DISTRIBUTED MULTIDISCIPLINARY DESIGN AND COLLABORATIVE OPTIMIZATION

I. Kroo
Stanford University, U.S.A.

1. Summary

These notes describe some recent ideas for distributed design and their application to large-scale aerospace systems. In this type of multidisciplinary optimization, design tasks are decomposed into domain-specific subproblems, and coordinated to achieve an optimal system. Focusing on collaborative optimization, one form of design decomposition, the notes detail the methods, summarize recent results, and suggest new variants of these approaches that improve performance.

2. Introduction

Initial applications of multidisciplinary optimization (MDO) involved the direct integration of multiple disciplinary analyses and an optimizer. For small problems, wiring together such a system is quite feasible and usually leads to a efficient, but sometimes hard to explain, procedure. As computational capabilities grew, engineers scaled this approach to larger problems and its limitations became apparent. The need for analysis and data management became better recognized and a second generation of MDO methods came into use. Distributed analysis systems could utilize multiple computers, increasing the practical scale of MDO problems (figure 1). Database management and modular analysis coordination improved efficiency and maintainability.

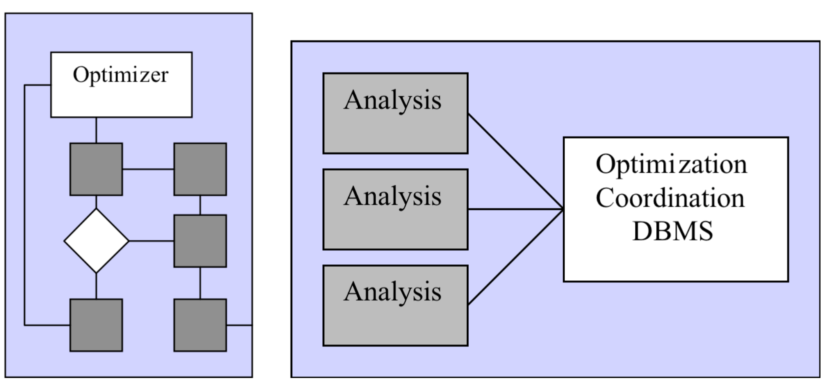

Figure 1. Initial architectures for multidisciplinary design involved integrated analysis and optimization. This was followed by application integration frameworks.

Despite these improvements, the reliance on a central optimizer as decision maker on all matters, is not a practical approach to enterprise-level system design. This deficiency has led to the development of what may be considered the third generation of MDO methods: strategies for distributed design optimization. This concept, illustrated in figure 2, involves decomposition of the design process itself into more manageable pieces.

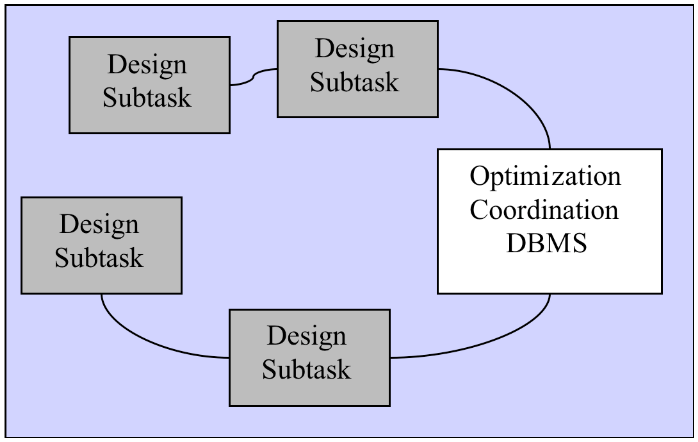

Figure 2. Distributed design architecture with subspace design problems and system-level coordination.

In fact, distributed design is the way any large-scale system is designed now. However, the approach to distributed design is often not well-planned. Sequential disciplinary designs and informal iteration can lead to designs that are sub-optimal. In many ad-hoc design procedures, individual design teams are assigned subsets of the design variables, parts of the analysis, and a local objective function that is only vaguely aligned with the overall goals of the system. Sequential choices by these design teams lead to a kind of non-cooperative game, which may reach an equilibrium that is not an optimum for the system.

Even when the goals of the subtasks appear aligned, problems may arise. Consider the problem of minimizing: $J = x^2 + y^2$ subject to the constraint: $g(x,y) = x - y - 2 > 0$. If the first group is responsible for the design variable y, and tries to minimize J for a given x, while the second group must vary x to minimize J subject to the constraint, $g>0$, the system will find the solution $y=0$, $x=2$. The correct solution $y=-1$, $x=1$ is not discovered because the decomposition introduces a feasible, but poor, equilibrium point.

An important current challenge for large scale MDO is the development of practical methods for distributed design optimization that are efficient and lead to good designs. The task of decomposing a problem whose components are strongly coupled is usually undertaken in an informal way, but may require more careful consideration.

The first step towards solution of a large-scale design problem involves decomposition of analyses that may be coupled in complex ways. Methods for decomposing and managing the analysis process formed the basis of much of the early work in MDO [1,2]. One method that has been very successful in multidisciplinary optimization with several analysis modules is termed optimizer-based decomposition (OBD). The concept is based on the analysis management structure shown in figure 3.

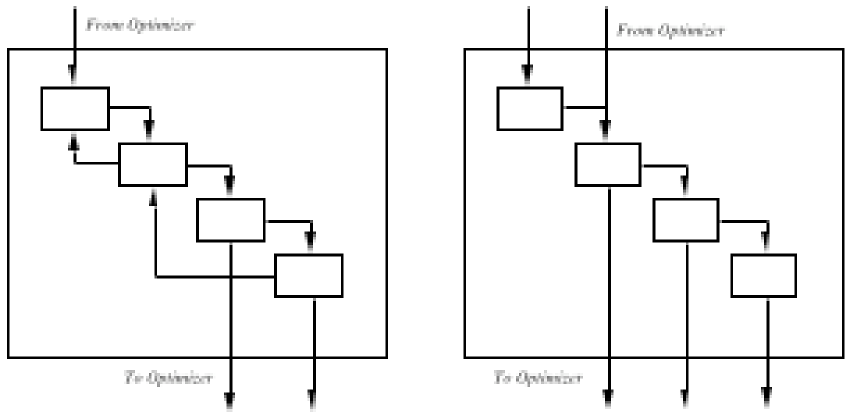

Figure 3. Removal of iteration loops in a series of analyses through the introduction of auxiliary design variables and compatibility constraints.

On the left side of figure 3, an optimizer is used to vary design parameters, with coupled analyses returning the objective function and constraint values. The sequential analyses involve some feedback loops when the output of a later analysis is required as input to the prior analysis. Fixed point iteration is used to converge the analysis loops, leading to issues with function precision and smoothness that are affected by the convergence criteria.

In an integrated optimization problem, one can remove feedback by introducing auxiliary design variables and compatibility constraints. The auxiliary variables, y', are new design variables that are "copies" of the computed values, y; they are prescribed by the optimizer rather than being input directly from another analysis module. The compatibility constraint, added to the optimization problem is that these values match:
y' = y, at the solution. This makes the optimization problem larger, but the analysis faster and smoother. This approach also resolves problems with poorly-convergent iteration loops (e.g. aeroelastic wing design near divergence), so that for some problems, the optimization is more efficient, despite the larger design space.

Although elimination of *feedback* with OBD is commonly used, the same idea may be used to eliminate *feedforward* and decompose the analysis into a series of parallel computations. This concept is shown in figure 4. Again, design variables are added to the optimization problem along with compatibility constraints. Although this type of decomposition generally increases the total number of function evaluations required for optimization, it may still be advantageous since the analyses may be run in parallel. With strong and high dimensionality coupling, the computational penalty for this approach may be excessive, but for problems that require only a few auxiliary variables, the simplicity of the decomposition may make it worthwhile.

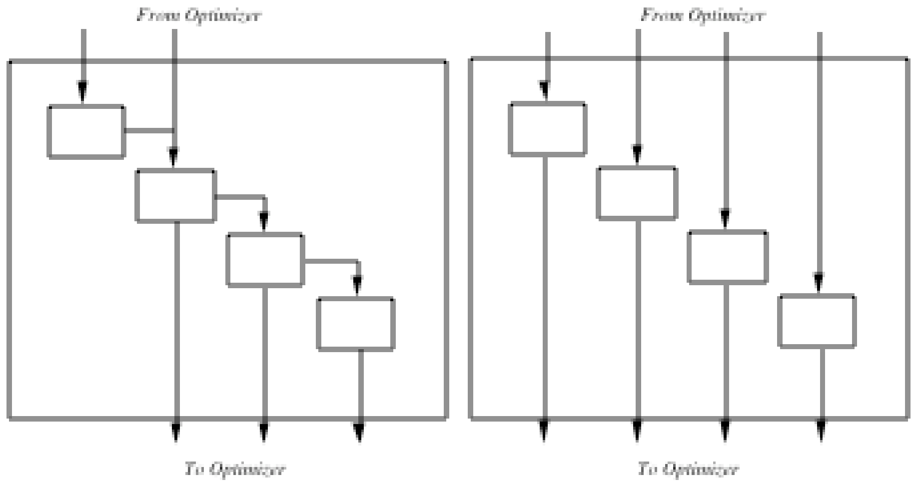

Figure 4. Removal of feed-forward in a series of analyses with auxiliary variables and additional constraints.

3. Approaches to Distributed Design

The concept for analysis decomposition described in the previous section may be applied when the analysis boxes in figure 4 represent design tasks. In this way the idea of optimizer-based decomposition, leads to a formal way of decomposing the design problem into parallel design tasks, coordinated by a system-level optimization algorithm. When the individual design tasks involve many local variables and the dimensionality of coupling is small, this process provides a very efficient method for distributed design.

Most methods for distributed design optimization are based on a related type of multi-level optimization process and have been studied for some time (see [3-10]). A few of these methods have been developed in some detail and applied to aeronautical design problems. Three of the more widely used strategies are described here.

3.1 Concurrent subspace optimization (CSSO)

Concurrent subspace optimization divides the design problem into several discipline-related subspaces, each of which shares some responsibility for satisfying constraints imposed on the system while trying to reduce a global objective (or objectives). A system level algorithm coordinates this process in different ways, depending on the implementation. Several versions of CSSO have been developed in recent years by researchers at NASA Langley, and Notre Dame [11]. A commercial framework based on this concept (SysOpt) has been developed and is described in a number of papers [12]. Figure 5 shows the basic concept.

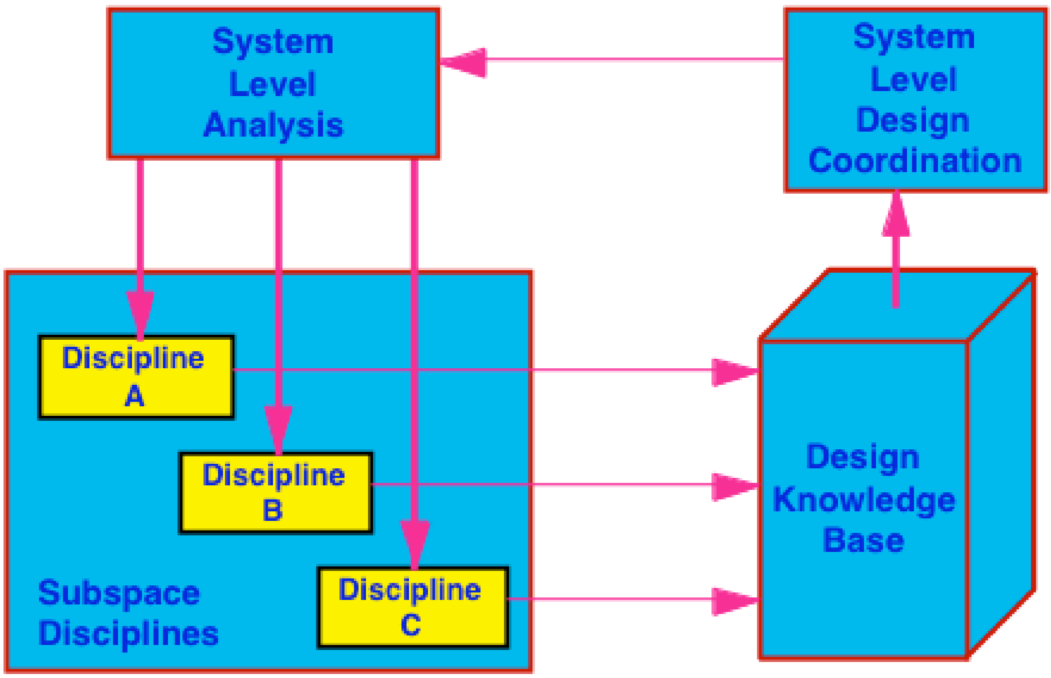

Figure 5. Concurrent Subspace Optimization involves shared responsibility for constraints, system level analysis and coordination, and a model of previous results in a design knowledge base.

One of the more interesting aspects of CSSO is the modeling of disciplinary analysis results in some form of design knowledge base. Neural networks and other response surfaces have been used effectively [13,14] for this purpose, although the complexity of the process, involving disciplinary optimization, system-level coordination and analysis, and subspace modeling, make implementation difficult and have limited its application.

3.2. Collaborative Optimization and BLISS

More recently, two closely-related architectures for distributed MDO have been developed and applied to aerospace design problems. Collaborative optimization (CO) [15] and bi-level integrated system synthesis (BLISS) [16] assign subsets of the system constraints to individual disciplinary designers that are most closely associate with these constraints. The basic decomposition is shown in figure 6.

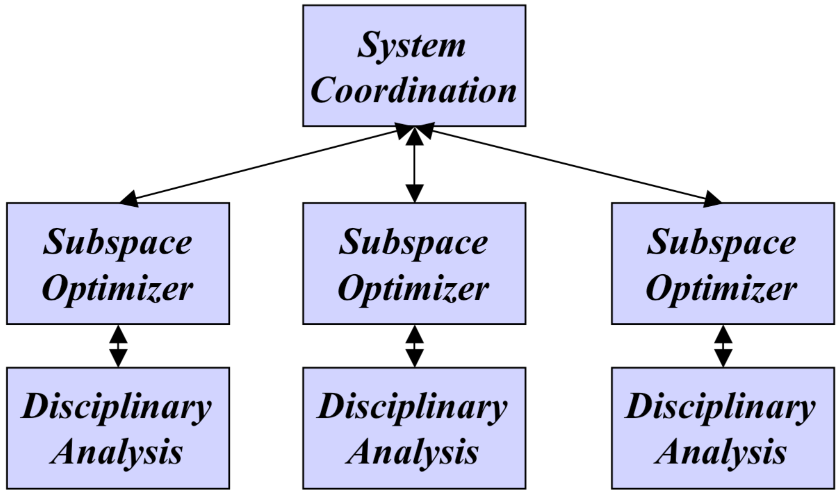

Figure 6. Basic structure of a collaborative optimization or BLISS problem.

Several variants of the BLISS and CO architectures have been proposed [17-20], but in each case subspace optimizers minimize a local objective that is related in some way to the system objective while satisfying constraints locally. Local degrees of freedom are used in this process, while global design variables are handled using a form of optimizer-based decomposition, with "copies" of the global variables used by the subspaces compatibility enforced at the system level.

Collaborative optimization treats shared variables as targets, whose selection is coordinated by the system-level optimizer. It creates more copies of interdisciplinary variables in an effort to ensure that local optimizers always have enough degrees of freedom to satisfy their local constraints.

The primary features of each of these architectures include the following:

1. Heterogeneous hardware or software may be used to solve the subspace optimization problems, employing optimization codes best suited for that domain.

2. The decomposition keeps domain-specific constraint information in the subproblem, rather than passing what might be thousands of constraints among many of the design teams.

3. The system leaves most of the design decisions (selection of local variables) to the disciplinary groups that understand the local problem.

This type of design decomposition makes the most sense for large problems (conventional fully-integrated, "all-at-once" optimization is best for small problems) that exhibit low dimensionality coupling (to reduce the number of auxiliary variables). It is especially useful when special methods exist for disciplinary design (e.g. adjoint methods for aerodynamic optimization or collocation methods for trajectory design) and is compatible with conventional organizational structures.

4. Collaborative Optimization

4.1. Basic Concepts

Collaborative optimization is described in more detail in [21-24], but consists basically of a two-level optimization architecture in which individual disciplinary teams are charged with satisfying local constraints. These subspace design teams are permitted to vary local parameters to accomplish that task. Since there may not exist sufficient local degrees of freedom to satisfy all of the constraints, subspaces are permitted to depart from the values of interdisciplinary parameters, established as targets by the system-level coordination method, although this departure is to be minimized. Thus, it is the job of the subspaces to satisfy constraints while working to define a design that everyone can agree on – hence the name collaborative optimization, or CO. The system is charged with adjusting the target values so that such agreement is possible while minimizing the system-level objective. The form of the subspace objectives and system level constraints may vary, but the basic idea is exemplified by the implementation in figure 7.

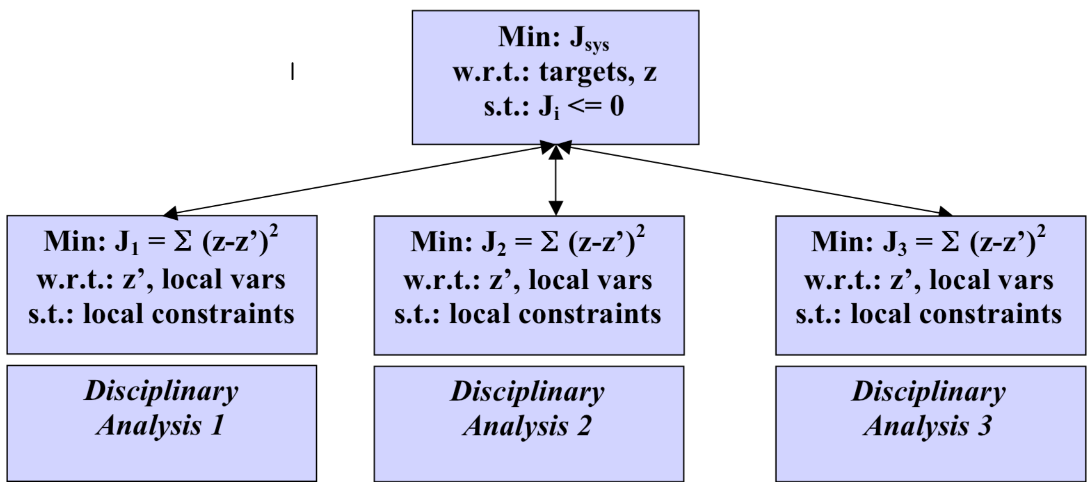

Figure 7. Basic structure of collaborative optimization architecture.

Figure 8 provides a graphical view of this process. The system chooses a set of targets, depicted by the point P. The subspaces then adjust their degrees of freedom in order to satisfy their own constraints while trying to match this target point as closely as possible. One may think of the subspaces as constrained to move along the lines shown, but connected to the target point with springs. The system tries to move the target in such a way that the subspace points coincide at the lowest possible objective value.

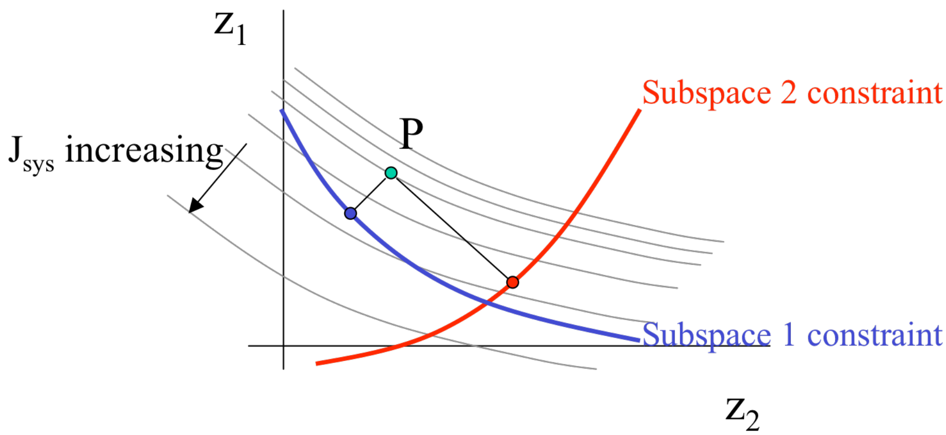

Figure 8. Graphical view of CO process. System picks target point (P); each subspace finds closest feasible point.

Several choices are possible for the forms of the subspace objectives and the system-level compatibility constraints. These have been investigated in a number of previous studies and it seems likely that improved selections will be identified [20]. Many of the fundamental characteristics of CO are apparent in its various forms, and most of the applications that have been studied have relied on simple L2 measures for the subspace objectives (i.e. the sum of the squared differences between the local values for a variable and the requested system targets). When the subspace objectives are used directly as the system level constraints, simple post-optimality sensitivities can be used to efficiently guide the system optimizer to an improved design [25]. Note that the use of the sum-of-squares corresponds to a hardening spring in the model of figure 8, meaning that as the subspace solutions get closer, the tension falls quickly. This sometimes causes difficulties as discussed in the subsequent section.

In collaborative optimization, unlike some multi-level optimization approaches, we must assure that each subspace is given sufficient degrees of freedom that it can always achieve a design that is feasible with respect to its local constraints. This is done by making copies of the system target variables and allowing the subspace to vary these copies. Although this increases the dimensionality of the subproblem, it avoids difficulties that have been encountered in some multi-level optimization methods when responsibilities for design variable selection and constraint satisfaction are distributed among levels as in the example of Section 2.

Details of the collaborative method have been developed by several researchers, primarily in the last ten years, although closely-related work is described in publications as early as 1977. Peterson et al. [26] describes a method in which subspaces are responsible for constraint satisfaction and CO-like targets are used. Thareja and Haftka [27] experimented with a very similar scheme with quadratic subspace objectives, encountering numerical difficulties, partly inherent in the formulation and partly associated with the optimizer. Sobieski, Riley, and Barthelemy [28,7] explored schemes in which the subspaces tried to satisfy a cumulative constraint. This is the alternative concept to collaborative optimization, in which interdisciplinary compatibility is maintained at each system-level iteration at the expense of disciplinary feasibility. It may be argued that this is an advantage, although in many cases a single infeasible design is no better than a set of not quite compatible designs. More recent algorithmic development of CO and CO-like methods are described in [29-31].

4.2. Algebraic Test Problems

In any architecture for distributed design it is important to verify that the process correctly finds the solutions to both simple and more complex problems. Many algebraic test problems have been considered in evaluating CO [20, 22, 32-34], A particularly simple problem was proposed in [34] and has been used to evaluate a range of decomposition methods. The simple two-discipline test problem is shown in figure 9 and consists of a quadratic global objective that depends on two shared variables. The problem is coupled with two linear constraints with adjustable coupling and that are distributed, one to each discipline. Clearly decomposition is not necessary for this problem and the problem is poorly suited to CO since all of the system level variables appear in all of the constraints, but it highlights how the problem may solved with CO.

$$\begin{aligned} \min_{X_1,X_2} F &= {X_1}^2 + {X_2}^2 \\ s.t.\ c_1 &= X_1 + \beta X_2 < 4 \\ c_2 &= \beta X_1 + X_2 > 2 \\ l_1 &< X_1 < u_1 \\ l_2 &< X_2 < u_2 \end{aligned}$$

Figure 9. Constrained quadratic test problem with adjustable coupling (Shankar) used to evaluate distributed optimization strategies.

The subproblems must introduce copies of the system variables: x_{11} and x_{12} are the local copies of X_1 in subspace 1 and 2 respectively.
Subspace 1 minimizes: $J_1 = (x_{11}-X_1)^2 + (x_{21} - X_2)^2$ with respect to x_{11} and x_{21} while satisfying $x_{11}+\beta x_{21}<4$.
Subspace 2 minimizes: $J = (x_{21}-X_1)^2 + (x_{22} - X_2)^2$ with respect to x_{21} and x_{22} while satisfying $\beta x_{11}+x_{21}>2$.
The system-level optimization problem is to minimize F subject to $J_1=J_2=0$.

In [22], this problem is solved using sequential quadratic programming (NPSOL) [35,36] at the system level. SQP may also be used at the subspace level, but in this simple problem, the subspace solutions are easily derived analytically. The decomposed version of the problem converged to the correct solution for every value of β and starting point, but required 14-19 system level iterations to converge, despite the fact that analytic gradients of the system-level constraints were available. This result is encouraging in that the system correctly converged to the solution and that the number of system level iterations is not terribly large, but one might have expected a more efficient system solution.

4.3. Implementation Details

Certain features of the architecture have created difficulties, especially for inexperienced developers. Experiments have suggested that the price that must be paid for the advantages of decomposition is a somewhat increased computational time. However, some studies have cited extremely large computational increments, which is usually due to implementation details. Several features of the method contribute to the ease by which one may arrive at a poor implementation. First, any multi-level distributed system is complex and with multiple optimizers and distributed databases, the initial implementation can be daunting. Secondly, the problem formulation and decomposition requires careful planning, a step that is not required for an integrated system. Finally, certain mathematical details can cause numerical difficulties for many existing optimization methods. In common with most multi-level schemes, the CO system-level problem may be sensitive to the selection of subspace optimization parameters such as feasibility or optimality tolerances. In some variants of CO, changes in the subspace active constraint set can lead to non-smooth behavior of the system-level constraints. Although decomposition may not introduce spurious local minima itself, the design topology is different from the integrated problem and may be more difficult to interpret. The choice of system constraint and subspace objective forms is critical to the system performance. This aspect of the method has received greater attention recently and warrants careful consideration as various versions of CO are developed.

The use of quadratic forms for the system level compatibility constraints means that near the solution, changes in system targets have little effect on the constraint values. Specifically, the gradient approaches zero, leading to difficulties for many optimizers, especially those that rely on linear approximations to these functions. This was observed in the early development of CO and can lead to slow convergence of the system near the solution as shown in figure 10 (from [37]) and noted in [27].

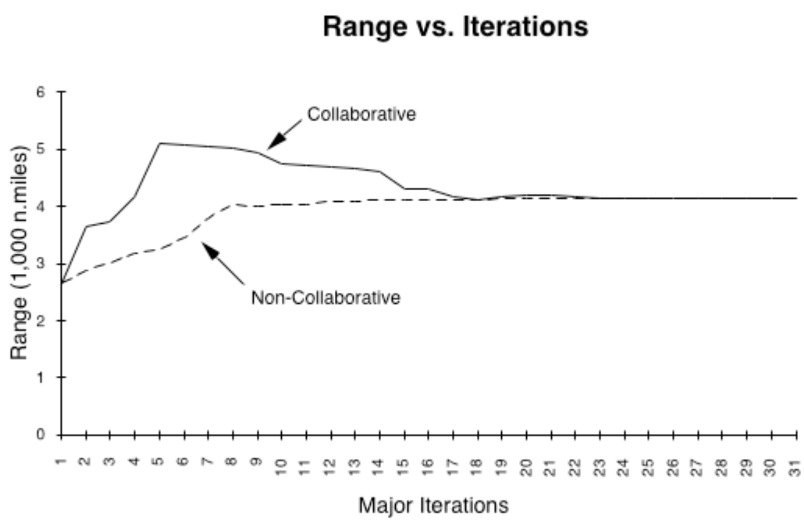

Figure 10. Convergence of the CO algorithm on an aircraft design problem. Since compatibility is enforced only at the solution, the objective remains optimistic until these constraints are satisfied.

The implications of the singular Jacobian are discussed in [32] using an SQP method at the system level. For a simple test case, convergence of the system problem was not achieved. This failure is attributed, not to a failure of the optimizer, but to the specific CO formulation. In fact, the problem is related to the combination of the optimizer and formulation details, since the exact problem and formulation used in [32] can be solved in a relatively efficient and very robust manner by other numerical optimizers (even the solver in Microsoft Excel). This is not to say that this feature of the formulation is to be ignored. Indeed SQP methods are some of the most efficient methods available for constrained problems and most of the CO problems that have been studied have also employed NPSOL. In [22] a large set of quadratic test problems was studied using a version of CO with NPSOL, achieving convergence in every case, so the difficulties cited may require additional consideration, but improvements in this aspect of CO have been the focus of recent work.

Alternate choices for the form of subspace objectives, system constraints, and optimizers are discussed in [20] and appear promising. Two other approaches have also been studied in this regard: the use of a less efficient system optimizer that is more tolerant of non-smooth functions, and the use of response surfaces to model the system constraints. The latter avoids the problem with singular Jacobians, not by smoothing the functions it represents, but by providing such an inexpensive representation of the function that very robust but inefficient optimizers are perfectly acceptable. This approach is discussed in more detail in the following section and in Refs. [38, 39].

When the conventional CO formulation is employed several small steps can make large differences in system performance. These include the following:

1. Check that the system level optimizer is capable of functioning on problems such as this. The SQP method if MatLab, for example, behaves very poorly on CO system-level problems, while NPSOL is more robust, and some reduced gradient approaches appear to work very well.
2. Use post-optimality gradient information. The original CO formulation was structured so that gradients of the system compatibility constraints would be available. When quadratic compatibility constraints are used, the system targets have no direct influence on the subspace constraints and so the gradient components of the i^{th} constraint with respect to the j^{th} system target variable may be written simply as:
$\partial J_i^* / \partial z_j = 2\,(z_j - z_j'^*)$
If this additional information is not used, a different formulation should be used.
3. The system level compatibility constraints may be written as: $J_i = 0 \;\forall$ i subproblems. However, because of the way most optimizers solve the constrained problem, large performance gains may be achieved by using $J_i \leq 0$, even though $J_i = 0$.

4.4. CO and Surrogate Models

The efficient incorporation of approximation models simplifies some aspects of collaborative optimization. This was addressed initially in [38], but additional developments are ongoing [cf. 40]. Multi-level methods may be better suited to, and benefit more from, the use of approximate modeling than other optimization methods. Techniques such as response surfaces are particularly appealing in CO for several reasons. When used to model the results of subspace optimization, the dimensionality of the response surface can be much smaller than would be required for fitting an integrated analysis system. Problems with low dimensionality interdisciplinary coupling, natural candidates for solution by CO, also gain most using this approach. Along with the usual response surface features that aid in parallel execution and load balancing, two other considerations suggest the important synergy that exists in this combination. As has been mentioned, the use of approximate models for the system constraints sidesteps an important efficiency difficulty with some versions of the method. In this way response surfaces can help CO. Conversely, we can exploit knowledge about the problem itself to improve the efficiency of the approximate model generation.

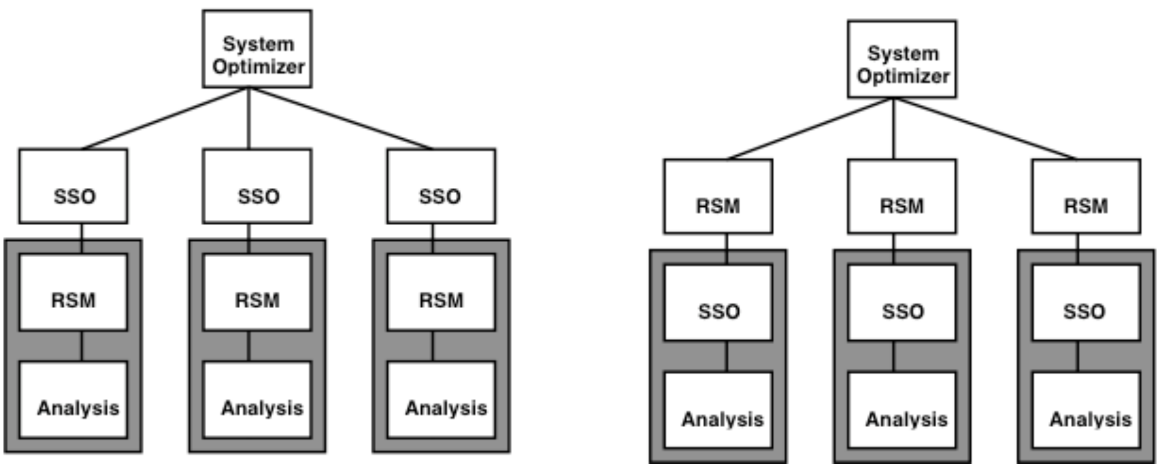

Figure 11. Use of response surface or other surrogate methods in CO. Left: modeling analysis results. Right: more efficient modeling of subspace design results.

Efficient model generation for CO is enabled by the fact that we do not require a method that approximates a general function, only functions that are solutions to CO subproblems. For example, to fit an optimum subspace objective, J_1^*, contours of which are shown in figure 12, as a function of the system targets a1 and x2 would likely require at least a dozen function evaluations. As can be seen from the figure, even if the function is quadratic in each region of the solution, it cannot be fit globally with a single quadratic approximation. And even if a local quadratic fit was sufficient, at least six solutions of the subproblem would be needed.

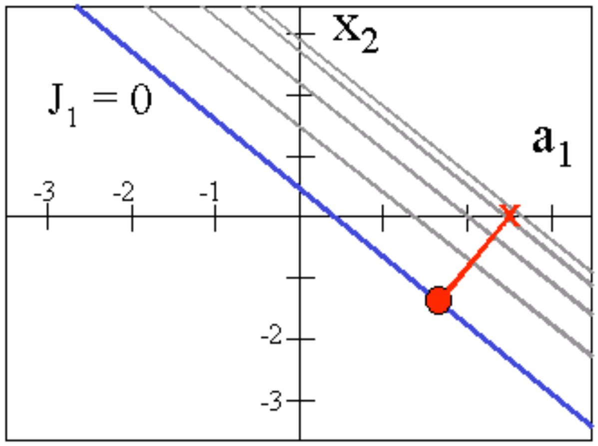

Figure 12. Function knowledge used to construct fit.

The solution to the subproblem provides much more than the value of J_1^*, however, and this information can be used to great advantage. Assume that, as shown in figure 12, the first target values are given as a_1=2.5, x_2=0. The subspace solution in this case is a_1'=1.5, x_2'=-1.0. Since the subspace returns these values, it is known that if the original targets were a_1=1.5, x_2=-1.0, the subspace could satisfy its constraints and match the targets exactly. So by evaluating only one point in the 2D space, the solution at two points is known. A quadratic model could therefore be built with as few as three subspace evaluations. Moreover, not only do the a_1=1.5, x_2=-1.0 target values lie on the J_1^*=0 contour, but this must be the closest point on the contour to the initial point, so the contour must be normal to the vector connecting these two points as shown in figure 12. Combined with other knowledge about the formulation, an exact fit of a quadratic problem is possible with only one subspace optimization. Of course general problems would take several approximation refinement steps (e.g. using trust region methods such as those in [41,42]) before a reasonable approximation was achieved; nonetheless, this represents a large improvement in efficiency as was demonstrated recently in [40].

5. Applications

In parallel with this method development, several researchers have applied CO to both simple test problems [20-22,32] and more complex design problems [43-51]. The latter have included applications to launch vehicle design, trajectory optimization, aeroelastic design, supersonic aircraft optimization, conceptual bridge design, undersea vehicles, and unmanned aircraft. This experience has highlighted both the advantages and problems associated with collaborative optimization.

Braun applied collaborative optimization directly to the design of launch vehicles, combining sizing, cost, and trajectory design tools that had been developed previously [44]. Achieving the correct optimal system with both an all-at-once (integrated) method and CO, Braun concluded that the advantages associated with decoupling outweighed the increased computation time that was required for this problem.

Manning [46] used the response surface-based CO to combine aerodynamic design, structural sizing, and mission optimization in a study of supersonic aircraft (figure 13). Aerodynamic analysis relied on a high order surface panel model while structural analysis was performed with a finite element code.

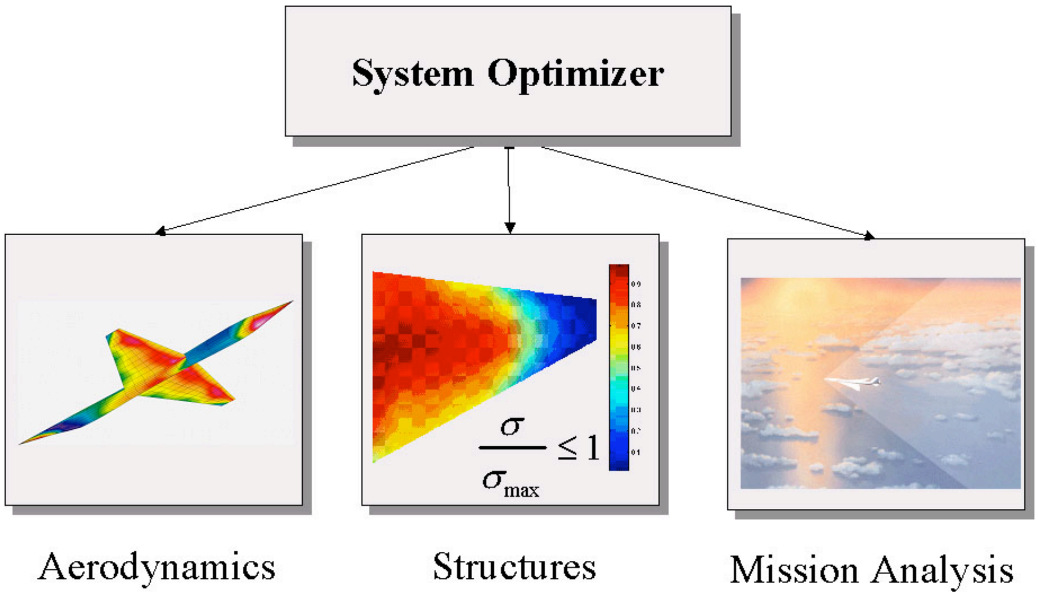

Figure 13. Solution of a large scale optimization problem (supersonic design) using collaborative optimization in [46] explored issues in integrating more complex subproblems than had been done previously.

The goal was to satisfy mission performance constraints along with structural constraints while minimizing the required take-off weight. Response surface models were updated using the trust region scheme described in [42]. In this work, a penalty function method was used to achieve system-level compatibility, with the system level objective defined as: J = take-off gross weight (TOGW) + Σ k_i J_i, with k_i a weighting coefficient and J_i, the subspace compatibility objectives. Later, the same approach was used to avoid numerical issues with SQP methods [20], but in this case, the choice assured that despite the approximate nature of the response surface models, the system level problem remained feasible. Several updates of the quadratic response surface models were required, with compatible, nearly optimal solutions reached after about a dozen update cycles (figure 14).

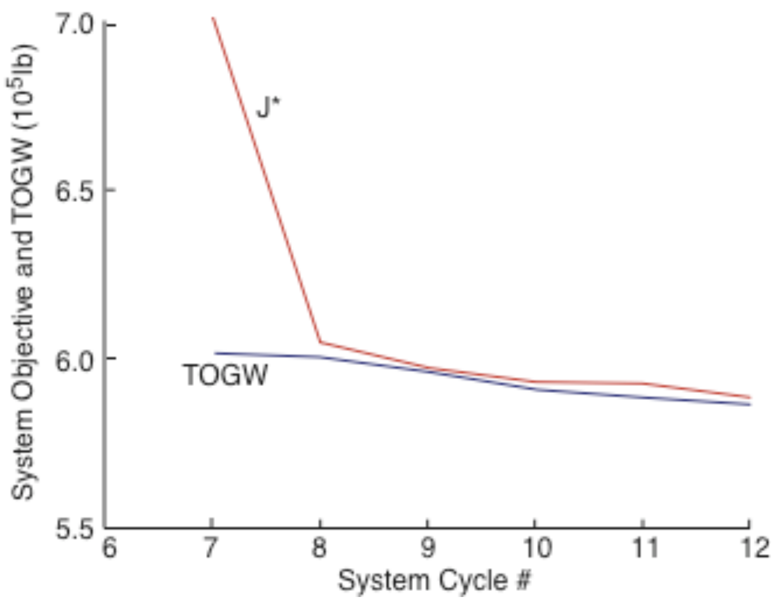

Figure 14. A compatible, good design was found using surrogate-based CO, but multiple trust region update cycles were required.

Recently, a variant of the CO approach was used for the distributed path planning problem involving four aircraft that minimized their flight path while satisfying separation constraints [52]. Figure 15 illustrates a typical result in this domain – a natural application of distributed optimization that will likely see additional development in related multiple vehicle optimization and control problems.

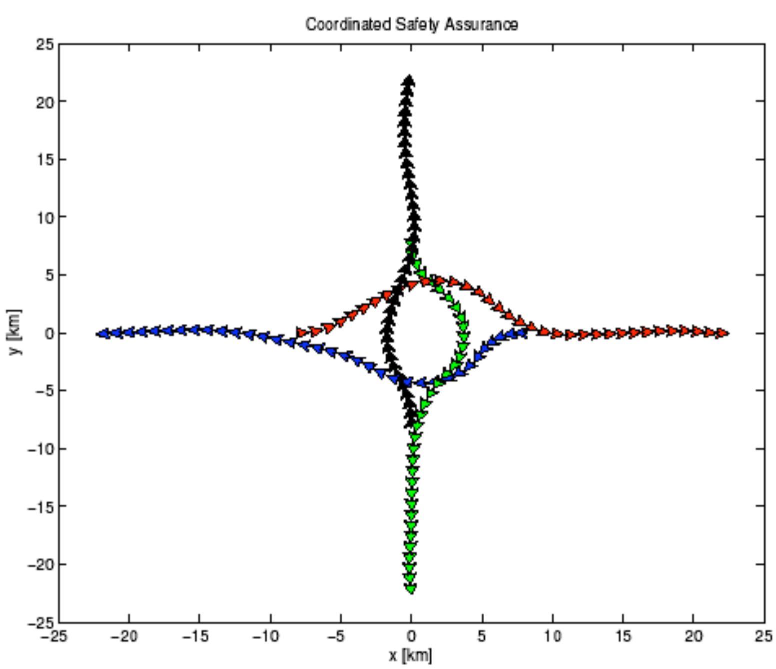

Figure 15 (from [52]). Optimizing aircraft routing using a decentralized optimization algorithm closely related to CO.

Additional examples of CO applications include the conceptual design of bridges in [50], launch vehicles in [44] and[49], underwater vehicles [45], wing design [47], and inlet/ejector systems [51].

Several advantageous features of collaborative optimization have been demonstrated in these applications but since CO is still relatively immature, little experience in true industrial environments is available. The examples cited here do show that the software integration task is expedited, since the process can be asynchronous and communication requirements are minimized. In addition, the computational effort can be divided easily among multiple platforms with heterogeneous hardware and software, including different optimizers for different disciplines. It is also anticipated that in an actual industry implementation the maintenance of disciplinary design variables and constraints within each disciplinary design task will be advantageous, matching closely the structure of existing practice. Many of the anticipated beneficial features of the architecture only exist for large, multidisciplinary problems, making it difficult to demonstrate with typically small test problems.

6. New Ideas

Current research on distributed design optimization has focused on modifications to CO or BLISS aimed at improving overall efficiency, permitting their use on problems with high dimensionality coupling, and simplifying their implementation.

Although the use of approximate models with CO or BLISS resolves many of the issues with slow system-level convergence additional, and probably substantial improvements in efficiency are possible by using all available information from the subspace design problems in order to construct approximate models of optimal target copies (the z' variables in figure 7). Since evaluation of the approximate models is inexpensive, explicit compatibility constraints (of the form $z = z'^*$) may be imposed for the system-level optimization, avoiding the large eigenvalues encountered in some previous formulations. This also frees the subspaces from having to use the system constraints as local objectives. Revised local objectives that include a combination of compatibility penalty and system objective are factored [53] with respect to the system level problem and improve the performance of some subspace optimization schemes that had difficulty with zero eigenvalues when the objective values became small.

Two approaches to handling problems with high-dimensionality interdisciplinary coupling in CO are being studied presently. The difficulty with such problems is that decoupling using optimizer-based decomposition introduces large numbers of system variables, leading to inefficient solution. The simplest approach to this problem is to represent the coupling using a reduced order model. This can be done accurately and efficiently for many problems with the suitable selection of basis functions as shown in [Manning]. Improved methods for generating such basis functions in the special case of CO coupling are possible. The second approach involves a more fundamental change to the CO architecture. Note that target copies were introduced to guarantee that the decomposed subspaces would be feasible. Yet if target values are computed from one routine (A) and required as input to another (B), the second routine does not require a set of target copies that span the entire space, but rather only a small subset to ensure feasibility of the decomposed problem. During the course of sampling for development of an approximate model for A the computed target solutions can be used to form an effective reduced basis set for B. A similar approach may be used when population-based design methods are employed for solution to the system-level problem, with computed results from A inserted directly into the population.

Simplified implementation of distributed optimization systems is needed to make the creation of this type of system not a research project in itself. Some steps in this direction have been taken by incorporating BLISS into the ModelCenter framework [54] and with a similar framework, designed for multilevel optimization from its inception [33]. More dramatic simplifications to distributed multidisciplinary design may be possible with single-level distributed systems, including versions of collectives [53] for design.

7. Conclusions

Distributed MDD and collaborative optimization, while still maturing, has been used successfully in many applications. It is ideal for integrating *design* codes that include their own optimizer as is often the case for trajectory optimization or optimal control design and for large scale problems with low dimensionality coupling between the subproblems.

The approach can be inefficient when not carefully employed. As with any optimization problem, the user much match the optimization algorithm to the problem of interest and understand issues related to scaling and function smoothness. Work remains on improving the robustness and efficiency of the system-level coordination problem through variations in the form of the subspace objectives and system constraints or through optimization algorithms better suited to the system-level problem.
CO with response surfaces that model subspace design results continues to appear promising. New general fitting techniques such as kriging may be useful here, but custom methods tailored to subspace fitting may be more efficient. New approaches to reduced basis modelling and trust region heuristics for efficient convergence will also be directly applicable.

The recognition that distributed design may be viewed as a multi-agent game has led to new ideas for both multi-level and singe-level distributed optimization. New techniques and additional applications of these ideas in multidisciplinary design, system of systems design, and multi-agent control are sure to make this an exciting field over the next several years.

8. Acknowledgments

Much of the work summarized here has been accomplished by doctoral students in the Aircraft Aerodynamics and Design Group at Stanford (in particular Drs. Ian Sobieski, Robert Braun, Valerie Manning, and Peter Gage). Interested readers should consult their papers and theses listed in the references section for further details. Support from NASA Langley Research Center and the Boeing Company was critical in the development of these methods and is gratefully acknowledged.

9. References

[1] Ashley, S., "Engineous Explores the Design Space," Mechanical Engineering, Vol. 114, No. 2, pp. 49-52, Feb. 1992.

[2] Sobieszczanski-Sobieski, J., “A Linear Decomposition Method for Large Optimization Problems,” NASA TM-83248, Feb.1982.

[3] Cramer, E., Dennis, J. Jr., Frank, P., Lewis, R., Shubin, G., “Problem Formulation for Multidisciplinary Optimization,” *SIAM Journal on Control and Optimization*, Vol.4, No.4, 1994, pp.754–776.

[4] Sobieszczanski-Sobieski, J., James, B., Dovi, A., “Structural optimization by multilevel decomposition,” in Proceedings of the AIAA/ASME/ASCE/AHS 24th Structures, Structural Dynamics, and Materials Conference, May 1983. AIAA Paper 83-0832-CP.

[5] Schmit, L.A., Jr., Mehrinfar, M., “Multilevel optimum design of structures with fiber-composite stiffened-panel components,” AIAA Journal, 20 1982, pp. 138-147.

[6] Sobieszczanski-Sobieski, J., James, B., Riley, M. F., “Structural sizing by generalized, multilevel optimization,” AIAA Journal, 25, pp. 139-145, 1987.

[7] Barthelemy, J. F. M., Riley, M.F., “Improved multilevel optimization approach for the design of complex engineering systems,” AIAA Journal, 26, pp. 353-360, 1988.

[8] Sobieszczanski-Sobieski, J., “Two alternative ways for solving the coordination problem in multilevel optimization,” Structural Optimization, 6, pp. 205-215, 1993.

[9] J. Walsh, K. Young, J. Pritchard, H. Adelman, W. Mantay, “Integrated aerodynamic / dynamic / structural optimization of helicopter rotor blades using multilevel decomposition,” Tech. Rep. NASA TP 3465, January 1995.

[10] A. Migdalas, P. Pardalos, and P. Varbrand, eds., *Multilevel Optimization: Algorithms and Applications*, Kluwer, 1998.

[11] Wujek, B., Renaud, J., Batill, S., "A Concurrent Engineering Approach for Multidisciplinary Design in a Distributed Computing Environment," in Multidisciplinary Design Optimization: State of the Art , N. Alexandrov and M. Y. Hussaini, editors, SIAM, 1997.

[12] Eason, E., Wright, J., "Implementation of Non-Hierarchic Decomposition for Multidisciplinary System Optimization," AIAA 92-4822, AIAA/NASA/AirForce Multidisciplinary Optimization Conference, Sept. 1992.

[13] Renaud, J., Gabriele, G., "Improved Coordination in Non-Hierarchic System Optimization," *AIAA J.*, Vol 31, No. 12, pp. 2367-2373, Dec. 1993.

[14] Seller, R., Batill, S., Renaud, J., "Response Surface Based, Concurrent Subspace Optimization for Multidisciplinary System Design," 34th AIAA Aerospace Sciences Meeting, Jan. 15-18, 1996, AIAA Paper No. 96-0714.

[15] Braun, R., Collaborative Optimization: An architecture for large-scale distributed design, Ph.D. thesis, Stanford University, May 1996.

[16] Sobieszczanski-Sobieski, J., Agte, J., Sandusky, R. Jr., "Bi-level integrated system synthesis (BLISS)," AIAA-1998-4916, AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 7th, St. Louis, MO, Sept. 2-4, 1998, Collection of Technical Papers.

[17] Sobieszczanski-Sobieski, J., Emiley, M., Agte, J., Sandusky, R. Jr., "Advancement of Bi-Level Integrated System Synthesis (BLISS)," AIAA-2000-421, Aerospace Sciences Meeting and Exhibit, 38th, Reno, NV, Jan. 10-13, 2000

[18] Sobieszczanski-Sobieski, J., Altus, T., Phillips, M., Sandusky, R. Jr., "Bilevel Integrated System Synthesis for Concurrent and Distributed Processing," AIAA-2002-5409 and *AIAA J.*, Vol. 41, No. 10, October2003.

[19] Balling, R. J., Sobieszczanski-Sobieski,J., "An algorithm for solving the system-level problem in multilevel optimization," AIAA Paper 94-4333, Fifth AIAA / USAF / NASA / ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Florida, September 7-9, 1994.

[20] DeMiguel, A., Murray, W., "An Analysis of Collaborative Optimization Methods," in Eighth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization AIAA-2000-4720, Sept. 2000.

[21] Kroo, I., Altus, S., Braun, R., Gage, P., Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," AIAA 94-4325, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, Sept. 7-9 1994. Vol. 1, pp. 697-707.

[22] Braun, R.D., Gage, P., Kroo, I., "Implementation and performance issues in collaborative optimization," *AIAA/NASA/ISSMO, Symposium on Multidisciplinary Analysis and Optimization, 6th, Bellevue, WA, Sept. 4-6, 1996* AIAA Paper 96-4017.

[23] Braun, R. D., and Kroo, I. M., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *Multidisciplinary Design Optimization: State of the Art*, edited by N. Alexandrov and M.Y. Hussaini, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, Hampton, Virginia, March 13-16, 1995, published in 1997, pp. 98-116.

[24] Kroo, I., "MDO for Large-Scale Design," in *Multidisciplinary Design Optimization: State of the Art*, N. M. Alexandrov and M. Y. Hussaini, eds., SIAM, 1997, pp. 22-44.

[25] Braun, R.D., Kroo, I., Gage, P., Post-Optimality Analysis in Aerospace Vehicle Design," AIAA-93-3932, Aug. 1993.

[26] Peterson, J., Garfield, J. , "The automated design of multistage solid rocket vehicles," AIAA 1976-744 and J. of Spacecraft & Rockets, Vol. 11, 1977.

[27] Thareja, R., R. T. Haftka, "Numerical difficulties associated with using equality constraints to achieve multi-level decomposition in structural optimization.," AIAA Paper 86-0854, 1986.

[28] Sobieszczanski-Sobieski, J., James, B., Riley, M.,"Structural Optimization by Generalized, Multilevel Decomposition," AIAA Journal Vol. 25, No. 1, pp139-145, Jan. 1987.

[29] V. Toropov, P. Zadeh, "Use of Global Approximations in the Collaborative Optimization Framework," AIAA-2004-4654, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug. 30-1, 2004

[30] C. McAllister, L. T. Simpson, , K. Lewis, State University of New York, Buffalo, NY; A. Messac, , "Robust Multiobjective Optimization Through Collaborative Optimization and Linear Physical Programming," AIAA-2004-4549, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug. 30-1, 2004.

[31] Lin, J. "Analysis and Enhancement of Collaborative Optimization for Multidisciplinary Design," AIAA Journal 2004, 0001-1452 vol.42 no.2 (348-360)

[32] Natalia M. Alexandrov; Robert Michael Lewis, Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design, AIAA Journal 2002, vol.40 no.2 (301-309).

[33] Kroo, I., Manning, V., "Collaborative Optimization -- Status and Directions," AIAA-2000-4721, AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary

Analysis and Optimization, 8th, Long Beach, CA, Sept. 6-8, 2000

[34] Shankar, J., Ribbens, C., Haftka, R., Watson, L., "Computational study of nonhierarchical decomposition algorithm," *Computational Optimization and Applications*, 2:273–293, 1993.

[35] P. E. Gill, W. Murray, M.A. Saunders, and M. H. Wright, User's guide for NPSOL (Version 5.0): A Fortran package for nonlinear programming, Stanford University, 1995.

[36] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[37] Kroo, I.; Altus, S.; Braun, R.; Gage, P.; and Sobieski, I.: "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," AIAA 94-4325, Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, September 7-9, 1994, Panama City, Florida.

[38] I. P. Sobieski, Collaborative Optimization Using Response Surface Estimation, Ph.D. thesis, Stanford University, 1998.

[39] I. P. Sobieski; I. M. Kroo, Collaborative Optimization Using Response Surface Estimation," AIAA Journal, vol.38 no.10 (1931-1938), 2000.

[40] Hu, Y., Li, W., "A New Effective Multidisciplinary Design Optimization Algorithm," ICAS 2002-1.4.1, 23rd ICAS Congress, Toronto, Sept. 2002

[41]. Rodriguez, J., Renaud, J., Watson, L., "Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization," ASME Journal of Mechanical Design, Vol. 120, No. 1, pp. 58-66, 1998, published by the American Society of Mechanical Engineers.

[42] Alexandrov, N., Dennis, J.E., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," NASA TM 112870, 1997.

[43] Braun, R., Powell R., Lepsch, R., Stanley, D., and Kroo, I., "Comparison of Two Multidisciplinary Optimization Strategies for Launch Vehicle Design," Journal of Spacecraft and Rockets, Vol. 32, No. 2, Mar.-Apr. 1995, pp 404-410.

[44] Braun, R., Moore, A., Kroo, I., "Collaborative Approach to Launch Vehicle Design," Journal of Spacecraft and Rockets, 34 (1997), pp. 478-486.

[45] Belegundu, A.D., Halberg, E., Yukish, M.A., Simpson, T.W., "Attribute-Based Multidisciplinary Optimization of Undersea Vehicles, AIAA 2000-4865, Sept. 2000.

[46] Manning, V., High Speed Civil Transport Design via Collaborative Optimization," Ph.D. thesis, Stanford University, 1999.

[47] S. Jun and Y. Jeon,, J. Rho, D. Lee, "Application of Collaborative Optimization Using Response Surface Methodology to an Aircraft Wing Design," AIAA-2004-4442, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug. 30-1, 2004.

[48] Z. Huque and N. Jahingir, "Application of Collaborative Optimization on a RBCC Inlet/Ejector System," AIAA-2002-3604, 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Indianapolis, Indiana, July 7-10, 2002

[49] Cormier, T.A., Scott, A., Ledsinger, L., McCormick, D., Way, D., Olds, J., "Comparison of collaborative optimization to conventional design techniques for a conceptual RLV," AIAA-2000-4885, AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 8th, Long Beach, CA, Sept. 6-8, 2000

[50] Rawlings, M., Balling, R., "Collaborative optimization with disciplinary conceptual design," AIAA-1998-4919, AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 7th, St. Louis, MO, Sept. 2-4, 1998.

[51] Butuk, N., Huque, Z., Lynch, D., "Optimization of an integrated inlet/ejector of an RBCC engine using collaborative optimization," AIAA-1998-3567, 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cleveland, OH, July 13-15, 1998.

[52] Inalhan, G., Stipanovic, D. M., Tomlin, C. J., "Decentralized Optimization, with Application to Multiple Aircraft Coordination," In the Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, December 2002.

[53] Wolpert, D.H., Tumer, K., "Avoiding Braess' Paradox through Collective Intelligence," Tech Rep No. NASA-ARC-IC-1999-124, Jan. 2000.

[54] Kim, H., Ragon, S., Soremekun, G., Malone, B., Sobieszczanski-Sobieski, J., "Flexible Approximation Model Approach for Bi-Level Integrated System Synthesis," AIAA-2004-4545, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug. 30-1, 2004.