

**ENHANCED COLLABORATIVE OPTIMIZATION:  
A DECOMPOSITION-BASED METHOD FOR MULTIDISCIPLINARY DESIGN**

**Brian D. Roth and Ilan M. Kroo**

Aircraft Aerodynamics and Design Group  
Department of Aeronautics and Astronautics  
Stanford University  
Stanford, California 94305  
Email: bdroth@stanford.edu

**ABSTRACT**

Astute choices made early in the design process provide the best opportunity for reducing the life cycle cost of a new product. Optimal decisions require reasonably detailed disciplinary analyses, which pose coordination challenges. These types of complex multidisciplinary problems are best addressed through the use of decomposition-based methods, several of which have recently been developed. Two of these methods are collaborative optimization (CO) and analytical target cascading (ATC). CO was conceived in 1994 in response to multidisciplinary design needs in the aerospace industry. Recent progress has led to an updated version, enhanced collaborative optimization (ECO), that is introduced in this paper. ECO addresses many of the computational challenges inherent in CO, yielding significant computational savings and more robust solutions. ATC was formalized in 2000 to address needs in the automotive industry. While ATC was originally developed for object-based decomposition, it is also applicable to multidisciplinary design problems. In this paper, both methods are applied to a set of test cases. The goal is to introduce the ECO methodology by comparing and contrasting it with ATC, a method familiar within the mechanical engineering design community. Comparison of ECO and ATC is not intended to establish the computational superiority of either method. Rather, these two methods are compared as a means of highlighting several promising approaches to the coordination of distributed design problems.

**1 Introduction**

The greatest freedom to exploit potential tradeoffs between aircraft subsystems for the optimization of the design occurs earliest in the design process. However, the capacity to utilize this freedom is limited by the fact that the analysis tools applied at this stage are the most limited [1]. More detailed analysis can contradict the findings of earlier, simpler, analysis. Ideally, detailed analysis should be used as early in the design process as possible [2]. The big challenge in using more detailed models earlier in the process is the enormous detail required to thoroughly describe a complex system such as an aircraft. Consider, for example, the process used to coordinate the preliminary design of the Boeing 777. The 777 preliminary design was developed by 3000 people. Coordination was facilitated through weekly design meetings of 25 lead engineers, each representing 100+ engineers in their specialty. Each leader presented their work once every ten weeks [3]. This approach clearly could not facilitate detailed trade studies to explore the tradeoffs between disciplinary groups.

The ad hoc design approach used for the Boeing 777 worked well primarily because Boeing had 40 years of experience with developing aircraft very similar to the 777 (i.e., a tube and wings). The interaction of different systems was relatively well understood, and statistical data existed that could be leaned on heavily in making preliminary tradeoffs without resorting to highly detailed models of the aircraft [2]. However, for new

concepts without a long history of success from which to draw, unexpected tradeoffs often have an important impact on vehicle performance. This highlights the need for a formalized approach to making early design decisions.

In preliminary design, the central challenge is performing design optimization in a distributed environment made up of distinct disciplinary design teams with individual solution strategies, locally defined variables and constraints, potentially costly computational analyses, and interdisciplinary coupling. The nature of this sort of design environment precludes the use of direct iterative optimization methods, such as those used in many local subsystem design problems. In response to this, a variety of multidisciplinary design optimization methods have been developed that enable a formalized design optimization process in the preliminary design phase. These are related to one another in their use of a coordinating optimization problem to make progress toward optimal designs. The methods differ from each other in how they handle local feasibility, interdisciplinary compatibility, and local design autonomy [2]. One such method is collaborative optimization.

Collaborative optimization (CO) is a method for the design of complex, multidisciplinary systems that was originally proposed [4] in 1994. CO is one of several decomposition-based methods that divides a design problem along disciplinary (or other convenient) boundaries. The idea is to mirror the natural divisions found in aerospace design companies. In these settings, engineers are often divided into design groups by disciplinary expertise. Disciplinary analysis tools tend to be complex in nature, and it is often impractical to integrate multiple analysis codes for the purpose of multidisciplinary optimization. Rather, CO offers a means of coordinating separate analyses, even leveraging discipline-specific optimization techniques. Relative to other decomposition-based methods, CO provides the disciplinary subspaces with an unusually high level of autonomy. This enhances their ability to independently make design decisions pertinent primarily to their discipline.

Collaborative optimization has been successfully applied to a variety of mathematical test problems and practical engineering design problems. For example, it has been used for the conceptual design of launch vehicles [5, 6], high speed civil transports [7], unmanned aerial vehicles [2], and aircraft family design [8]. However, it also suffers from several challenges, as documented by Alexandrov [9, 10], DeMiguel [11], and others. For example, the system level Jacobian is singular at the solution. In addition, the Lagrange multipliers in the subspace problems are either zero or converge to zero as  $\mathbf{z}$  converges to the optimum ( $\mathbf{z}^*$ ). This paper introduces a new version of CO called Enhanced Collaborative Optimization (ECO). It represents an improvement over CO in two key areas: (1) it eliminates most of the numerical difficulties cited by Alexandrov and DeMiguel, and (2) it provides disciplinary teams with greater influence over the optimization process.

Computational efficiency is improved through two sources. First, efficiency is improved by providing each subspace with direct information regarding design constraints in all other subspaces. This prevents subspaces from engaging in a tug-of-war struggle over appropriate choices for shared design variables. Fortuitously, it can be achieved with no increase in system level dimensionality. Second, efficiency is improved by eliminating the numerical challenges inherent in the CO architecture. For example, the system level problem is now unconstrained, eliminating the potential for a singular Jacobian. In addition, the subspace objective has been augmented with additional terms, as discussed in the next section. As a result, the subspace Lagrange multipliers no longer converge to zero at the solution. This combination of improved communication and improved numerical properties has yielded computational savings of nearly one order of magnitude for a suite of test cases [12].

The second improvement provided by ECO is enhanced subspace design authority. One of the strengths of the original CO method is that subspaces have exclusive control over local design decisions (i.e., subspace-specific variables). However, the subspace objective focuses exclusively on satisfying compatibility rather than directly reducing the global objective. In contrast, in the design of a complex aerospace system, the aerodynamics group would expect to work toward minimizing drag rather than simply seeking to match a set of design targets. So, it seems preferable to enable the subspaces to work directly on relevant portions of the global objective. This idea has been incorporated into ECO, as illustrated in this paper.

## 2 Description of the Method

This section provides an introduction to enhanced collaborative optimization (ECO) and a brief review of analytical target cascading (ATC).

### 2.1 Enhanced Collaborative Optimization

ECO decomposes the design problem into two levels. The upper level is termed the “system level,” and its responsibility is to coordinate the optimization process. The lower level is made up of “subspaces,” with one subspace for each relevant engineering discipline. The system level is an unconstrained minimization problem. The objective is to ensure that all subspaces use the same values of shared variables while satisfying their local constraints. Note that the global objective (i.e., the overarching design goal) is not present in the system level objective. The system level’s entire goal is to achieve compatibility between subspaces. The following subsections introduce the two levels of the decomposition. Both the system level and subspace problems are written in compact notation to emphasize the essential features of the method. An unabridged version can be found in related references [12, 13].

The ECO solution process is composed of three main steps: (1) construct constraint models, (2) solve the subspace optimization problem, and (3) solve the system level optimization problem. These steps are described in additional detail below. This three-step process is repeated until compatibility is achieved.

1. Construct constraint models: The system level sends targets ( $\mathbf{z}$ ) to constraint modeling subroutines. These constraint modelers construct linear models of all subspace constraints, and return the corresponding coefficients to the system level.
2. Solve the subspace optimization problem: The system level sends targets ( $\mathbf{z}$ ) and constraint model coefficients to the subspaces. These targets and coefficients are treated as parameters in the subspaces. The subspaces solve their local optimization problems and return target responses ( $\mathbf{x}_s^*$ ) to the system level.
3. Solve the system level optimization problem: The system level treats the target responses ( $\mathbf{x}_s$ ) as parameters while solving its optimization problem. (Note that this differs from the original version of CO, where each iteration at the system level required a complete subspace optimization.) The solution to the system level problem is a set of new targets.

**2.1.1 ECO - System Level** The standard version of collaborative optimization (CO) provides a significant degree of independence for each disciplinary subspace. This enables disciplinary experts to run their own codes using discipline-preferred optimization techniques. However, each subspace has very limited knowledge of the actions and preferences of the other subspaces. Information is only shared indirectly through the system level targets. As a result, the system level must retain responsibility for selecting the shared variables. In contrast, ECO provides each subspace with a direct understanding of the other subspaces' preferences (i.e., constraints). This enables the transfer of most of the system level decision-making process to the individual subspaces. (The subspaces direct the system level optimization process through target responses.) The system level coordination task is now limited to providing dynamic "move limits," which prevent the subspaces from taking large steps in the wrong direction based on limited (i.e., linearly approximated) information from the other subspaces. The following is a compact mathematical description of the system level.

$$\min_{\mathbf{z}} J_{sys} = \sum_i \|\mathbf{z} - \mathbf{x}^*\|_2^2 \quad (1)$$

subject to No constraints

where  $\mathbf{z}$  are system level targets for shared variables

$\mathbf{x}^*$  are subspaces' attempts to match system targets,

subject to local constraints

Note that, since the system level problem is unconstrained, the system level optimum is simply the average of the target responses returned from the subspaces.

**2.1.2 ECO - Subspace Level** The subspaces are responsible for most of the design decisions. Their objective function includes three components: a quadratic model of the global objective, a quadratic measure of compatibility, and a set of slack variables. Their constraint set includes local constraints and models of constraints from other subspaces. The subspace receives targets ( $\mathbf{z}$ ) and constraint model coefficients ( $\partial g^{(j)} / \partial x_s$ ) from the system level, which are treated as parameters. The subspace returns target responses ( $\mathbf{x}_s^*$ ). Note that each subspace (as illustrated by the  $i^{th}$  subspace) requires models of the constraints from all other subspaces. Though the original constraints are typically a function of both local and shared variables, the constraint models used in subspace  $i$  are functions only of the shared variables in subspace  $i$ . The constraint models are described in the next section. The inclusion of constraint models in each subspace will increase the size of the constraint set (relative to CO or ATC). In specific, the size of the constraint set within each subspace will be equivalent to the total number of constraints in the original (integrated) problem. However, this is not expected to significantly increase subspace computational effort since the inclusion of constraint models does not affect the number of degrees of freedom. In addition, many optimizers use special techniques to efficiently handle linear constraints.

The  $i^{th}$  subspace is defined as follows.

$$\min_{\bar{\mathbf{x}}=[\mathbf{x}_s, \mathbf{x}_L, \mathbf{s}]} J_i = \tilde{F} + \lambda_C \|\mathbf{x}_s - \mathbf{z}\|_2^2 + \lambda_F \sum \mathbf{s} \quad (2)$$

$$\begin{aligned} \text{subject to } & \mathbf{g}^{(i)}(\mathbf{x}_s, \mathbf{x}_L) \geq 0 \\ & \tilde{\mathbf{g}}^{(j)}(\mathbf{x}_s) + \mathbf{s}^{(j)} \geq 0, \quad j = 1..n, \quad j \neq i \\ & \mathbf{s} \geq 0 \end{aligned}$$

where  $\mathbf{x}_s$  are shared variables

(i.e., variables relevant to multiple subspaces)

$\mathbf{x}_L$  are local variables

(i.e., variables relevant only to  $i^{th}$  subspace)

$\mathbf{s}$  are slack variables,

which ensure a feasible subspace problem

$\lambda_C$  is a compatibility penalty parameter

$\lambda_F$  is a feasibility penalty parameter

$\tilde{F}$  is a quadratic model of the global objective

$\mathbf{g}^{(i)}$  are local constraints in subspace  $i$

$\tilde{\mathbf{g}}^{(j)}$  are linear models of constraints in subspace  $j$

The subspace objective is a combination of three terms: (1) a

quadratic model of the global objective, (2) a compatibility term, and (3) a constraint violation term. Terms (2) and (3) are accompanied by penalty parameters,  $\lambda_C$  and  $\lambda_F$ . The following is a brief description of the rationale for selecting these two parameters.  $\lambda_C$  determines the compromise between exploration (small  $\lambda_C$ ) and exploitation (large  $\lambda_C$ ). While its selection impacts the computational efficiency of this CO variant, convergence should be obtained for a reasonable range of values. The optimal choice depends on the degree of constraint non-linearity. For the simple test cases in this paper, any positive value of  $\lambda_C$  yields convergence.  $\lambda_F$  determines the emphasis placed on the constraint model(s). As long as  $\lambda_F$  is larger than the largest Lagrange multiplier, the constraint models should be satisfied to the extent that the subspace has sufficient degrees of freedom to do so.

The compatibility term warrants an additional note. Unlike the standard version of CO, the compatibility term in the subspace objective does NOT ensure compatibility. As with any quadratic penalty function, it will only be precisely satisfied in the limit as  $\lambda_C \rightarrow \infty$ . Rather, it acts as a dynamic “move limit,” guiding the optimization process. This “guide” is needed since each subspace has only limited knowledge of the other subspaces’ constraints. Without the “guide,” this limited knowledge might be wrongly exploited.

**2.1.3 ECO - Constraint Modeling** Consider a model of the  $k_{th}$  inequality constraint in subspace  $i$ . To ease the notation, we’ll drop the superscript ( $i$ ).

$$\tilde{g}_k = g_k|_{\mathbf{z}, \mathbf{x}_L^*} + \frac{d\mathbf{g}_k}{d\mathbf{x}_s}|_{\mathbf{z}, \mathbf{x}_L^*} (\mathbf{x}_s - \mathbf{z}) \quad (3)$$

$$\text{where } \frac{dg}{dx_s}|_{x_s=z, x_L^*} = \left( \frac{\partial g}{\partial x_s} \right) + \left( \frac{\partial g}{\partial x_L} \right) \left( \frac{dx_L^*}{dx_s} \right)$$

As written in Equation 3, the constraint models are functions only of the shared variables,  $\mathbf{x}_s$ . This is important, since variables relevant only to the  $i^{th}$  subspace should remain local to it in order to preserve the low dimensionality of the system level problem. In other words, the goal is to *indirectly* capture the effect of the local variables without sharing them outside of the subspace. At first glance, Equation 3 may seem to violate the chain rule since both  $\mathbf{x}_s$  and  $\mathbf{x}_L$  are independent variables in the subspace optimization problem. To circumvent this problem, consider treating the shared variables ( $\mathbf{x}_s$ ) in the  $i^{th}$  subspace problem as parameters with values,  $\mathbf{z}$ . In this case, the subspace problem (Equation 2) reduces to Equation 4. (The slack variables are added simply to ensure that all constraints can be satisfied.) All of the terms in Equation 3 now have physical meaning. In particular,  $\partial x_L / \partial x_s$  takes into account the impact of  $\mathbf{x}_L$  on all constraints and selects the value that minimizes cumulative con-

straint violation for a given change in one of the shared variables. Equation 4 is referred to as a constraint violation minimization (CVM) problem.

$$\begin{aligned} \min_{\bar{\mathbf{x}}=[\mathbf{x}_L, \mathbf{s}]} J_{i_{SVM}} &= \sum s & (4) \\ \text{subject to } \mathbf{g}(\mathbf{x}_s, \mathbf{x}_L) + \mathbf{s} &\geq 0 \\ \mathbf{s} &\geq 0 \\ \text{where } \mathbf{x}_s &\text{ are parameters with values, } \mathbf{x}_s = \mathbf{z} \\ \mathbf{g}(\mathbf{x}_s, \mathbf{x}_L) &\text{ are local constraints} \end{aligned}$$

The solution to this optimization problem yields  $\mathbf{x}_L^*$ , and additional information can be obtained from post-optimality sensitivity analysis [14]. Models of  $\mathbf{g}$  are constructed, as follows.

$$\left( \frac{dg}{dx_s} \right) = \left( \frac{\partial g}{\partial x_s} \right) + \left( \frac{\partial g}{\partial x_L} \right) \left( \frac{dx_L^*}{dx_s} \right) \quad (3)$$

where  $\left( \frac{\partial g}{\partial x_s} \right)$  is obtained via differentiating constraints

$\left( \frac{\partial g}{\partial x_L} \right)$  is obtained via differentiating constraints

$\left( \frac{dx_L^*}{dx_s} \right)$  is obtained from 2nd-order

post-optimality sensitivity analysis

The last term requires solution of the following set of linear equations, where the  $\mathbf{x}_s$  are treated as parameters with values of  $\mathbf{z}$ , and the set of independent variables is  $\mathbf{x} = [\mathbf{x}_L, \mathbf{s}]$ .

$$\begin{bmatrix} \nabla_{\mathbf{x}}^2 \mathcal{L}^* & -(A^*)^T \\ A^* & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial x_s} \\ \frac{\partial \lambda}{\partial x_s} \end{bmatrix} = \begin{bmatrix} -\nabla_{x_s x}^2 \mathcal{L}^* \\ -\frac{\partial g}{\partial x_s} \end{bmatrix} \quad (5)$$

where  $\mathcal{L} = f - \lambda g$

$A^*$  is the Jacobian

$g$  is the vector of constraints

For problems of moderate size, the components in Equation 5 are relatively inexpensive to compute. (See reference 13 for additional details.) Effort is currently focused on several alternative approaches to further reduce the computational cost of constructing the constraint models. With the current formulation, the computational cost to construct a set of constraint models is approximately equivalent to the cost of solving the subspace problem.

The preceding paragraphs have focused on developing models of the subspace constraints that can be shared with other subspaces. Since local analyses also effect the solution of the subspace optimization problem, it is important to model these analyses in other subspaces. This is accomplished using the same process outlined for constraints [13].

**2.1.4 ECO - Objective Modeling** The subspace objective function includes a model of the “global objective” (i.e., the overarching design goal such as minimizing cost or maximizing performance). Similar to the constraint models, the objective model is a function only of the shared variables. To construct the model, local variables (if present) are treated as parameters with values ( $x_L^*$ ) given by the solution to Equation 4. For simple problems such as those described in this paper, the objective function can be constructed from analytic derivatives. For more general design problems, finite differencing is required. The information required to construct this model is typically available as part of the constraint model construction process. Standard techniques for multidimensional quadratic curve fitting are used [13].

## 2.2 Analytical Target Cascading

ATC was formalized in 2000 to address needs in the automotive industry [15]. It was originally developed for object-based decomposition, translating top-level product targets into detailed design specifications. Since then it has been updated to broaden applicability and significantly improve computational efficiency [16]. While ATC can be applied to problems that are decomposed into more than two levels, multidisciplinary design problems are typically bi-level. A lower level “subproblem” is assigned to each relevant discipline (i.e., structures, aerodynamics, dynamics & control, etc.). The upper level problem coordinates the process, ensuring that compatibility is achieved among all disciplines.

ATC uses the same template for all levels of the decomposition, as illustrated by the  $i^{th}$  subproblem shown in Equation 6. The objective function has two components: a local objective (if it exists) and a compatibility function. The compatibility function,  $\pi$ , penalizes nonzero values in the deviation vector  $\mathbf{c}$ . The deviation vector quantifies the difference between shared quantities computed locally,  $\mathbf{z}_i$ , and the corresponding shared quantities computed by other subproblems,  $\hat{\mathbf{z}}_i$ . Shared quantities for element  $i$  consist of shared variables ( $\mathbf{x}_{si}$ ), and input and output linking variables ( $\mathbf{y}_{ij}$  and  $\mathbf{y}_{ji}$ ). (ATC’s “shared” and “linking” variables are collectively referred to as “shared” variables in ECO.) The components of  $\hat{\mathbf{z}}_i$  are fixed parameters during the optimization of subproblem  $i$ . Each “subproblem” optimization is subject to local constraints, which can be functions of local

variables and of shared and linking variables.

$$\begin{aligned} \min_{\mathbf{x}_i, \mathbf{y}_{ij}, \mathbf{x}_{l_i}} \quad & f(\mathbf{x}_i, \mathbf{y}_{ij}, \mathbf{x}_{l_i}) + \pi(\mathbf{c}) \\ \text{s.t.} \quad & \mathbf{g}_i(\mathbf{x}_i, \mathbf{y}_{ij}, \mathbf{x}_{l_i}) \geq \mathbf{0} \\ \text{where} \quad & \pi(\mathbf{c}) = \mathbf{v}^T \mathbf{c} + \|\mathbf{w} \circ \mathbf{c}\|_2^2 \\ & v^{(k+1)} = v^{(k)} + 2w^{(k)} \circ w^{(k)} \circ c^{(k)}; \\ & w^{(k+1)} = \beta w^{(k)} \\ & \mathbf{c} = \mathbf{z}_i - \hat{\mathbf{z}}_i \end{aligned} \quad (6)$$

Since each optimization problem is decoupled, all of the subproblems at a particular level can be solved in parallel. A popular ATC coordination strategy is to solve the top level problem (with initial guesses for top-level targets), use the results to update the target values for the next level down, solve the problems in the second level, and so on until the bottom level is reached. This large outer loop is repeated until all of the deviation vector values stop changing. Efficient penalty function methods can speed convergence, and have been shown to produce convergence in as few as 3 outer loop iterations [16, 17].

## 3 Illustrative Examples

Both CO and ATC have been used to solve a number of test cases and practical design problems [8, 12, 15]. This section explores the application of ECO and ATC to two analytic test cases that have often been used to evaluate decomposition-based methods. For both test cases, sufficient problem formulation details are provided to enable the reader to compare and contrast the salient features of ECO and ATC. Both methods achieve rapid, accurate solutions, as illustrated by the accompanying results.

### 3.1 Rosenbrock Problem

The Rosenbrock function is a classic function often used as a test case for optimization algorithms. This problem is of particular interest for ECO. One of the challenges with the original formulation of collaborative optimization is that subspace minimizers are often non-unique. This implies that the system level compatibility constraints are, in fact, set-valued functions. This trait was illustrating by Braun [6] using the Rosenbrock problem. One of the beneficial features of ECO is that, by adding a model of the global objective to each subspace objective function, one is virtually assured that the subspace minimizer is unique.

In order to create a problem that is better-suited to decomposition-based methods, the Rosenbrock problem is modified by adding constraints and implementing a change-of-variables [12]. This yields the integrated (single level) problem

shown in Equation 7.

$$\begin{aligned}
\min_{z_1, z_2, z_3, z_4} \quad & f_{global} = 100z_2^2 + z_3^2 \\
\text{subject to} \quad & h_1 = z_2 - (z_4 - z_1^2) = 0 \\
& h_2 = z_3 - (1 - z_1) = 0 \\
& g_1 = 0.575 - \sqrt{z_1^2 + z_4^2} \geq 0 \\
& z_1 \geq 0.01
\end{aligned} \tag{7}$$

The bound on  $z_1$  is added to ensure that the problem has a unique minima. (A local minima exists at  $z = [-0.5101, 0.0027, 0.2654]$ .) The solution to the Rosenbrock problem is shown in Table 1.

Consider the solution of the Rosenbrock problem using ECO. The problem can be decomposed into a bi-level structure with a system level problem and two subspaces. The system level contains no local constraints, as shown in Equation 8. So, its solution is simply an average of target responses from the subspaces.

$$\begin{aligned}
\min_{z_1, z_2, z_3} \quad & J_{sys} = (z_1 - x_1^{(1)})^2 + (z_2 - x_2^{(1)})^2 \\
& + (z_1 - x_1^{(2)})^2 + (z_2 - x_2^{(2)})^2 + (z_3 - x_3^{(2)})^2 \\
\text{subject to} \quad & \text{No Constraints}
\end{aligned} \tag{8}$$

The subspaces retain most of the responsibility for guiding the optimization process, seeking to minimize the global objective ( $f_{global}$ ) while ensuring their own compatibility with the other subspace. This high level of control is enabled by modeling the global objective within the subspace objective, and by modeling the effect of constraints from other subspaces. This is illustrated for subspaces one and two in Equations 9 and 10, respectively. As noted in the ECO method description, a wide range of penalty parameter values yield convergence. For the Rosenbrock problem, the following values were used:  $\lambda_C = 0.1, \lambda_F = 5.0$ .

$$\begin{aligned}
\min \quad & J_{ss1} = [100x_2^2] + \lambda_C [(x_1 - z_1)^2 + (x_2 - z_2)^2] \\
& + \lambda_F [s_1 + e_1] \\
\text{w.r.t. } \mathbf{x} = \quad & [x_1, x_2, x_L, s_1, e_1] \\
\text{s.t. } \quad & g_1 = 0.575 - (x_1^2 + x_L^2)^{1/2} \geq 0 \\
& h_1 = x_2 - (x_L - x_1^2) = 0 \\
& \tilde{h}_2 = h_2(z) + \left(\frac{dh_2}{dx_1}\right)(x_1 - z_1) + \left(\frac{dh_2}{dx_2}\right)(x_2 - z_2) \\
& + s_1 - e_1 = 0 \\
& s_1, e_1 \geq 0
\end{aligned} \tag{9}$$

$$\begin{aligned}
\min \quad & J_{ss2} = [100x_2^2 + x_3^2] + \lambda_F [s_1 + s_2 + e_1] \\
& + \lambda_C [(x_1 - z_1)^2 + (x_2 - z_2)^2 + (x_3 - z_3)^2] \\
\text{w.r.t. } \mathbf{x} = \quad & [x_1, x_2, x_3, s_1, s_2, e_1] \\
\text{s.t. } \quad & h_2 = x_3 - (1 - x_1) = 0 \\
& \tilde{h}_1 = h_1(z) + \left(\frac{dh_1}{dx_1}\right)(x_1 - z_1) + \left(\frac{dh_1}{dx_2}\right)(x_2 - z_2) \\
& + s_1 - e_1 = 0 \\
& \tilde{g}_1 = g_1(z) + \left(\frac{dg_1}{dx_1}\right)(x_1 - z_1) + \left(\frac{dg_1}{dx_2}\right)(x_2 - z_2) \\
& + s_2 \geq 0 \\
& s_1, s_2, e_1 \geq 0
\end{aligned} \tag{10}$$

The subroutine for subspace 1 constraint modeling is illustrated in Equation 11. The subspace 2 constraint modeling subroutine is trivial since subspace 2 has no local variables. It simply computes the value of the local constraints and their partial derivatives, evaluated at  $z$ . These values are returned to the system level for use in constructing linear models of the subspace 2 constraints.

$$\begin{aligned}
\min \quad & J_1 = [s_1 + s_2 + e_1] \\
\text{w.r.t. } \mathbf{x} = \quad & [x_L, s_1, s_2, e_1] \\
\text{subject to} \quad & g_1 = 0.575 - (z_1^2 + x_L^2)^{1/2} + s_1 \geq 0 \\
& h_1 = z_2 - (x_L - z_1^2) + s_2 - e_1 = 0 \\
& s_1, s_2, e_1 \geq 0
\end{aligned} \tag{11}$$

As a brief aside, note that when the global objective is removed from the system level and divided among the subspaces, the set of shared variables can often be reduced. For the Rosenbrock problem, the set of shared variables can (and probably should) be reduced from three ( $x_1, x_2$ , and  $x_3$ ) to one ( $x_1$ ). As a first step, the Rosenbrock problem was successfully solved with three shared variables.

Consider the solution of the Rosenbrock problem using ATC. Both ATC and ECO use the same problem decomposition, yielding a system level problem and two subproblems. The system level problem is shown in Equation 12. Note that, similar to ECO, the system level problem is unconstrained.

$$\begin{aligned}
\min_{z_1, z_2, z_3} \quad & J_{sys} = [100x_2^2 + x_3^2] + \pi(c) \\
\text{subject to} \quad & \text{No Constraints}
\end{aligned} \tag{12}$$

The ATC subproblems are shown in Equation 13. Briefly compare them with the ECO subspaces. For this problem, the ATC subproblem objective consists entirely of a compatibility

function. In contrast, the ECO subspace objectives contain three terms: (1) a model of the global objective, (2) a compatibility term, and (3) a feasibility term. In ATC, compatibility is driven exclusively by the compatibility function. In ECO, compatibility is actually enforced through the feasibility term via linear models of the constraints from all other subspaces. In ECO, the compatibility term acts primarily as a soft move-limit, ensuring that the linear constraint models from other subspaces remain relatively accurate.

$$\begin{aligned}
\min \quad & J_{ss1} = \pi(\mathbf{c}) \\
\text{w.r.t. } \quad & \mathbf{x} = [x_1, x_2, x_L] \\
\text{subject to } \quad & g_1 = 0.575 - (x_1^2 + x_L^2)^{1/2} \geq 0 \\
& h_1 = x_2 - (x_L - x_1^2) = 0
\end{aligned} \tag{13}$$

$$\begin{aligned}
\min \quad & J_{ss2} = \pi(\mathbf{c}) \\
\text{w.r.t. } \quad & \mathbf{x} = [x_1, x_2, x_3] \\
\text{s.t. } \quad & h_2 = x_3 - (1 - x_1) = 0
\end{aligned}$$

The solutions via ATC and ECO are shown in Table 1. Note that all three methods (integrated, ECO, and ATC) find the global minima with relatively few function evaluations. The number of “top level iterations” is listed to provide the reader with a sense of the relative computational efficiency of the methods. While only a single case is shown, the results are representative of those for a wide range of starting points. In addition, some effort has been made to select penalty parameters for all methods that yield the best possible rates of convergence. In terms of system (top) level iterations, ECO appears more efficient than ATC. However, one must also consider the computational effort required to construct the constraint models. The net result is that both methods require approximately the same amount of computational effort. The more important message is that ATC and ECO work well for the example problems investigated in this paper, while taking distinctly different solution approaches.

### 3.2 A Geometric Programming Problem

This section explores the application of ECO and ATC to a 14 variable geometric programming problem [15]. Geometric programming problems with posynomials are known to have a unique, globally optimal solution [18]. The example in this section has a quadratic object, 14 variables, 6 inequality constraints, and 4 equality constraints, along with non negativity constraints on the variables, as shown in Equation 14. When  $x_1, x_2, x_3$  and  $x_6$  are viewed as analysis responses, the equality constraints can

	Integrated	ECO	ATC
$x_1$	0.5124	0.5126	0.5124
$x_2$	$-1.6E^{-3}$	$-2.3E^{-3}$	$-1.6E^{-3}$
$x_3$	0.4876	0.4874	0.4874
$x_4$	0.2609	0.2605	0.2609
Objective	0.2380	0.2380	0.2378
Top Level Iter	18	16	21

also be viewed as analysis models. The integrated (single-level) problem formulation is shown in Equation 14. The resulting solution is shown in Table 2.

$$\begin{aligned}
\min_{\bar{\mathbf{x}}=[x_3, x_4, \dots, x_{14}]} \quad & f = x_1^2 + x_2^2 \\
\text{Inequality Constraints} \quad & g_1 = 1.0 - \frac{(x_3^{-2} + x_4^2)}{x_5^2} \geq 0 \\
& g_2 = 1.0 - \frac{(x_5^2 + x_6^{-2})}{x_7^2} \geq 0 \\
& g_3 = 1.0 - \frac{(x_8^2 + x_9^2)}{x_{11}^2} \geq 0 \\
& g_4 = 1.0 - \frac{(x_8^{-2} + x_{10}^2)}{x_{11}^2} \geq 0 \\
& g_5 = 1.0 - \frac{(x_{11}^2 + x_{12}^{-2})}{x_{13}^2} \geq 0 \\
& g_6 = 1.0 - \frac{(x_{11}^2 + x_{12}^2)}{x_{14}^2} \geq 0 \\
\text{Equality Constraints} \quad & h_1 : x_1^2 = x_3^2 + x_4^{-2} + x_5^2 \\
& h_2 : x_2^2 = x_5^2 + x_6^2 + x_7^2 \\
& h_3 : x_3^2 = x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2 \\
& h_4 : x_6^2 = x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 \\
\text{Bounds} \quad & x_3, x_4, \dots, x_{14} \geq 0
\end{aligned} \tag{14}$$

To solve the G.P. problem using ECO, it is most convenient to decompose it into three subspaces. While a two-subspace decomposition is also possible, this decomposition eliminates all local constraints from the system level. The resulting system level optimization problem is reduced to a simple average of sub-

space target responses, as shown in Equation 15.

$$\begin{aligned} \min_{\bar{\mathbf{z}}=[z_3, z_6, z_{11}]} J_{sys} &= \left[ \left( z_3 - x_3^{(1)} \right)^2 + \left( z_{11} - x_{11}^{(1)} \right)^2 \right] \quad (15) \\ &+ \left[ \left( z_6 - x_6^{(2)} \right)^2 + \left( z_{11} - x_{11}^{(2)} \right)^2 \right] \\ &+ \left[ \left( z_3 - x_3^{(3)} \right)^2 + \left( z_6 - x_6^{(3)} \right)^2 \right] \\ \text{subject to} & \text{ None} \end{aligned}$$

Each subspace is responsible for selecting optimal values of local design variables and for satisfying local constraints. Each subspace also seeks to satisfy linear models of the constraints from all other subspaces. (This is in contrast to CO and ATC, where subspaces do not directly consider the impact of their local decisions on the constraints in other subspaces.) In the equations shown below, subspace one is fully described while subspaces two and three are provided in compact notation. As noted in the ECO method description, a wide range of penalty parameter values yield convergence. For this problem, the following values were used:  $\lambda_C = 0.1$ ,  $\lambda_F = 4.0$ .

Subspace 1 is defined as:

$$\begin{aligned} \min J_1 &= [x_3^2] + \lambda_C [(x_3 - z_3)^2 + (x_{11} - z_{11})^2] \\ &+ \lambda_F [e_1 + e_2 + e_5 + e_6 + e_{h_2} + s_{h_2}] \\ \text{w.r.t } \mathbf{x} &= [x_3, x_8, x_9, x_{10}, x_{11}, e_1, e_2, e_5, e_6, e_{h_2}, s_{h_2}] \\ \text{subject to } \tilde{g}_1 &= g_1(z) + (dg_1/dz_3)(x_3 - z_3) + e_1 \geq 0 \\ \tilde{g}_2 &= g_2(z) + (dg_2/dz_3)(x_3 - z_3) + e_2 \geq 0 \\ g_3 &\geq 0 \\ g_4 &\geq 0 \\ \tilde{g}_5 &= g_5(z) + (dg_5/dz_{11})(x_{11} - z_{11}) + e_5 \geq 0 \\ \tilde{g}_6 &= g_6(z) + (dg_6/dz_{11})(x_{11} - z_{11}) + e_6 \geq 0 \\ \tilde{h}_2 &= h_2(z) + (dh_2/dz_{11})(x_{11} - z_{11}) + e_{h_2} - s_{h_2} = 0 \\ \text{Bounds } \mathbf{x} &\geq 0 \\ \text{Analysis } h_1 &: x_3^2 - [x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2] = 0 \end{aligned}$$

Subspace 2 is defined as:

$$\begin{aligned} \min J_2 &= [x_6^2] + \lambda_C [(x_6 - z_6)^2 + (x_{11} - z_{11})^2] \\ &+ \lambda_F [e_1 + e_2 + e_3 + e_4 + e_{h_1} + s_{h_1}] \\ \text{w.r.t. } \mathbf{x} &= [x_6, x_{11}, x_{12}, x_{13}, x_{14}, e_1, e_2, e_3, e_4, e_{h_1}, s_{h_1}] \\ \text{subject to } g_5, g_6, \tilde{g}_1, \tilde{g}_2, \tilde{g}_3, \tilde{g}_4, \tilde{h}_1, \mathbf{x} &\geq 0 \\ \text{Analysis } h_2 &: x_6^2 - x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 = 0 \end{aligned}$$

Subspace 3 is defined as:

$$\begin{aligned} \min J_3 &= [x_1^2 + x_2^2] + \lambda_C [(x_3 - z_3)^2 + (x_6 - z_6)^2] \\ &+ \lambda_F [e_3 + e_4 + e_5 + e_6 + e_{h_1} + e_{h_2} + s_{h_1} + s_{h_2}] \\ \text{w.r.t. } \mathbf{x} &= [x_3, x_4, x_5, x_6, x_7, e_3, e_4, e_5, e_6, e_{h_1}, e_{h_2}, s_{h_1}, s_{h_2}] \\ \text{subject to } g_1, g_2, \tilde{g}_3, \tilde{g}_4, \tilde{g}_5, \tilde{g}_6, \tilde{h}_1, \tilde{h}_2, \mathbf{x} &\geq 0 \\ \text{Analysis } x_1^2 &= x_3^2 + x_4^{-2} + x_5^2 \\ x_2^2 &= x_5^2 + x_6^2 + x_7^2 \end{aligned}$$

The ECO process also requires a constraint modeling step. This is illustrated for subspace one (Equation 16). For sake of brevity, the remaining constraint modeling subroutines are not shown. However, they can be deduced from provided information.

$$\begin{aligned} \min J_1 &= e_1 + e_2 + e_{h_1} + s_{h_1} \quad (16) \\ \text{w.r.t } \mathbf{x} &= [x_8, x_9, x_{10}, e_1, e_2, e_{h_1}, s_{h_1}] \\ \text{Parameters } x_3 &= z_3, \quad x_{11} = z_{11} \\ \text{subject to } g_3 + e_1 &\geq 0 \\ g_4 + e_2 &\geq 0 \\ h_1 + e_{h_1} - s_{h_1} &= 0 \\ \text{Bounds } \mathbf{x} &\geq 0 \end{aligned}$$

The solution to the G.P. problem using ECO is provided in Table 2. Note that only a small number of system level iterations are required in order to achieve convergence. This is particularly important in a design environment where the product development schedule places a practical limit on the number of design iterations that can be completed before proceeding to the next phase of the design process.

To solve the G.P. problem using ATC, it is more convenient to decompose it into two subspaces. A three-subspace decomposition is also possible and would eliminate all local constraints from the system level. In such a case, an analytic solution to the system level problem is possible. However, in practice, an optimization algorithm is nearly always used to solve the system level problem (Equation 17). So, in contrast to ECO, ATC does not particularly benefit from eliminating all local constraints

from the system level.

$$\begin{aligned}
& \min f = z_1^2 + z_2^2 + \pi(\mathbf{c}) & (17) \\
& \text{w.r.t. } \mathbf{z} = [z_3, z_4, z_5, z_6, z_7, z_{11}] \\
& \text{subject to } g_1(\mathbf{z}) \geq 0 \\
& \quad g_2(\mathbf{z}) \geq 0 \\
& \text{Bounds } z_3, z_4, z_5, z_6, z_7, z_{11} \geq 0 \\
& \text{Analysis } z_1^2 = z_3^2 + z_4^{-2} + z_5^2 \\
& \quad z_2^2 = z_5^2 + z_6^2 + z_7^2
\end{aligned}$$

The ATC subproblems are shown below. Note that the sole objective is to minimize the compatibility function. This is in contrast to ECO where the objective function addresses compatibility *and* includes terms that model the global objective and constraints from other subspaces. Subspace 1 is defined as:

$$\begin{aligned}
& \min_{\bar{\mathbf{x}}=[x_8, x_9, x_{10}, x_{11}]} f = \pi(\mathbf{c}) \\
& \text{subject to } g_3(\mathbf{x}) \geq 0 \\
& \quad g_4(\mathbf{x}) \geq 0 \\
& \text{Bounds } x_8, x_9, x_{10}, x_{11} \geq 0 \\
& \text{Analysis } x_3^2 = x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2
\end{aligned}$$

Subspace 2 is defined as:

$$\begin{aligned}
& \min_{\bar{\mathbf{x}}=[x_{11}, x_{12}, x_{13}, x_{14}]} f = \pi(\mathbf{c}) \\
& \text{subject to } g_5(\mathbf{x}) \geq 0 \\
& \quad g_6(\mathbf{x}) \geq 0 \\
& \text{Bounds } x_{11}, x_{12}, x_{13}, x_{14} \geq 0 \\
& \text{Analysis } x_6^2 = x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2
\end{aligned}$$

Table 2 outlines the solution of the G.P. problem using ATC. Note that both ATC and ECO yield good results. Though not shown, both achieved excellent compatibility between subspaces. In terms of system (top) level iterations, ECO appears more efficient than ATC. However, when the computational effort required for constraint model construction is considered, the net result is that both methods require approximately the same amount of computational effort. Again, the more important message is that ATC and ECO work well for the example problems investigated in this paper, while taking distinctly different solution approaches.

Table 2. Geometric Programming Problem

	Integrated	ECO	ATC
$x_1$	2.84	2.84	2.85
$x_2$	3.09	3.09	3.07
$x_3$	2.36	2.37	2.38
$x_4$	0.76	0.76	0.76
$x_5$	0.87	0.87	0.87
$x_6$	2.81	2.81	2.80
$x_7$	0.94	0.94	0.94
$x_8$	0.97	0.97	0.98
$x_9$	0.87	0.85	0.84
$x_{10}$	0.80	0.79	0.78
$x_{11}$	1.30	1.29	1.29
$x_{12}$	0.84	0.84	0.84
$x_{13}$	1.76	1.76	1.75
$x_{14}$	1.55	1.54	1.54
Objective	17.59	17.62	17.59
Top Level Iter	43	14	25

## 4 Closing Remarks

This paper has provided an introduction to a new and improved version of collaborative optimization (CO). The key idea in this new approach (ECO) is to infuse the subspace optimization problems with additional information while maintaining the low dimensionality of the system level (coordination) problem. The new information includes linear models of all subspace constraints and a quadratic model of the global objective. In this paper, ECO is applied to two analytic test cases: the Rosenbrock function and a geometric programming problem. The problem formulations are compared and contrasted with ATC, a well-established decomposition-based method for solving hierarchical problems. Both ECO and ATC successfully locate the global minima in an efficient manner.

Recent work has focused on implementing ECO on more realistic engineering problems, including an aircraft family design problem. Current effort is focused on improving the computational efficiency of the constraint modeling process. Effort is also focused on developing robust methods for the selection (and potential updating) of penalty parameters.

## REFERENCES

- [1] "Current State of the Art on Multidisciplinary Optimization," *AIAA White Paper*, AIAA Technical Committee on Multidisciplinary Design Optimization, 1991.
- [2] Sobieski, I., *Multidisciplinary Design Using Collaborative Optimization*, Ph.D. thesis, Stanford University, August 1998.
- [3] Sabbagh, C., *21st Century Jet*, Scribners, 1996.
- [4] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," 1994, AIAA paper 94-4325.
- [5] Braun, R., Kroo, I., and Moore, A., "Use of the Collaborative Optimization Architecture for Launch Vehicle Design," 1996, AIAA paper 96-4018.
- [6] Braun, R., *Collaborative Optimization: An Architecture For Large-Scale Distributed Design*, Ph.D. thesis, Stanford University, April 1996.
- [7] Manning, V., *Large-Scale Design of Supersonic Aircraft Via Collaborative Optimization*, Ph.D. thesis, Stanford University, 1999.
- [8] Allison, J., Roth, B., Kokkolaras, M., Kroo, I., and Papalambros, P., "Aircraft Family Design Using Decomposition-Based Methods," *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA, September 6-8, 2006.
- [9] Alexandrov, N. and Lewis, R., "Analytical and Computational Aspects of Collaborative Optimization and Multidisciplinary Design," *AIAA Journal*, Vol. 40, No. 2, February 2002, pp. 301–309.
- [10] Alexandrov, N. M. and Lewis, R., *Engineering Design Optimization*, chap. Comparative Properties of Collaborative Optimization and Other Approaches to MDO, MCB Univ. Press, Bradford, England, U.K., 1999, pp. 39–46.
- [11] DeMiguel, A. and Murray, W., "An Analysis of Collaborative Optimization Methods," *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, Sept. 6-8, 2000.
- [12] Roth, B. and Kroo, I., "Enhanced Collaborative Optimization," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, BC, September 6-8, 2008.
- [13] Roth, B., *Aircraft Family Design Using Enhanced Collaborative Optimization*, Ph.D. thesis, Stanford University, June, 2008.
- [14] Sobieszczanski-Sobieski, J., Barthelemy, J., and Riley, K., "Sensitivity of Optimum Solutions of Problem Parameters," *AIAA Journal*, Vol. 20, No. 9, September 1982, pp. 1291–1299.
- [15] Kim, H., Michelena, N., Papalambros, P., and Jiang, T., "Target Cascading in Optimal System Design," *Journal of Mechanical Design*, Vol. 125, September 2003, pp. 474–480.
- [16] Tosserams, S., Etman, L., Papalambros, P., and Rooda, J., "Augmented Lagrangian Relaxation for Analytical Target Cascading," *6th World Congress on Structural and Multidisciplinary Optimization*, May 30-June 3, 2005.
- [17] Allison, J., Walsh, D., Kokkolaras, M., Papalambros, P., and Cartmell, M., "Analytical Target Cascading in Aircraft Design," *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2006-1325, Reno, Nevada, January 9-12, 2006.
- [18] Beightler, C. and Phillips, D., *Applied Geometric Programming*, John Wiley and Sons, Inc., New York, 1976.