undesirable traits. For example, applying CO to a problem with system analysis requires the analysis to be brought to the subspace level, increasing the communication requirements over an approach that retains analysis at the top level. The second example problem demonstrated how the natural strengths of each formulation may be exploited to approach a design problem in a complementary manner.

The CO and ATC paradigms for problem solving are substantially different—practitioners take on contrasting perspectives when formulating problems. The needs a method's development was based upon colors its nature. CO focuses on integrating non-hierarchical discipline-based analyses, and a user thinks in terms of IO relations between disciplines on the same level. ATC is centered on product development problems. The practitioner considers how a problem can be conceived as a hierarchy of object-based elements, and thinks in terms of a system analysis integrating subsystem analyses. Each approach is naturally suited to certain classes of problems. Understanding across these paradigms and flexibility to use each approach where most appropriate will result in more successful implementations of system optimization methods.

9 Acknowledgements

This work was partially supported by a US NSF Graduate Research Fellowship and Grant DMI0503737, and by the Automotive Research Center, a US Army Center of Excellence at the University of Michigan. Any opinions expressed in this publication are only those of the authors. Special thanks are given to Brian Roth and Ilan Kroo of Stanford for their collaborative efforts toward this work.

10 References

[1] American Institute for Aeronautics and Astronautics Inc. (AIAA). Current state of the art in multidisciplinary design optimization. Technical report, MDO Technical Committee, January 1991.

[2] E.J. Cramer, J.E. Dennis Jr., P.D. Frank, R.M. Lewis, and G.R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal of Optimization*, 4:754–776, 1994.

[3] J.T. Allison, M. Kokkolaras, and P.Y. Papalambros. On the impact of coupling strength on complex system optimization for single-level formulations. In *ASME Design Engineering Technical Conference DETC2005-84790*, Long Beach CA, September 24–28 2005. To appear.

[4] R.D. Braun. *Collaborative Optimization: An Architecture For Large-Scale Distributed Design*. PhD thesis, Stanford University, April 1996.

[5] N. Michelena, H.M. Kim, and P.Y. Papalambros. A system partitioning and optimization approach to target cascading. In *Proceedings of the 12th International Conference on Engineering Design, Munich, Germany*, 1999.

[6] R.L. Daft. *Organization Theory and Design*. Thomson South-Western, Ohio, eigth edition, 2004.

[7] L. Chen, Z. Ding, and S. Li. A formal two-phase method for decomposition of design problems. *Journal of Mechanical Design*, 127:184–195, March 2005.

[8] N.M. Alexandrov and R.M. Lewis. Analytical and computational aspects of collaborative optimization and multidisciplinary design. *AIAA Journal*, 40(2):301–309, February 2002.

[9] A.V. DeMiguel and W. Murray. An analysis of collaborative optimization methods. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, Sept. 6-8 2000.

[10] H.M. Kim. *Target Cascading in Optimal System Design*. PhD thesis, University of Michigan, 2001.

[11] P.Y. Papalambros and D.J. Wilde. *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, New York, second edition, 2000.

[12] J.J. Michalek and P.Y. Papalambros. An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading. *Journal of Mechanical Design*, 127(2):206–214, 2005.

[13] J.J. Michalek and P.Y. Papalambros. Technical brief: Weights, norms, and notation in analytical target cascading. *Journal of Mechanical Design*, 127(3):499–501, 2005.

[14] S. Tosserams, L.F.P. Etman, P.Y. Papalambros, and J.E. Rooda. Augmented lagrangian relaxation for analytical target cascading. In *6th World Congress on Structural and Multidisciplinary Optimization*, May 30–June 3 2005.

[15] N.F. Michelena, H.A. Park, and P.Y. Papalambros. Convergence properties of analytical target cascading. *AIAA Journal*, 41(5):897–905, 2003.

[16] J.T. Allison. Complex system optimization: A review of analytical target cascading, collaborative optimization, and other formulations. Master's thesis, Department of Mechanical Engineering, University of Michigan, 2004.