AIAA JOURNAL
Vol. 45, No. 8, August 2007

Multi-Objective Pareto Concurrent Subspace Optimization for Multidisciplinary Design

C.-H. Huang*
Industrial Technology Research Institute,
310 Hsin Chu, Taiwan, Republic of China
and
J. Galuski† and C. L. Bloebaum‡
University at Buffalo, Buffalo, New York 14260

DOI: 10.2514/1.19972

Most real-world design problems are complex and multidisciplinary, with almost always more than one objective (cost) function to be extremized simultaneously. The primary goal of this research is to develop a framework to enable multi-objective optimization of multidisciplinary design applications, wherein each discipline is able to retain substantial autonomous control during the process. To achieve this end, we have extended the capability of the concurrent subspace optimization method to handle multi-objective optimization problems in a multidisciplinary design optimization context. Although the conventional concurrent subspace optimization approach is easily able to deal with multi-objective optimization problems by applying the weighted sum approach, the main disadvantage is that the weighted sum cannot capture Pareto points on any nonconvex part of the Pareto frontier. Moreover, an aggregate objective function simply cannot reflect the true spirit of the concurrent subspace optimization method, which was developed to allow groups of specialists to independently have more control over their own design criteria and goals, even while maintaining system level coordination. In this paper, the multi-objective Pareto concurrent subspace optimization method is proposed in which each discipline has substantial control over its own objective function during the design process, while still ensuring responsibility for constraint satisfaction in coupled subspaces. The proposed approach is particularly useful given the realities of geographical distribution, computational platform variation, and dependence upon legacy codes within individual disciplines that so predominates the design of large-scale products such as aircraft and automobiles. As part of the multi-objective Pareto concurrent subspace optimization method developed here, it is demonstrated that the endpoints of the Pareto frontier can be easily identified, together with an ability to generate Pareto points within prescribed limits to ensure a reasonably even distribution across the entire frontier. The approach is validated (using three multidisciplinary design optimization test problems) against Pareto frontiers generated using the weighted sum approach.

Nomenclature

C_K = cumulative constraint associated with Kth subspace
C_K^0 = current value of C_K at start of cycle
e_{Ki} = effectiveness coefficients for Kth subspace and ith design variable
F_K = objective function associated with Kth subspace
F_{SYS} = composite weighted system level objective function
F_K^0 = current value of F_K at start of cycle
r_K^P = responsibility coefficient in the Kth subspace associated with the Pth cumulative constraint
T_i = prescribed target for ith objective function
$X_{K\mathrm{sub}}$ = design variables allocated to the Kth subspace
ε = range threshold parameter
λ_{FP} = Lagrange multiplier for F constraint in Pth subspace
λ_{CP} = Lagrange multiplier for C constraint in Pth subspace
ρ = cumulative constraint parameter

Received 9 September 2005; revision received 31 March 2006; accepted for publication 25 October 2006.

*Research Scientist, Mechanical and Systems Research Laboratories; chhuang96@itri.org.tw.

†Graduate Student, Department of Mechanical and Aerospace Engineering; jgaluski@buffalo.edu.

‡Professor for Competitive Product and Process Design, Department of Mechanical and Aerospace Engineering; clb@buffalo.edu.

Introduction

IN RECENT years, industry has paid more attention to the formalization of complex system analysis and design (i.e., aircraft, automotive, and electronics industries, etc.) by seeking better ways to improve efficiency in the design process to meet time, cost, and quality demands. Multidisciplinary design optimization (MDO) has emerged as an engineering discipline that focuses on development of new design and optimization strategies for complex systems [1]. MDO problems, such as are found with automobile and aircraft design, involve interactions among several disciplines, which substantially complicate the design process. Hence, MDO researchers strive to reduce the time and cost associated with this "coupled interaction." Decomposition approaches provide many advantages for the solution of complex multidisciplinary problems, as they enable a partitioning of a large coupled problem into smaller, more manageable and tractable subproblems. The resulting computational benefits, besides the obvious one associated with the solution of smaller problems, include creating a potential distributed processing environment. The primary benefit, however, pertains to the savings in person hours, because groups are no longer required to wait around for other groups in the process to complete their design tasks.

Sobieski's landmark work in 1990 developed and presented the global sensitivity equation (GSE) method [2], aimed at providing total derivatives for nonhierarchic (i.e., coupled) systems. The feasibility and viability of the GSE was first demonstrated in [3,4]. Once the total system derivatives are found using the GSE, they can

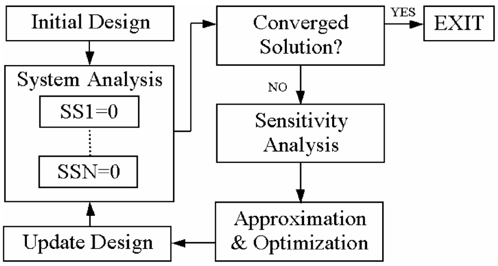


Fig. 1 General MDO design synthesis approach (MDF).

be used in a variety of MDO approaches to perform a multidisciplinary optimization for a coupled system. The general multidisciplinary design synthesis approach, as typified in a multidisciplinary feasible (MDF) [5,6] formulation, is illustrated in Fig. 1. Here, a coupled system analysis is performed at every cycle of the process, followed by a system level optimization in which sensitivity-based approximations are commonly employed. This MDF formulation assumes a single system level objective with a set of system level constraints and design variables.

The MDF [5,6], all-at-once (AAO) [5,6], individual-discipline feasible (IDF) [5,6], collaborative optimization (CO) [7,8], and concurrent subspace optimization (CSSO) [9,10] approaches are the major formulations that have been identified for posing and solving large-scale complex MDO problems. The primary differences in the MDF, AAO, and IDF formulations pertain to how often the full system analysis must be implemented. All these approaches assume a single system level objective function, although a couple of these have been extended to address multiple objectives.

The CSSO and CO approaches are two key architectures developed to support a collaborative multidisciplinary design environment through the use of distributed design optimizations among the different disciplinary groups. Another similar approach is the bilevel integrated system synthesis (BLISS) [11], which alternates between disciplinary problems, dealing with a set of local design variables and a system level optimizer, which deals with system level design variables. CSSO and CO were developed to mimic the design process in a design organization where tasks are distributed among groups of specialists. Hence, both methods enable the disparate disciplinary design teams to have a substantial degree of autonomous control. In CSSO, each subspace optimization minimizes the system objective function subject to its own constraints as well as constraints contributed from the other subspaces, with a system level coordination optimization procedure whose goal is to update coupling parameters. In CO, each subspace or disciplinary subset minimizes the difference between local design variables representing disciplinary couplings and target coupling values, subject to local constraints, with a system level optimizer as a separate coordination procedure. However, CO is most suitable for loosely coupled MDO problems, because the CO problem formulation size increases dramatically with coupling density [12,13]. Conversely, the number of auxiliary design variables (i.e., responsibility and tradeoff variables used in the subspace problem formulations) introduced in CSSO only depends on the number of disciplines. Moreover, with CSSO's design variable allocation feature, the subspace optimizations are performed concurrently with respect to a disjoint subset of design variables, which substantially reduces the complexity of the optimization problem within each disciplinary group. This provides designers a significant potential benefit in terms of computational effort. Both the CSSO and CO methods were developed initially for a single objective system level MDO problem.

The reality, however, is that use of a single objective function is typically a forced fit in multidisciplinary design optimization applications. Generally, there is a set of conflicting objective functions. Rather than forcing some of these functions to be represented as constraints, it would be preferable to use a multi-objective optimization approach in which the distributed environment that inherently exists can become an advantage, rather than a bottleneck. This is the goal in developing the multi-objective Pareto CSSO (MOPCSSO) method in this paper.

MOP Background

The solution of multi-objective optimization problems (MOPs) yields a set of equally good optimal points, based on the optimization criteria specified. These points are considered Pareto optimal points, the set of which defines the Pareto frontier. Ultimately, from a design perspective, a single point must be chosen as the best or most acceptable solution, which will then lead to development of a product. This choice requires human preferences and decisions. Therefore, it is essential to provide as many alternative solutions as possible to aid this decision making. Hence, the main goal in MOP is to generate sufficient Pareto points (i.e., capture the completeness of the Pareto frontier) so as to allow designers to make rational decisions. There are two ways in which the Pareto frontier becomes important for design. First is the generation of the Pareto frontier and second is the selection of a single design point or small set of points on the Pareto frontier to represent the final best design.

MOPs, in contrast to single objective problems, involve some set of objectives that might be cooperative or competitive in nature. Typically, the case of competitive objectives is the most interesting in the research field of MOP, because the choice of an "acceptable" or "best" solution relies on the tradeoffs of the objective functions, which ultimately depends on human preferences and decisions. Moreover, most engineering design problems can be categorized into this type of MOP (e.g., automobile design, aircraft design, etc). Certainly, MDO problems do, given the coupled nature of the disciplines involved.

Competitive objectives in a MOP always result in a set of solutions termed Pareto optimum solutions, wherein no one solution is superior to the others. The concept of Pareto optima was first introduced by Vilfredo Pareto in the early 1900s [14]. Mathematically, for minimization problems, the MOP can be formulated as follows:

$$\begin{aligned} \min \quad & F_i(\mathbf{X}) \quad && i = 1, p \\ \text{s.t.} \quad & g_j(\mathbf{X}) \leq 0 \quad && j = 1, m \\ & h_k(\mathbf{X}) = 0 \quad && k = 1, l \end{aligned} \tag{1}$$

where p is the number of objective functions, m is the number of inequality constraints, and l is the number of equality constraints. A feasible point X^* is said to be a Pareto optimum (or strong Pareto optimum) if and only if there is no other feasible point such that $F_i(X) \leq F_i(X^*)$ for all objective functions and $F_i(X) < F_i(X^*)$ for at least one objective function. Therefore, the Pareto optimum always gives a set of solutions instead of one solution, with the region defined by the Pareto optimum called the Pareto frontier. The necessary conditions for Pareto optimality for a multi-objective problem are referred to as the Karush–Kuhn–Tucker (KKT) conditions [15].

There are many methods available to solve multi-objective optimization problems, and they can be classified as either heuristic based or quantitative based. Heuristic-based methods (non-derivative based) include the multi-objective simulated annealing method [16,17] and the multi-objective genetic algorithm method [18–21], among others. Quantitative-based methods (derivative-based) include the weighted sum method [22], the goal programming method [23,24], the compromise programming method [25], the min–max method [26], the constraint method [27], and the physical programming method [28,29], to name a few of the key methods. The main disadvantage of the weighted sum method is that it is unable to capture Pareto points on the concave part of the Pareto frontier. The constraint method is able to capture any Pareto point on both the convex and nonconvex Pareto frontier but a drawback is that designers must have a priori knowledge of the suitable range of values for the ε parameter used in the formulation, which is generally not possible. The compromise programming method is also able to capture any Pareto point on both the convex and nonconvex portions of the Pareto frontier. In this approach, however, the utopia point must be known to solve the MOP, requiring additional computational effort. If the computational effort is not extremely expensive, finding the utopia point may prove to be worthwhile because this information is often useful as a reference and can contribute to the generation of an evenly distributed Pareto frontier [29].

There have been several efforts to extend MOP applications to MDO problems. Among these is the multi-objective collaborative optimization method [30,31], the multi-objective multidisciplinary genetic algorithm (M-MGA) method [32], and the interactive multi-objective optimization design strategy (iMOODS) [33]. There have been a few approaches developed for multi-objective collaborative optimization, including a genetic algorithm based approach (M-MGA) [32]. These approaches retain the same drawbacks of the single objective CO method, in that problem size increases with increased numbers of couplings. The iMOODS approach incorporates designer preferences interactively through the use of Pareto sensitivities. The designer is guided along the Pareto frontier based on objective function tradeoffs. This will yield a single Pareto point that already reflects designer preferences, but does not provide flexibility for making changes in preferences after selection.

None of these methods can easily deal with the challenging issues of 1) highly coupled MDO problems; 2) separate but coupled disciplinary groups; 3) a desire to avoid the need for a priori knowledge for parameter setting at the MOP level; and 4) a desire to capture the entire Pareto frontier, including nonconvex regions, to foster flexibility in later tradeoff studies. The proposed MOPCSSO deals with all these issues.

As previously mentioned, the choice of the best or most acceptable among the Pareto solutions in a MOP usually involves a human decision. Hence, depending on the participation of designer's preferences in the optimization process, these methods can be further classified into four different categories, as follows: 1) the no preference methods (never establish preferences); 2) the priori methods (preferences set before implementation); 3) the posteriori methods (preferences implemented after optimization); and 4) the interactive methods (preferences set during implementation) [34]. Here, the proposed MOPCSSO is categorized as a no preference method. It would be used to populate the Pareto solutions using a collaborative multidisciplinary design environment wherein distributed design optimizations are performed among the different disciplinary groups. It should be noted that MOPCSSO is an objective driven approach because the final Pareto solution is heavily dependent on the initial objective function values. This is easily addressed through the implementation of a forced Pareto solution distribution approach. Further, the capability of capturing any point on the Pareto frontier is a basic requirement for solving MOPs.

In the research field of MOP, the GA-based evolutionary algorithms have been widely adopted to populate the Pareto frontier because they are able to generate a set of Pareto solutions after a single run of optimization. However, these algorithms are considered inefficient and computationally costly. Moreover, their performance highly depends on the appropriate setting of internal parameters that are often problem dependent (such as population size, sharing factor, fitness assignment, penalty coefficients, etc.) [19,20]. Although derivative-based methods must perform a series of separate runs to generate sufficient Pareto points, they are easier to apply and the solutions are generally more accurate.

In this work, the CSSO method is used as an enabling structure to develop the multi-objective Pareto concurrent subspace optimization method that can be used to populate a Pareto frontier for a complex MDO problem. The next section presents a brief overview of the CSSO method, as a lead into the MOPCSSO development that follows. Validation results are presented for the method to demonstrate its viability and efficacy. The Conclusions section follows the discussion of results.

CSSO Method Background

Sobieski first proposed the subspace optimization method in 1988 [9], and the method was further developed and subsequently named the concurrent subspace optimization method [10]. Renaud [33–35] developed a second-order variant of the GSE method and an alternative potential coordination procedure for the CSSO method, as well as response surface based representations and shared design variable strategies, among other investigations. In this study, the original CSSO method of [9,10] is used for the development of the multi-objective Pareto concurrent subspace optimization method.

The CSSO method allows a complex coupled system to be decomposed into smaller, temporarily decoupled subspaces, each corresponding to different disciplines. Subsequently, the optimization in each of these subspaces can be achieved concurrently. Therefore, the CSSO method is particularly suited to applications in a design organization where tasks are distributed among different analysis groups. The flowchart of the CSSO method is shown in Fig. 2.

Design variables are initially distributed among disciplinary groups according to historical precedence. Each group that has responsibility for a subset of the design variables initializes them according to their experience and prior practice. After a system analysis convergence, the GSE is used to find system derivatives. On the first pass of the method, design variables are reallocated to the most appropriate subspaces (i.e., wherein they have the greatest impact). Although the design variables could remain with the disciplinary groups that originally claimed them, the reality is that the multidisciplinary nature of the optimization problems often results in variables having a greater impact, through the inherent couplings that exist, in a different discipline altogether. Reallocating variables to the subspaces where they have the greatest impact on both objectives and constraints enables them to achieve the greatest reduction in the function possible while maintaining constraint satisfaction. Further, if a variable is incorrectly allocated to a discipline where it has an inherently limited design freedom (i.e., due to only a very small impact on the objective function and constraints), it is possible that constraint feasibility might not be able to be achieved, with a resultant lack of ultimate method convergence. Although it may seem initially that this limits the degree of autonomy for each participating discipline, it actually has little impact. Design variables still reside in their original discipline for the system analysis and sensitivity analysis phases. The reallocation is only within the subspace optimization phase of the method.

Following design variable allocation, coefficients of responsibility (r), tradeoff (t), and switch (s) are determined, which are subsequently used to enable individual disciplines to perform subspace optimizations simultaneously. The r coefficients capture each subspace's responsibility for satisfying constraints contained in

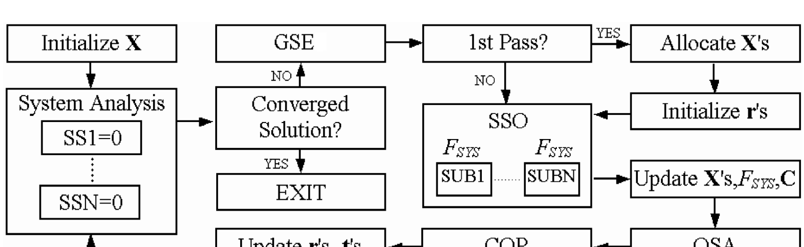


Fig. 2 Flowchart of CSSO method.

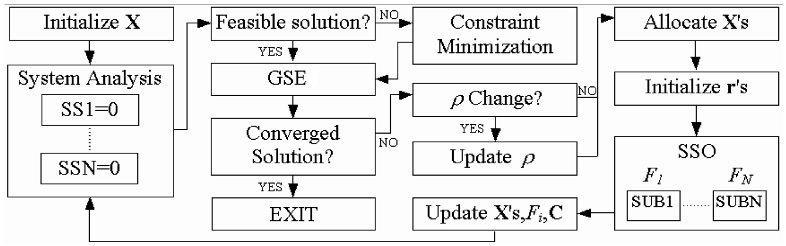


Fig. 3 Flowchart of MOPCSSO method.

other subspaces. In other words, because it is a coupled problem for which design variables have impact in multiple disciplines (i.e., subspaces), but for which they have been allocated to only one, it is recognized that they still have a responsibility for satisfying the constraints of other subspaces. The t coefficients signify a potential for trading off constraint violations in some subspaces, as long as there is oversatisfaction in others, in order to achieve a significant objective function savings that might not otherwise be obtained. The s coefficients switch the formulation from a responsibility to a tradeoff representation, as both cannot be implemented simultaneously. The constraints are represented within each subspace by a cumulative constraint formulation, denoted by C_K, which essentially represents an envelope function [36]. A parameter ρ is used to identify to what degree the cumulative constraint represents all the constraints or some subset of the most active or violated constraints.

Following the subspace optimizations are update and system level coordination procedures, including a coordination optimization problem (COP) in which new values of the parameters (r, s, and t) are determined, based on optimum sensitivity analysis (OSA) information. Hence, the CSSO method permits a substantial degree of autonomous control within each discipline while still maintaining a crucial coordination through the use of the coordination parameters. This optimization process continues until system level optimal convergence is achieved.

What is particularly attractive about the CSSO formulation for multidisciplinary design optimization is the significant degree of autonomy that is possible within each subspace. The CSSO method structure lends itself readily to a multi-objective optimization formulation, which would be philosophically just a further extension of such autonomy. In the next section, MOPCSSO method is developed.

MOPCSSO Method Development

In the CSSO method, design variable allocation and responsibility coefficient determination are the two crucial factors that affect the system feasibility after system decomposition. The design variables are assigned to the subspaces where they have the greatest impact. They may stay in that subspace until the end of the optimization process or they may be reallocated during the process depending on changing priorities. The implementations of the OSA and COP procedures have two significant purposes. One is to meaningfully update the responsibility and tradeoff coefficients (which represent the changing priorities). The other is to legitimize the design variable allocation. Because the goal of this work is to enable each disciplinary group to use their own objective function, without using a weighted sum approach to combine the objectives across subspaces, a system level objective function is no longer desirable or usable. Hence, it is not suitable to perform the OSA and COP, as originally developed, in a multi-objective MDO problem.

To replace the OSA and COP, three strategies have been adopted to modify the original concurrent subspace optimization method so as to prepare it for multi-objective formulations. It must first be noted that the tradeoff coefficients are eliminated in this formulation, since all tradeoffs will now occur directly with minimization of individual objective functions. The three modifications required to prepare CSSO for a multi-objective formulation (using responsibility coefficients only) include 1) the development of new design variable allocation and responsibility coefficient update procedures, because OSA and COP are no longer used; 2) the development of a procedure for updating cumulative constraint parameter ρ, which is critical for establishing responsibility coefficient values; and 3) the development of a procedure for ensuring the reduction of constraint infeasibilities before subspace optimizations, so as to ensure smooth convergence. The flowchart for the MOPCSSO is shown in Fig. 3. A description of the development associated with each of the three modifications follows.

Design Variable Reallocation and Responsibility Coefficient Determination

The purpose of the OSA and COP procedures in CSSO is to find new values for the coordination coefficients for use in the next optimization cycle. Because the OSA and COP procedures require a system level objective function, these must be replaced with a new mechanism for updating the responsibility coefficients required in MOPCSSO. Hence, the most straightforward approach is to reinitialize them at each cycle. This is substantially less of a computational cost than the OSA and COP procedures would be. However, to reinitialize the responsibility coefficients, it is critical that the design variables are always allocated to the subspaces where they have the greatest impact. Therefore, in this formulation, design variables are allocated at every cycle. There is only a minimal impact on computational efficiency because the needed derivative information for computing the effectiveness coefficients is already obtained after the GSE method. In actuality, the increased computations required to allocate design variables at every cycle are easily balanced by the fact that the OSA analysis and COP suboptimization process are no longer implemented.

Because MOPCSSO deals with individual objective functions concurrently, a new strategy for finding effectiveness coefficients [36] is developed to allocate the design variables appropriately. Clearly, with the newly added auxiliary function constraints, the impact of design variables with respect to each subspace must be quantified such that the effect of all objective functions in Eq. (2) should be accounted for. Hence, in this research, effectiveness coefficients corresponding to the Kth subspace are defined as follows:

$$e_{Ki} = \frac{\mathrm{d}C_K/\mathrm{d}X_i}{\sum_{K=1}^{NSS} \mathrm{d}F_K/\mathrm{d}X_i} \qquad i = 1, n \tag{2}$$

where NSS is the number of subspaces and n is the total number of design variables in the system. Through a comparison of all e_{Ki} coefficients, design variables can be allocated appropriately to the subspaces upon which they are determined to have the greatest impact. This is important for the update of the responsibility coefficients as well.

Update of the ρ Parameter

An inappropriate selection of ρ in the cumulative constraint formulations [36] within the subspace optimizations might lead to premature convergence. Hence, a simple modification strategy is developed and adopted here. In this research, ρ starts from a small value and increases in a user-defined step size when the design meets the criteria for updating ρ. The main reason to take such an approach is that, in the early stages of the optimization process it is desirable to have as much information as possible taken into consideration. This allows designers to not only capture the integrity of the optimization problem but also find the appropriate search direction. Hence, a smaller value of ρ initially allows more constraints to impact the optimization process, which then fulfills the intent stated above. As the optimization progresses, the value of ρ increases such that the critical constraints dominate the cumulative constraint. This then allows the optimization to proceed in the most effective direction.

Infeasibility Minimization

In connection with the responsibility coefficients, each subspace is responsible for reducing the violated cumulative constraints from the other subspaces. Because move limits are also typically imposed, a constraint minimum may not always be obtained at the completion of a subspace optimization due to the cumulative constraint violation from other subspaces. This can occur due to the way in which responsibility coefficients are assigned to each subspace. Because only the maximum sensitivities associated with each cumulative constraint are taken into consideration, the responsibility coefficients may not thoroughly reflect the responsibility of the allocated design variables toward other subspaces. An infeasible starting point is more likely to make this occur. Therefore, it is critical to devote some effort to generating a feasible (or at least less infeasible) starting point. Here, we choose to directly minimize the constraint infeasibilities (i. e., an unconstrained minimization with penalty-type objective function) [20], so that a feasible point is the one that will result in an objective function value of zero.

Subspace Optimization Problem Statement

In addition to the three modifications previously described, the more subtle modification in this development effort is in the definition of the subspace optimizations, which now have unique objective functions within each subspace and which incorporate the other subspace objective functions as constraints in such a manner that Pareto optimality can be assured. In MOPCSSO, objective functions from other subspaces are introduced as constraints during the subspace optimization phase in the sense that the solution path will satisfy the concept of Pareto optimality (i.e., implementing any objectives without worsening the others). The general form for the Kth subspace optimization can be represented as follows:

$$\begin{aligned} \min \quad & F_K(\mathbf{X}_{K\text{sub}}) \\ \text{s.t.} \quad & C_P(\mathbf{X}_{K\text{sub}}) \le C_P^0\left(1 - r_K^P\right) \quad P = 1, NSS \\ & F_P(\mathbf{X}_{K\text{sub}}) \le F_P^0 \quad P = 1, NSS \,\&\, P \ne K \end{aligned} \tag{3}$$

where NSS is the number of subspaces and $\mathbf{X}_{K\text{sub}}$ represents the design variables allocated to the Kth subspace. As an example, therefore, the MOPCSSO subspace formulation for a generic biobjective problem, with subspaces A and B, is stated as follows:

$$\begin{array}{ll|ll} \multicolumn{2}{c|}{\text{Subspace A}} & \multicolumn{2}{c}{\text{Subspace B}} \\ \min & F_A(\mathbf{X}_{A\text{sub}}) & \min & F_B(\mathbf{X}_{B\text{sub}}) \\ \text{s.t.} & C_A(\mathbf{X}_{A\text{sub}}) \le C_A^0\left(1 - r_A^a\right) & \text{s.t.} & C_A(\mathbf{X}_{B\text{sub}}) \le C_A^0\left(1 - r_B^a\right) \\ & C_B(\mathbf{X}_{A\text{sub}}) \le C_B^0\left(1 - r_A^b\right) & & C_B(\mathbf{X}_{B\text{sub}}) \le C_B^0\left(1 - r_B^b\right) \\ & F_B(\mathbf{X}_{A\text{sub}}) \le F_B^0 & & F_A(\mathbf{X}_{B\text{sub}}) \le F_A^0 \end{array} \tag{4}$$

It is conceivable that there might be a desire or need for multiple objectives within a single subspace, rather than a single objective. The MOPCSSO structure could certainly handle this because the autonomy enables the different subspace groups to implement whatever optimization routine they might want. If subspace A had two objectives, for instance, that group might run any standard MOP approach. However, they would need to eventually have a single set of values for the design variables allocated to that subspace. This means that preferences would have to be implemented within that subspace optimization to find a single Pareto point so that design variables, $X_{A\text{sub}}$ could be used as a starting point for the next cycle. The autonomy enabled by the MOPCSSO structure provides this type of flexibility within subspaces.

It should be noted that any optimization method used to solve these subspace optimization problems should include normalization to avoid any magnitude disparities related to the variables, objectives, and constraints. In addition to limiting the feasible design space, constraints also serve as a supplementary guidance for the search direction. In MOPCSSO, the improvement of each objective function depends more on its own subspace optimization rather than the imposed auxiliary objective constraint in other subspaces. The imposed auxiliary objective constraints in the other subspaces are simply acting as guidance for seeking the optimum in that subspace optimization. As a result, after the completion of subspace optimizations, the imposed auxiliary objective constraints should at least maintain their current value but will not necessarily be improved. In other words, the tradeoffs among different disciplinary objectives occur based on the current objective values. This is easily demonstrated if one considers an approximation on each objective function within each of the two subspaces defined in Eq. (4). Using a first-order Taylor series expansion (TSE), with respect to the design variables within subspace A, we obtain the following expressions as an extension of Eq. (4).

$$F_A(\mathbf{X}_{A\text{sub}}^*) = F_A^0 + \left.\frac{\mathrm{d}F_A}{\mathrm{d}\mathbf{X}_{A\text{sub}}}\right|_{\mathbf{X}_{A\text{sub}}^0} * \left(\mathbf{X}_{A\text{sub}}^* - \mathbf{X}_{A\text{sub}}^0\right) \le F_A^0 \tag{5}$$

$$F_B(\mathbf{X}_{A\text{sub}}^*) = F_B^0 + \left.\frac{\mathrm{d}F_B}{\mathrm{d}\mathbf{X}_{A\text{sub}}}\right|_{\mathbf{X}_{A\text{sub}}^0} * \left(\mathbf{X}_{A\text{sub}}^* - \mathbf{X}_{A\text{sub}}^0\right) \le F_B^0 \tag{6}$$

With regards to the design variables in subspace B, we obtain the following extensions to Eq. (4):

$$F_B(\mathbf{X}_{B\text{sub}}^*) = F_B^0 + \left.\frac{\mathrm{d}F_B}{\mathrm{d}\mathbf{X}_{B\text{sub}}}\right|_{\mathbf{X}_{B\text{sub}}^0} * \left(\mathbf{X}_{B\text{sub}}^* - \mathbf{X}_{B\text{sub}}^0\right) \le F_B^0 \tag{7}$$

$$F_A(\mathbf{X}_{B\text{sub}}^*) = F_A^0 + \left.\frac{\mathrm{d}F_A}{\mathrm{d}\mathbf{X}_{B\text{sub}}}\right|_{\mathbf{X}_{B\text{sub}}^0} * \left(\mathbf{X}_{B\text{sub}}^* - \mathbf{X}_{B\text{sub}}^0\right) \le F_A^0 \tag{8}$$

In Eqs. (5–8), $\mathbf{X}_{A\text{sub}}^*$ and $\mathbf{X}_{B\text{sub}}^*$ are the design variable values allocated to subspaces A and B, respectively. Hence, it follows, given that Eqs. (5–9) hold for the allocated variables and both functions, that the following relations hold for the set of system variables:

$$F_A(\mathbf{X}_{\text{SYS}}^*) \le F_A^0 \quad \text{and} \quad F_B(\mathbf{X}_{\text{SYS}}^*) \le F_B^0 \tag{9}$$

where $\mathbf{X}_{\text{SYS}}^*$ is the vector containing the new design variable values from both subspaces.

Hence, in the MOPCSSO formulation, the subspace optimization phase improves all objective functions simultaneously or at least improves any individual objective function without worsening any other. Therefore, a Pareto optimum will be obtained at the successful conclusion of the optimal design process. Further, given the fact that it retains much of the behavior of the constraint method, it has the ability to find any Pareto point on both convex and nonconvex portions of the Pareto frontier. Moreover, as mentioned earlier, a Pareto optimum can be theoretically verified by the KKT conditions. In MOPCSSO, the subspace optimizations are performed concurrently and independently with respect to a disjointed subset of design variables. Hence, the application of the KKT conditions toward the MOPCSSO method will hold. One can note that each subspace optimization in MOPCSSO is similar in concept to the constraint method [37]. Therefore, the KKT conditions, as applied in the constraint method, can be carried out in each subspace in MOPCSSO. The KKT condition for the kth subspace can be represented, for our purpose, as

$$
\begin{aligned}
&\nabla F_K(\mathbf{X}^*_{K\text{sub}}) + \sum_{\substack{P=1\\P\neq K}}^{NSS} \lambda_{FP}\nabla\Big[F_P(\mathbf{X}_{K\text{sub}}) - F_P^0\Big]\\
&\quad + \sum_{P=1}^{NSS} \lambda_{CP}\nabla C_P(\mathbf{X}^*_{K\text{sub}}) = 0\\
&\lambda_{FP}\nabla\Big[F_P(\mathbf{X}^*_{K\text{sub}}) - F_P^0\Big] = 0 \qquad P\neq K\\
&\lambda_{CP}\nabla C_P(\mathbf{X}^*_{K\text{sub}}) = 0\\
&\lambda_{FP} > 0, \qquad \lambda_{CP}\geq 0, \qquad \mathbf{X}_{K\text{sub}}\in S
\end{aligned}
\tag{10}
$$

where NSS is the number of subspaces, $\mathbf{X}_{K\text{sub}}$ represents the design variables allocated in the Kth subspace, and S represents the feasible region.

The MOPCSSO method presented here will yield Pareto solutions, is capable of capturing any Pareto point on both the convex and nonconvex Pareto frontier, enables each disciplinary group to have substantial autonomy through implementation of their own objectives and constraints, and requires no a priori knowledge. However, the final Pareto solutions obtained by MOPCSSO are dependent upon the initial design variable values [and, by extension, objective function values, as shown in Eq. (9)]. To address the issue of ensuring an evenly distributed Pareto frontier, together with an assurance of capturing the entire frontier, two additional developments to the method are made. First, we show that it is easy, using the MOPCSSO framework, to obtain endpoints of the Pareto frontier. Then, we present an approach for partitioning the frontier so as to force Pareto solutions to fall within prescribed performance regions, thereby ensuring a reasonably even distribution of points along at least one function.

Pareto Frontier Endpoint Determination

A completely represented Pareto frontier is one that is neither under- nor overrepresented. To produce a frontier of this nature, it is required to have the entire range of the objectives known, thus requiring the identification of the Pareto frontier endpoints. Recall that the MOPCSSO subspace optimizations contain disjointed subsets of design variables, with the effect from the other subspaces represented in the form of the auxiliary objective function constraints yielded from other subspaces. For the biobjective example previously discussed, the auxiliary constraint in subspace A requires that an improvement (or no worsening) in objective function B must be attained during each design cycle. A similar requirement exists in subspace B. To identify the endpoints of each subspace, however, it is not necessary to force the auxiliary objective function constraints to be satisfied. It is only necessary to identify, independent from the other objective functions, the minimum value corresponding to the objective function under investigation. Hence, one would implement the following subspace optimizations to identify the endpoints, which corresponds logically to the minimum values of each separate objective function. The next issue pertains to identifying the corresponding maximum values.

$$
\begin{array}{ll|ll}
\multicolumn{2}{c|}{\text{Subspace A}} & \multicolumn{2}{c}{\text{Subspace B}}\\
\min & F_A(\mathbf{X}_{A\text{sub}}) & \min & F_B(\mathbf{X}_{B\text{sub}})\\
\text{s.t.} & C_A(\mathbf{X}_{A\text{sub}}) \leq C_A^0\Big(1-r_A^a\Big) & \text{s.t.} & C_A(\mathbf{X}_{B\text{sub}}) \leq C_A^0\Big(1-r_B^a\Big)\\
& C_B(\mathbf{X}_{A\text{sub}}) \leq C_B^0\Big(1-r_A^b\Big) & & C_B(\mathbf{X}_{B\text{sub}}) \leq C_B^0\Big(1-r_B^b\Big)
\end{array}
\tag{11}
$$

Locating the maximum values for each individual objective function simply means locating the maximum for each individual objective that corresponds to the minimum for some other objective (given the competing nature of the objectives). Figure 4 shows this concept quite clearly for a biobjective problem.

For a problem with more subspaces (i.e., objectives), a simple extension is made, as is done in the normalized normal constraint method [31]. Consider a three-objective optimization problem as shown in Fig. 5. In each individual subspace optimization, the

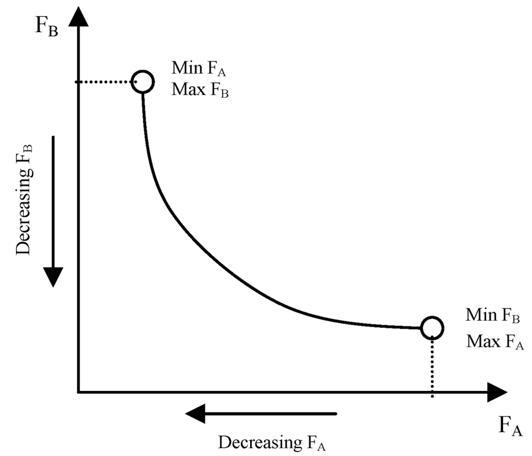


Fig. 4 Two-objective endpoint identification.

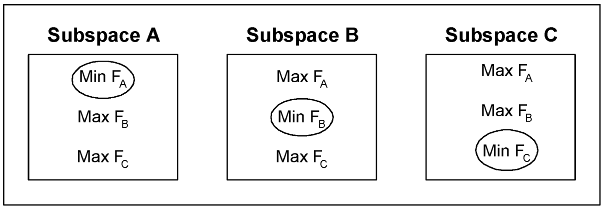


Fig. 5 Three-objective generic endpoint identification.

optimization algorithm locates the local minimum objective function value (i.e., min F_A in subpace A, min F_B in subspace B, and min F_C in subspace C). Within subspace A, the minimization will yield associated maximum values of F_B and F_C. However, another maximum value of F_B is also produced from the subspace C minimization, while a second maximum value of F_C is yielded from the subspace B minimization. Only one of the maximum F_Bs will be the true maximum for that particular function. It then becomes a simple approach to compare the two and choose that which is greater for the true maximum F_B (with a similar comparison to choose F_C). This approach is easily extensible to an arbitrary number of subspaces (i.e., functions).

Ensuring a Distribution of the Pareto Frontier

Recall that the Pareto frontier is the set of all candidate solutions that are deemed optimal for a particular design problem. Therefore, it is necessary to allow the designers to have the most complete frontier available from which to subsequently select the final design that will be developed into the product. With the Pareto endpoints easily identified using MOPCSSO, it is now possible to produce a Pareto frontier that is neither over- nor underrepresented.

Previous work suggests that an evenly distributed Pareto frontier is achievable through implementation of constraints during the process to limit the ranges of the objective functions [29]. In Huang and Bloebaum [38], the multi-objective range and target concurrent subspace optimization (MORTCSSO) method is introduced which allows for designer's preferences in the form of objective function ranges and targets to be taken into account during the subspace optimizations. The MORTCSSO method is formulated with its foundation similar to that of goal programming (GP). Essentially, GP accounts for the addition of the designer's preferences through imposing an additional constraint into the optimization problem. Upon specifying a certain value the designer considers desirable (i.e., the goal), the objective function will then attempt to minimize the deviations between the goal and the specific objective function. Mathematically, this can be expressed as Eq. (12), where T_i represents the specific goal set by the designer for the ith objective function. In this research, ensuring the even distribution of Pareto solutions is accomplished using a similar approach to Huang and Bloebaum's MORTCSSO development, which was in turn based on GP.

$$\text{Minimize} \sum_{i=1}^{k} |f_i(X) - T_i| \tag{12}$$

Consider the biobjective problem previously considered. At the start of the evenly distributed range technique, it must be decided if the range will be set on objective function A or objective function B. If the range is to be set on objective function A, the final objective function value for objective A will lie within the prescribed range. In this research, selection of one objective function over the other is arbitrary. In practice, selection of the objective function to be more evenly distributed may be based on designer preference.

If the distribution is applied to objective function FA, the overall range can be represented as range$FA = |\max FA - \min FA|$. Let us then say that we will require a distribution of points to fall within five different regions within this overall range (i.e., $n = 5$). Therefore, with the overall range on objective function A specified, as well as the number of discrete regions stated, an expression for ε is derived as shown in Eq. (13).

$$\varepsilon = \frac{\text{range}FA}{n} \tag{13}$$

As the optimization algorithm is executed, the desired range is shifted based upon the ε value (as shown in Fig. 6). The optimization process continues until the entire region has been explored, thus forcing the MOPCSSO method code to locate a solution within the given range. In discretizing the space, a relatively evenly distributed Pareto frontier will result. Because we only require a resulting Pareto point to fall within the specified range, and do not require it to be minimum in the range, it is not perfectly distributed. Although this could easily be accomplished, this would require a substantially greater amount of computational effort, with little resulting benefit. It would be easier to discretize the desired function into smaller increments, rather than forcing the approach to go to the minimum of a region.

Extending this concept to a three-objective case is relatively straightforward and it can then show how the approach is generalized for an arbitrary number of subspaces (i.e., objectives). Huang and Bloebaum demonstrate that for MOPs with more than two objective functions, it is possible to place a desired range on more than one of the objective functions using MORTCSSO [38]. For a three-objective case, let us say that objective B and objective C are chosen as the objectives on which ranges are set. With the location of the endpoints (minimums and maximums), the overall ranges on objective B and objective C can be identified. In addition, the number of discretizations n for each of the subspaces is also specified, yielding ε_B and ε_C in a similar fashion as was done in Eq. (13).

With these ranges set, it is clear that objective function A does not have any restrictions on where its minimum should lie. The way in which the optimization process will be executed is similar to populating an array, as in Eq. (14), in which the size of the array is a function of the number of discretizations, n_B and n_C.

$$\begin{array}{cc} & F_c \\ F_B & \begin{bmatrix} F_{B1}F_{C1} & F_{B1}F_{C2} & \cdots & F_{B1}F_{C_{nC}} \\ F_{B2}F_{C1} & F_{B2}F_{C2} & \cdots & F_{B2}F_{C_{nC}} \\ \vdots & \vdots & \ddots & \vdots \\ F_{B_{nB}}F_{C1} & F_{B_{nB}}F_{C2} & \cdots & F_{B_{nB}}F_{C_{nC}} \end{bmatrix} \end{array} \tag{14}$$

The subscripts F_{B1} and F_{C1}, as an example, denote the appropriate range that is being executed. For $n = 10$, therefore, the first row and first column position in Eq. (14) denotes the first 10% range on objective function B and the first 10% range on objective function C. The matrix is populated across the rows until all ranges of objective function C have been explored. At this point, objective function B is then shifted to its next range, F_{B2}, and the process continues. The matrix represented in Eq. (14) is populated for each range increment on objective functions B and C, while F_A is minimized with no restraint. This concept can be easily applied to problems with more than three subspaces (i.e., objective functions). In the next section, three MDO test problems are used to validate all aspects of the MOPCSSO approach presented in this work.

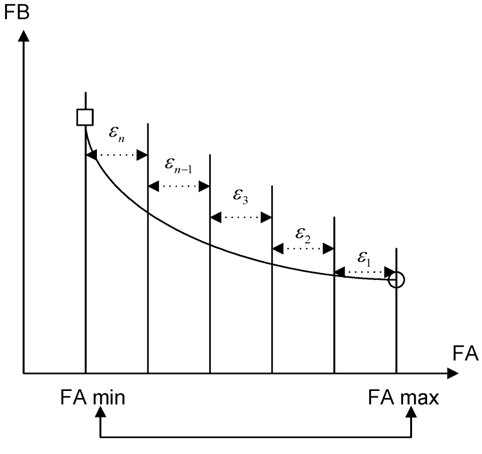


Fig. 6 Range distribution illustration for biobjective problem.

MOPCSSO Validation

Test Problem Description

In the MDO field, it is recognized that there are no large-scale practical engineering test problems (i.e., "real-life" problems) that can be easily identified and used for method testing due to the inherent complexity of such problems. This has been a recognized problem in the field since the mid-1980s. Any real-life test problems that exist are extremely small in size and are therefore not suitable to use for testing of a large-scale MDO method. The NASA Langley Research Center MDO Test Suite [39] has classified CASCADE [40] as a generator of class I test problems, which are useful for testing and evaluating new MDO methods. Three test problems generated by CASCADE are used to verify the validity of the three initial CSSO modifications made in preparation for the full MOPCSSO development. To demonstrate the feasibility of these modifications, 54 randomly generated initial points are investigated for test problems 1, 2, and 3, respectively. The initial points are diversely generated to broadly cover the objective function space.

It should also be recognized that the solution validation for multi-objective problems has continued to be an extremely difficult issue for all researchers. To date, the most practical way to achieve validation is to use visualization to verify the obtained results (for up to three objective functions). Here, the modified method of feasible direction (MMFD) [41] is used as the optimizer within each subspace, the weighted sum approach [22] is used to create the Pareto frontier for each test problem, and regression techniques (built in the SIGMAPLOT$^{\text{TM}}$) are subsequently used to approximate the complete Pareto frontier for all test problems. It can be noted that constructing the approximated Pareto frontiers for test problems 1 and 2 is easy, because there are only two objectives. However, for test problem 3, which has three objectives, close to 400 different combinations of weighting parameters (i.e., $w_1 + w_2 + w_3 = 1.0$, $w_1, w_2, w_3 \geq 0$) were implemented to construct a more accurate Pareto frontier, as opposed to only 15 combinations for problems 1 and 2. These approximated Pareto frontiers are used as a basis to compare the results obtained by the proposed method in this research. All results can be easily visualized in the performance space. To save space in this paper, the semantics of these test problems have been omitted. All details for these problems can be found in Huang's dissertation [42].

Test problem 1 consists of 2 objective functions, 10 design variables, 4 behavior variables, and 10 constraints (Fig. 7); test problem 2 consists of 2 objective functions, 20 design variables, 4 behavior variables, and 15 constraints (Fig. 8); and test problem 3 consists of 3 objective functions, 18 design variables, 6 behavior variables, and 14 constraints (Fig. 9). The goal in selecting these test

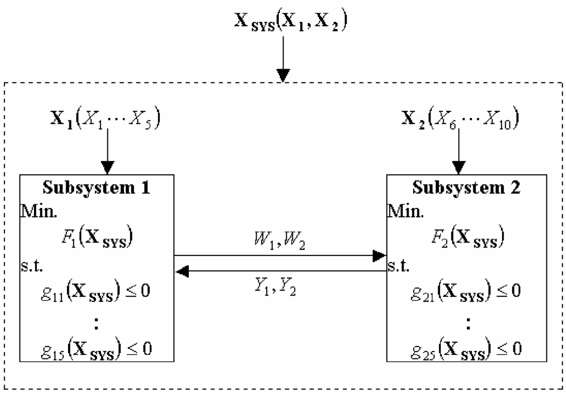


Fig. 7 Representation of test problem 1.

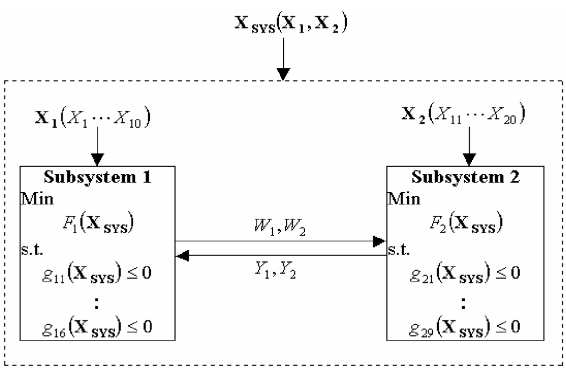


Fig. 8 Representation of test problem 2.

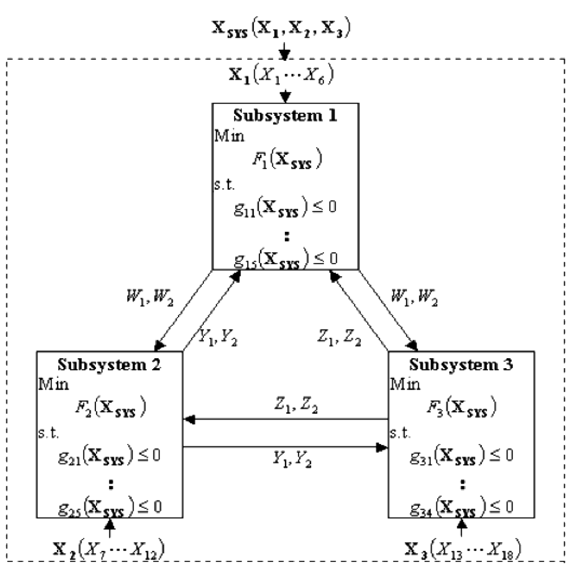


Fig. 9 Representation of test problem 3.

problems was to investigate differing sizes of problems, with varying numbers of design variables and constraints within disciplines.

Validation Results

The validation procedure involves four separate investigations using the three test cases described previously: 1) the viability of the three modifications to CSSO; 2) the proposed MOPCSSO approach itself; 3) the identification of the endpoints; and 4) the ability to force a distribution in the Pareto frontier. The first set of results pertains to an implementation in which all individual subspace objective functions are combined into a single weighted objective so as to investigate the modifications pertaining to design variable allocation, constraint feasibility, and constraint parameter update. It is necessary to demonstrate the viability of these modifications before jumping into the implementation of the MOPCSSO method using these modifications. The second set of results demonstrates that MOPCSSO does indeed accurately result in Pareto points. The third set of results pertains to the issue of identifying endpoints of the Pareto frontier using MOPCSSO. Lastly, the fourth set of results shows that a simple modification of the subspace optimization objective function expressions in MOPCSSO can yield a reasonably evenly distributed Pareto frontier. These four sets of results are presented in the next four subsections.

Validation of CSSO Modifications

Here, the three initial modifications of the CSSO (i.e., responsibility coefficient update and design variable allocation procedures, new ρ determination procedures, and constraint minimization procedures) are implemented using a weighted sum approach to generate a system level objective function, with 54 initial points investigated for each test problem to determine the ability of the modified CSSO to find the Pareto frontier. The generated system level objective function used equal weights of 0.5 for each individual function (identified as obj1 and obj2) in test problems 1 and 2, and 0.3333 for each individual function (identified as obj1, obj2, and obj3) in test problem 3.

In Figs. 10a and 11a, the curve represents the approximated Pareto frontier, the circles represent all the initial points, and the triangles represent the optimal Pareto values corresponding to test problems 1 and 2, respectively. Figures 10b and 11b show the final results of the objective function values with respect to each initial point for both test problems. Diamonds and squares represent the optimal values corresponding to the individual objective functions, $F_1(\mathrm{X})$ and $F_2(\mathrm{X})$, and triangles represent a composite objective function, $F_1(X) + F_2(X)$. This composite objective function, identified in the figure as objall, is not the one being minimized in each subspace in this initial validation procedure (in which weights were each 0.5). This composite function was selected to visually demonstrate the convergence characteristics of the approach across all initial points.

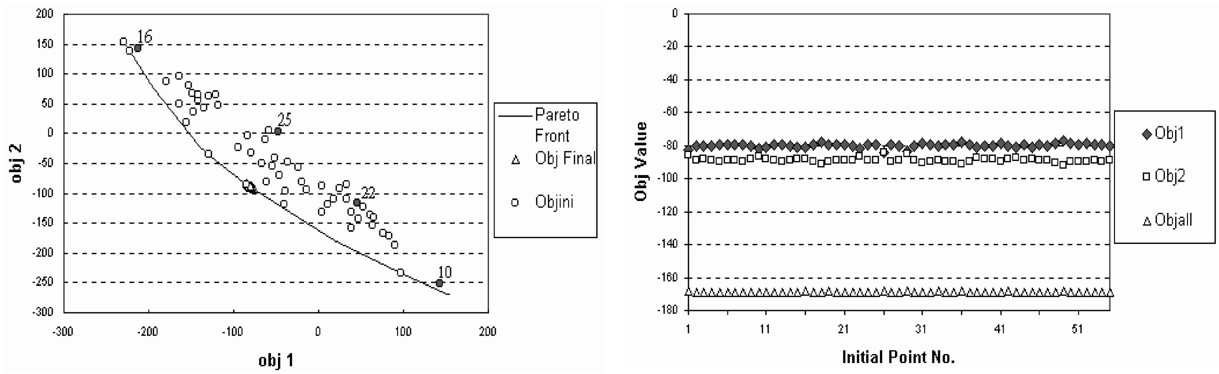
In Fig. 12a, the initial points (cubes) and the final results (spheres) are displayed in the performance space. The surface represents the approximated Pareto frontier. The final objective function results related to each initial point are plotted in Fig. 12b. Diamonds, squares, circles, and triangles represent the optimum values with respect to individual objective function $F_1(\mathrm{X})$, $F_2(\mathrm{X})$, and $F_3(\mathrm{X})$ (identified for simplicity on the figure as obj1, obj2, and obj3), and to the composite objective function, $F_1(\mathrm{X}) + F_2(\mathrm{X}) + F_3(\mathrm{X})$.

Figures 13a–13c represent the objective function value versus cycle number corresponding to an arbitrarily chosen initial point (i.e., initial point 25 for test problem 1 and initial point 2 for test problem 2, and initial point 17 for problem 3).

From Figs. 10a, 11a, and 12a, we see that regardless of the starting point, the method consistently converges to the same final optimal result. In Figs. 10b, 11b, and 12b, the final results of the system level objective function associated with each initial point converged to within 2%. Even though very slight variations might exist in each of the final individual objective function values, the composite function is accurate for all points tested. Figures 13a–13c demonstrate that the composite function decreases throughout the optimization procedure, while the individual objective functions make appropriate tradeoffs in a consistent manner. These figures clearly demonstrate that the modifications implemented to prepare CSSO for a full-fledged MOP application are viable and provide for a consistent and reliable solution process.

Validation of MOPCSSO with Subspace Objectives

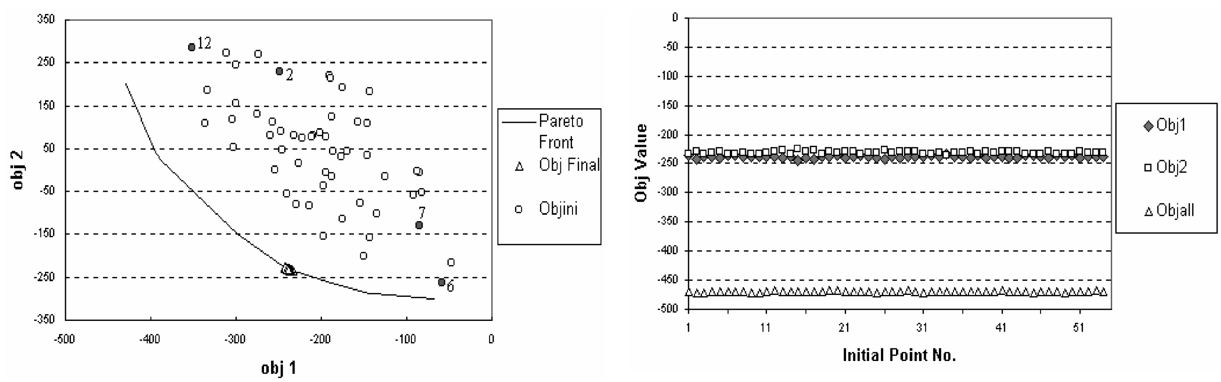
Now that the three modifications have been demonstrated to be viable, the full-fledged MOPCSSO method is implemented for the three test problems. For test problem 1, 55 initial points were randomly generated. For test problem 2, 54 randomly generated initial points were used, while test problem 3 had an associated 58 initial randomly generated points.



a) Initial and final objective values

b) Final objective values for each initial point

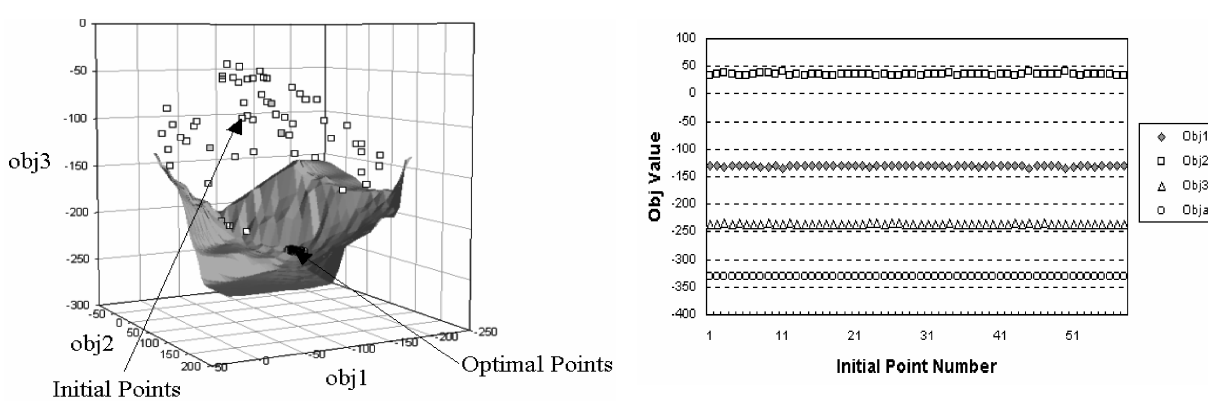
Fig. 10 Test problem 1 results.



a) initial and final objective values

b) Final objective values for each initial point

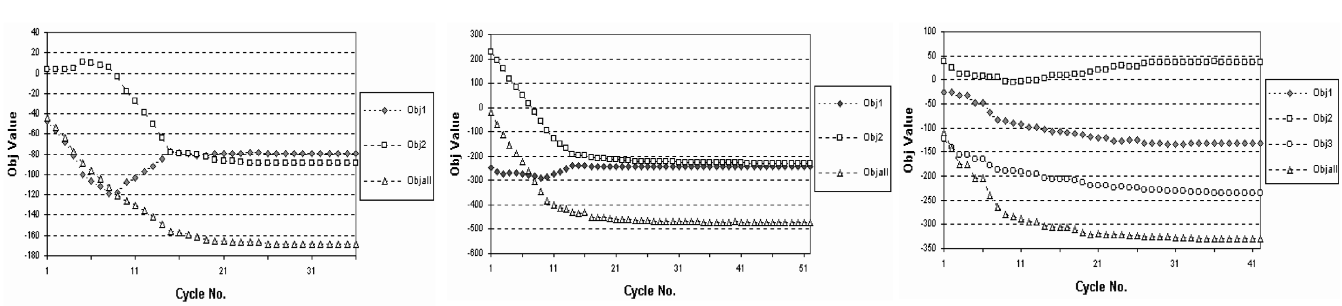
Fig. 11 Test problem 2 results.



a) Initial and final objective values

b) Final objective values for each initial point

Fig. 12 Test problem 3 results.



a) Problem 1

b) Problem 2

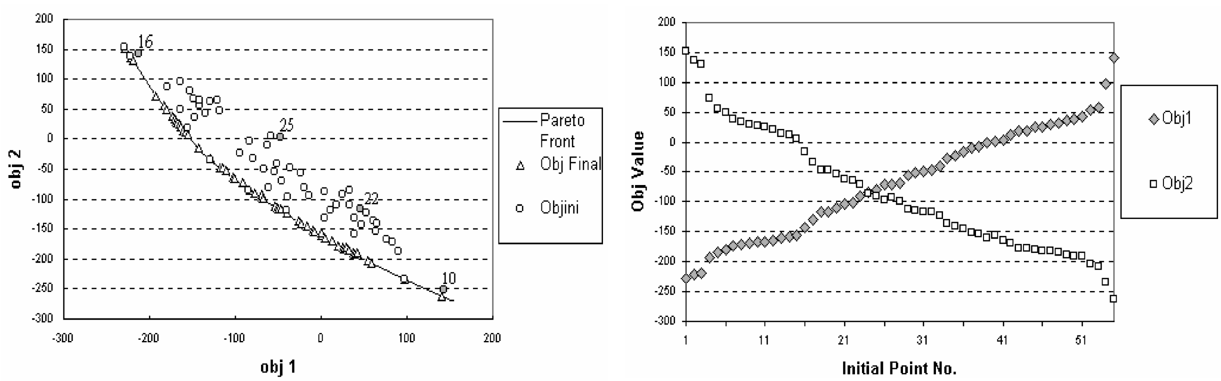
c) Problem 3

Fig. 13 Convergence results.

In Figs. 14a and 15a, the curve represents the approximated Pareto frontier, the circles represent all the initial points, and the triangles represent the Pareto optimal values corresponding to test problems 1 and 2, respectively. Sorted by increasing order of obj1 value, Figs. 14b and 15b show the final results of the objective function values with respect to each initial point related to test problems 1 and 2, respectively. Diamonds and squares represent the optimal values corresponding to the individual objective functions, obj1 and obj2. Figures 17a and 17b represent the objective function value versus cycle number corresponding to an arbitrarily chosen initial point (i.e., initial point 25 for test problem 1 and initial point 2 for test problem 2).

In Fig. 16a, the initial points (cubes) and the final results (spheres) are displayed in the objective function space. The surface represents the approximated Pareto frontier. The top view of the final results related to each initial point is plotted in Fig. 16b. Figure 17c represents the objective function value versus cycle number corresponding to initial point 17. Diamonds, squares, circles, and triangles represent the optimum values with respect to individual objective function obj1, obj2, and obj3, and to the composite objective function, objall (obj1 + obj2 + obj3), respectively. Recall that the composite function is used here strictly as a convenience so that the convergence characteristics of the MOPCSSO method are more readily apparent.

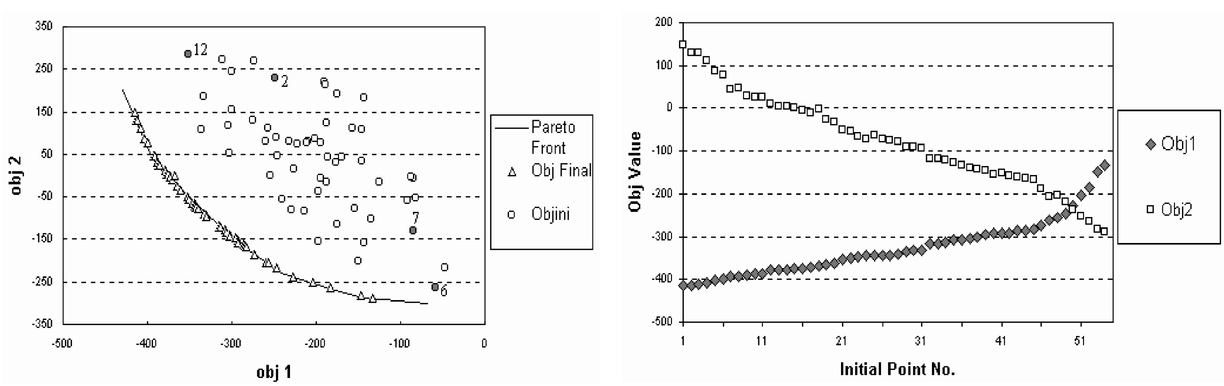
It is clear from Figs. 14a, 15a, and 16a that the MOPCSSO was very effective in finding solutions along the Pareto frontier. Further, from Figs. 14b and 15b, we see that as one objective function decreases in value, the other increases, in typical Pareto behavior for competing objectives. Furthermore, as seen in Fig. 17, the individual objective functions improved simultaneously or at least without



a) Initial and final objective values

b) Final objective values for each initial point

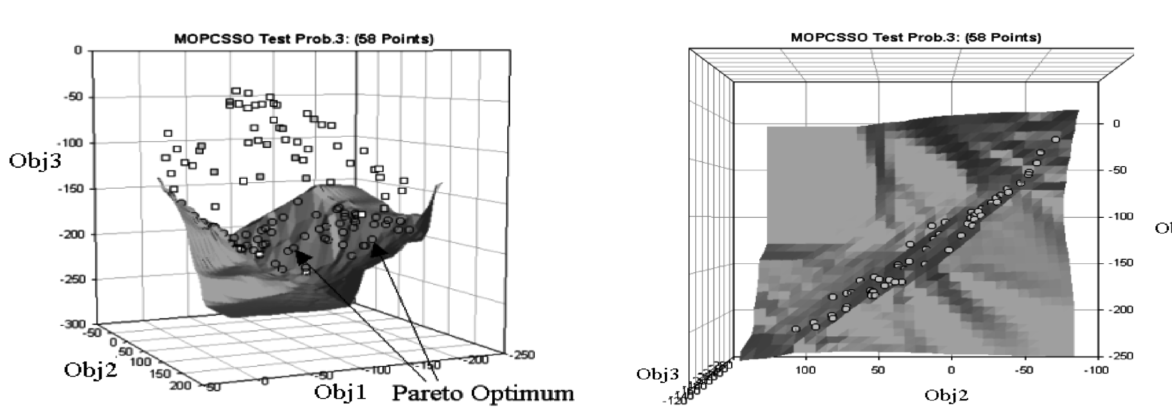
Fig. 14 Test problem 1 result of MOPCSSO.



a) Initial and final objective values

b) Final objective values for each inital point

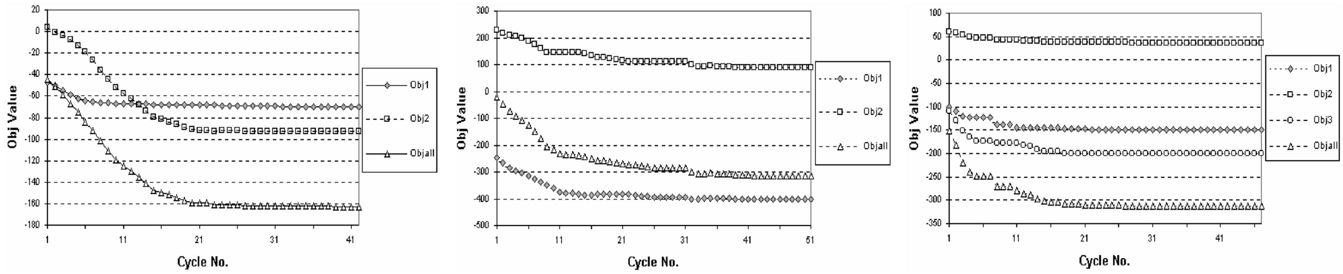
Fig. 15 Test problem 2 result of MOPCSSO



a) Initial and final objective values

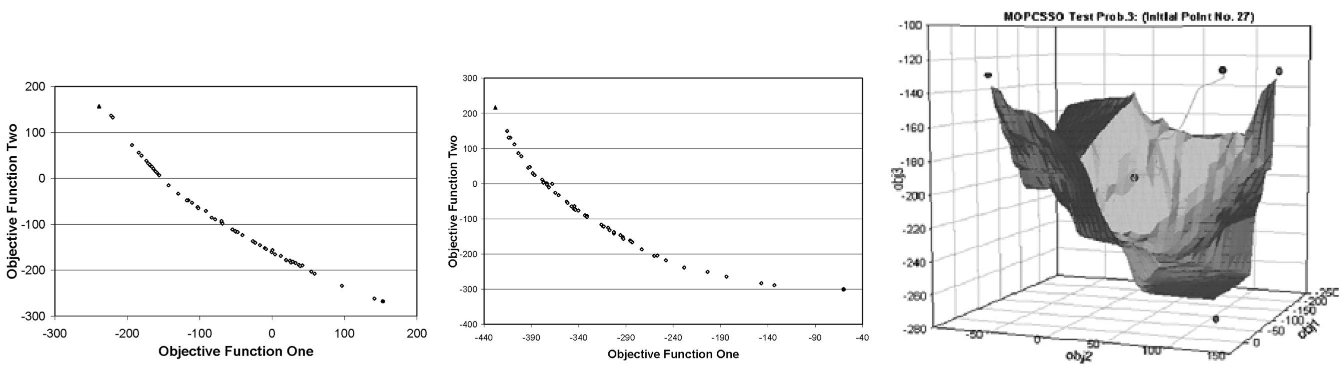
b) Final objective values for each initial point

Fig. 16 Test problem 3 result of stage MOPCSSO



a) Problem 1 (for initial pt. 25) b) Problem 2 (for initial pt. 2) c) Problem 3 (for initial pt. 17)

Fig. 17 Convergence results.



a) Test problem 1 **b) Test problem 2** **c) Test problem 3**

Fig. 18 Pareto frontiers with endpoints.

worsening the others during the optimization process. This is accomplished with the auxiliary constraints introduced from other subspaces' objective functions. These auxiliary constraints ensure that each subspace optimization considers the interests of the other subspaces while improving its own objective function. The optimal design process will stop at the point where no further improvement is possible for any objective, which means that a Pareto optimum is found.

For all the test problems, convergence of all objective functions can be seen to be consistent, with the Pareto frontier being reached in every case. Further, test problem 3 clearly shows that multiple subspaces (i.e., more than 2) can be easily handled by the MOPCSSO method. Recall that with MOPCSSO, the negotiation between the various subspace optimizations occurs automatically due to the representation of the other subspace objectives and constraints within each subspace optimization.

One drawback, as previously mentioned, is the randomness in the resulting Pareto solution, depending on the initial solution. Although it is true that initial points scattered across the performance space will yield Pareto points across the Pareto frontier, there is no guarantee of capturing the entire Pareto frontier. For this reason, two simple applications are explored in the next two subsections. The first demonstrates the ease of capturing the endpoints of the Pareto frontier using MOPCSSO while the second addresses the issue of achieving a more uniform distribution of Pareto points.

Endpoint Validation

The approach previously described for finding endpoints was used for all three test problems. Figure 18 shows the endpoints superimposed on the Pareto frontier for all three test problems. In Fig. 18c, there is also a point with a path leading down to the Pareto frontier, for the purpose of demonstrating the path the MOPCSSO method took for one random starting point. From these figures, we see that while MOPCSSO managed to capture most of the Pareto frontier in each case, merely through launching MOPCSSO at different initial design points, it is obvious that the entire frontier was not represented. With this simple modification, therefore, endpoints can be easily found. With this information for each objective, the desire for a more even distribution can also be addressed.

Achieving a Distributed Pareto Frontier

Here, the MOPCSSO algorithm was run for a range size of 10% of the overall range, for simplicity. This percentage is a variable that can be set to whatever is desirable to the designer. Objective function 1 was chosen to have the range restriction imposed upon it, although similar results were obtained when imposed on objective function 2. Test points 34 for test problem 1 and 17 for test problem 2 were used to demonstrate the distribution process. The vertical lines in Fig. 19 and horizontal lines in Fig. 20 are included to show where the 10% intervals fall for these problems. It is clear that the imposed range on the objectives results in a distribution of Pareto points across the entire frontier. Recall that the approach used in this work only requires finding a Pareto point for which the range is satisfied—not one for which the minimum of that function is found within the imposed range. The reality is that generating more points within each range will automatically provide a more complete Pareto frontier, as would the expedient of requiring a finer distribution of points. It is felt

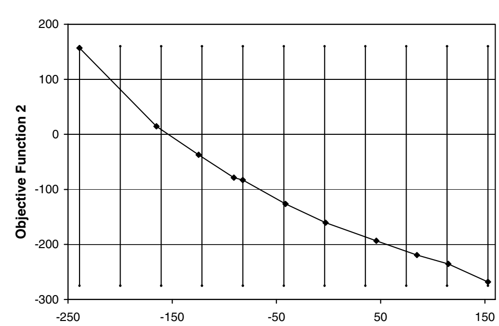


Fig. 19 Test problem 1—range size of 10% of overall $F1$ range, initial point 34.

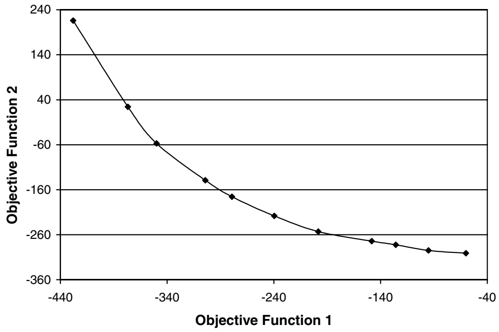


Fig. 20 Test problem 2—range size of 10% of overall $F1$ range, initial point 17.

that both of these solutions would have more value than wasting computational effort to force a Pareto point to a particular target on the Pareto frontier.

Conclusions

In this paper, we presented a new approach for solving multi-objective optimization problems in an MDO context, wherein large-scale multidisciplinary optimization problems are commonplace. The results presented here clearly demonstrate the MOPCSSO method is a feasible and usable optimization approach that has many potential advantages. By means of the MOPCSSO method, disparate disciplinary design teams are able to handle the design with their own criteria (i.e., disciplinary objective, constraints, etc.) while exercising the tradeoffs that exist between their different interests (accomplished by introducing the auxiliary constraints into each subspace). This enables tremendous flexibility, as each design team can use the analysis and optimization methods most appropriate to their own discipline within the MOPCSSO structure. Also, in MOPCSSO, no a priori knowledge is required to obtain the Pareto results. The Pareto optimum obtained by MOPCSSO simply depends on the initial design (i.e., the initial value of the design variables and, therefore, objective functions). The initial design provided by designers can be regarded as a baseline for the final solution. By definition, in MOPCSSO, if the initial design is feasible and a solution exists, the final value of objective functions should be less to or equal to the initial design. Additionally, with MOPCSSO's ability to handle individual objectives simultaneously, it eliminates the need to formulate an aggregate objective function through the weighting parameters. Further, it was demonstrated that MOPCSSO can very easily be used to obtain the endpoints of the Pareto frontier and that a simple modification in the subspace optimization objectives can be used to obtain a more evenly distributed Pareto frontier. These characteristics distinguish MOPCSSO from other existing MOP methods.

Recall that in the formulation of MOPCSSO, each individual objective function is considered equally important. Because there is no accepted definition of global optimum in multi-objective optimization, the choice about what the best solution is ultimately depends on human decision making and preferences. Therefore, a key research issue in future work pertains to obtaining Pareto results that directly satisfy designer's preferences within the MOPCSSO framework.

Acknowledgments

Support of this work by the New York State Center for Engineering Design and Industrial Innovation (NYSCEDII) and, by extension, the State of New York, is gratefully acknowledged. We also gratefully acknowledge the valuable insights of Kemper Lewis and Timothy Simpson during the development of MOPCSSO.

References

[1] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization*, Vol. 14, No. 1, Aug. 1997, pp. 1–23.

[2] Sobiesezczanski-Sobieski, J., "On the Sensitivity of Complex, Internally Coupled Systems," *AIAA Journal*, Vol. 28, No. 1, 1990, pp. 153–160.

[3] Hajela, P., Bloebaum, C. L., and Sobieszczanski-Sobieski, J., "Application of Global Sensitivity Equations in Multidisciplinary Aircraft Synthesis," *Journal of Aircraft*, Vol. 27, No. 12, 1990, pp. 1002–1010.

[4] Sobieszczanski-Sobieski, J., Bloebaum, C. L., and Hajela, P., "Sensitivity of Control-Augmented Structure Obtained by a System Decomposition Method," *AIAA Journal*, Vol. 29, No. 2, 1991, pp. 264–270.

[5] Cramer, E. J., Frank, P. D., Shubin, G. R., Dennis, J. E., and Lewis, R. M., "On Alternative Problem Formulations for Multidisciplinary Optimization," AIAA Paper 92-4752, Sept. 1992.

[6] Cramer, E. J., Dennis, J. E., Frank, P. D., Lewis, R. M., and Shubin, G. R., "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.

[7] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," AIAA Paper 94-4325, Sept. 1994.

[8] Braun, R. D., Kroo, I. M., and Moore, A. A., "Use of the Collaborative Optimization Architecture for Launch Vehicle Design," AIAA Paper 96-4018, Sept. 1996.

[9] Sobiesezczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems," *Recent Advances in Multidisciplinary Analysis and Optimization*, proceedings of a symposium cosponsored by NASA Langley Research Center, NASA Lewis Research Center, and the Wright Research Development Center, National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Division, Washington, D. C., 1989, OCLC: 22481989.

[10] Bloebaum, C. L., Hajela, P., and Sobieszczanski-Sobieski, J., "Non-Hierarchic System Decomposition in Structural Optimization," *Engineering optimization*, Vol. 19, No. 3, 1992, pp. 171–186.

[11] Sobieszczanski-Sobieski, J., Agte, J., and Sandusky, R., "Bilevel Integrated System Synthesis," *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 164–172.

[12] Alexandrov, N. M., and Kodiyalam, S., "Initial Results of an MDO Method Evaluation Study," AIAA Paper 98-4884, Sept. 1998.

[13] Alexandrov, N. M., and Lewis, R. M., "Analytical and Computational Properties of Distributed Approaches to MDO," AIAA Paper 2000-4718, Sept. 2000.

[14] Pareto, V. (1909), *Manual of Political Economy*, English translation by A. S. Schwier, Augustus M. Kelley, New York, 1971.

[15] Kuhn, H. W., and Tucker, A., "Nonlinear Programming," *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 1951.

[16] Serafini, P., "Simulated Annealing for Multiple Objective Optimization Problems," *Multiple Criteria Decision Making: Proceedings of the 10th International Conference: Expand and Enrich the Domains of Thinking and Application*, edited by G. H. Tzeng, Springer–Verlag, New York, 1994, ISBN 0387942971 9780387942971, OCLC 30027483.

[17] Ulungu, E. L., Teghem, J., Fortemps, P., and Tuyttens, D., "MOSA Method: A Tool for Solving Multi-Objective Combinatorial Optimization Problems," *Journal of Multi Criteria Decision Analysis*, Vol. 8, No. 4, 1999, pp. 221–236.

[18] Coello, C. A. C., "A Comprehensive Survey of Evolutionary-Based Multi-Objective Optimization," *Knowledge and Information Systems*, Vol. 1, No. 3, 1999, pp. 269–308.

[19] Coello, C. A. C., "An Updated Survey of GA-Based Multi-Objective Optimization Techniques," *ACM Computing Surveys*, Vol. 32, No. 2, 2000, pp. 109–143.

[20] Fonseca, C. M., and Fleming, P. J., "Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalization," *Proceedings of the 5th International Conference on Genetic Algorithms*, edited by S. Forrest, Morgan Kaufmann Publishers, San Mateo, CA, July 1993, ISBN 1558602992 9781558602991, OCLC 39373759.

[21] Zitzler, E., "Evolutionary Algorithms for Multi-Objective Optimization: Methods and Applications," Ph.D. Dissertation, Department of Computer Science, Swiss Federal Institute of Technology (ETH), Aachen, Shaker Verlag, Germany, 1999.

[22] Das, I., and Dennis, J., "A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multi-Criteria Optimization Problems," *Structural Optimization*, Vol. 14, No. 1, 1997, pp. 63–69.

[23] Ignizio, J. P., *Goal Programming and Extensions*, Lexington Books, Lexington, MA, 1976.

[24] Ijiri, Y., *Management Goals and Accounting for Control*, North-Holland, Amsterdam, 1965.

[25] Chen, W., Wiecek, M., and Zhang, J., "Quality Utility: A Compromise Programming Approach to Robust Design," *Journal of Mechanical Design*, Vol. 121, No. 2, 1999, pp. 179–187.

[26] Hartley, J. R., *Concurrent Engineering. Shortening Lead Times, Raising Quality, and Lowering Costs*," Productivity Press, Cambridge, MA, 1992.

[27] Miettinen, K. M., *Nonlinear Multi-Objective Optimization*, Kluwer Academic Publishers, Boston, MA, 1998.

[28] Tappeta, R. V., Renaud, J. E., Messac, A., and Sundararaj, G. J., "Interactive Physical Programming: Tradeoff Analysis and Decision Making in Multi-Criteria Optimization," *AIAA Journal*, Vol. 38, No. 5, 2000, pp. 917–926.

[29] Messac, A., Ismail-Yahaya, A., and Mattson, C. A., "The Normalized Normal Constraint Method for Generating the Pareto Frontier," *Structural and Multidisciplinary Optimization*, Vol. 25, No. 2, 2003, pp. 86–98.

[30] McAllister, C., Simpson, L. T., Lewis, K., and Messac, A., "Robust Multiobjective Optimization Through Collaborative Optimization and Linear Physical Programming," AIAA Paper 2004-4549, Aug. 2004.

[31] Tappeta, R. V., and Renaud, J., "Multiobjective Collaborative Optimization," *Journal of Mechanical Design*, Vol. 119, No. 3, 1997, pp. 403–411.

[32] Gunawan, S., Azarm, S., Wu, J., and Boyars, A., "Quality Assisted Multi-Objective Multi-Disciplinary Genetic Algorithms," *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, Sept. 2002.

[33] Tappeta, R. V., Renaud, J. E., and Rodriguez, J. F., "An Interactive Multiobjective Optimization Design Strategy for Decision Based Multidisciplinary Design," *Journal of Engineering Optimization*, Vol. 34, No. 5, 2002, pp. 523–544.

[34] Andersson, J., "A Survey of Multi-Objective Optimization in Engineering Design," LiTHIKP TR R-1097, Department of Mechanical Engineering, Linköping University, Linköping, Sweden, 2000.

[35] Renaud, J. E., and Gabriele, G. A., "Improved Coordination in Non-Hierarchic System Optimization," *AIAA Journal*, Vol. 31, No. 12, 1993, pp. 2367–2373.

[36] Hajela, P., "Techniques in Optimum Structural Synthesis with Static and Dynamic Constraints," Ph.D. Dissertation, Department of Mechanical Engineering, Stanford University, Stanford, CA, 1982.

[37] Messac, A., Gupta, S., and Akbulut, B., "Linear Physical Programming: A New Approach to Multiple Objective Optimization," *Transactions on Operational Research*, Vol. 8, No. 2, 1996, pp. 39–59.

[38] Huang, C.-H., and Bloebaum, C. L., "Incorporation of Preferences in Multi-Objective Concurrent Subspace Optimization," *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Reston, VA, Sept. 2004.

[39] Padula, S. L., Alexandrov, N., and Green, L. L., "MDO Test Suite at NASA Langley Research Center," AIAA Paper 96-4028, Sept. 1996.

[40] Hulme, K. F., and Bloebaum, C. L., "Development of a Multidisciplinary Design Optimization Test Simulator," *Structural Optimization*, Vol. 14, Nos. 2–3, 1997, pp. 129–137.

[41] Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: With Applications*, Vanderplaats Research & Development, Colorado Springs, CO, 1999.

[42] Huang, C.-H., "Development of Multi-Objective Concurrent Subspace Optimization and Visualization Methods for Multidisciplinary Design," Ph.D. Dissertation, University at Buffalo, Buffalo, NY, 2003.

N. Alexandrov
Associate Editor