AIAA JOURNAL
Vol. 46, No. 10, October 2008

Bilevel Adaptive Weighted Sum Method for Multidisciplinary Multi-Objective Optimization

Ke-shi Zhang,* Zhong-hua Han,† Wei-ji Li,‡ and Wen-ping Song§
Northwestern Polytechnical University, 710072 Xi'an, People's Republic of China

DOI: 10.2514/1.36853

The primary goal of this research is to develop a framework for dealing with multi-objective, multidisciplinary optimization problems with a large number of variables. The proposed method is expected to provide a relatively uniformly spaced, widely distributed Pareto front. To achieve this end, a novel integration of the adaptive weighted sum method within a concurrent subspace optimization framework is presented. In the bilevel framework of concurrent subspace optimization, the adaptive weighted sum is used to make tradeoffs among multiple, conflicting objectives. To obtain better distributed solutions, two modifications are made. First, an additional equality constraint in suboptimization for each expected solution is relaxed because it causes slow convergence within the bilevel optimization framework. The probability of entrapment in local minima can also be reduced. Second, the mesh of the Pareto front patches is modified due to the low efficiency of the original scheme. The proposed method is demonstrated with three multidisciplinary design optimization problems: 1) a numerical multidisciplinary design optimization test problem with a convex Pareto front, available within the NASA multidisciplinary design optimization Test Suite; 2) a test problem with a nonconvex Pareto front, which is not easily solved; and 3) a conceptual design of a subsonic passenger aircraft, which consists of two objectives, four design variables, five coupling behavior variables, seven constraints in aerodynamics, and weight discipline. The primary results show that the proposed method is promising with regard to obtaining a uniformly spaced, widely distributed, and smooth Pareto front and is applicable in the design of large-scale, complex engineering systems such as aircraft.

I. Introduction

IN RECENT years, the industry has paid more attention to improving efficiency in the design of complex systems, such as aircraft. Multidisciplinary design optimization (MDO) has emerged as an engineering discipline that focuses on the development of new design and optimization strategies for complex systems. MDO researchers strive to reduce the time and cost associated with the coupling interaction among several disciplines. As stated in [1]

> Decomposition approaches provide many advantages for the solution of complex MDO problems, as they enable a partitioning of a large coupled problem into smaller, more manageable 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 personal hours, because groups are no longer required to wait around for other groups in the process to complete their design tasks.

Concurrent subspace optimization (CSSO) is one of the main decomposition approaches in MDO. It supports a collaborative and distributed multidisciplinary design optimization environment among different disciplinary groups. Sobieszczanski-Sobieski first proposed the subspace optimization method in 1989 [2]; Sobieszczanski-Sobieski's blueprint was further developed by Bloebaum and subsequently named the concurrent subspace optimization method [3]. Renaud and Gabriele developed a second-order variant of the global sensitivity equation (GSE) method and an alternative potential coordination procedure for the CSSO method [4–6]. Sellar et al. proposed to replace GSE with the neutral-network-based response surface method [7].

In the response-surface-based CSSO (RSCSSO) method, the subspace optimizations generate sufficient design information for approximation. After the concurrent subspace optimizations, a global approximation problem is formulated about the current design vector using information stored in the design database. It is the coordination procedure of global approximation that drives constraint satisfaction and overall system optimization. A fast and robust convergence appears in the RSCSSO method. However, with an increase in design variables the sample points needed for creating the response surface model will be greatly increased. Hence, it is more suitable for problems with few design variables. In the original CSSO (GSE-based CSSO, or GSECSSO) method, 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 coordination parameters. 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. Constraint responsibility and tradeoff factors are used to coordinate suboptimizations. The updated design vector is the simple combination of local optimal design subvectors. This provides designers with a significant potential benefit in terms of computational effort. Although sometimes the convergence of GSECSSO is oscillatory and premature, it is proven in practice that both of these can be much suppressed by removing tradeoff coefficients and using a self-adaptive parameter for the Kreisselmeier–Steinhauser (KS) function [1].

The CSSO method was developed initially for a single-objective MDO problem. However, most MDO problems are multi-objective. In recent years, more work [8–14] has focused on extending the existing MDO method to handle such multi-objective MDO problems by means of integrating a multi-objective optimization

Presented as Paper 0908 at the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 7–10 January 2008; received 26 January 2008; revision received 4 May 2008; accepted for publication 15 June 2008.

*Associate Professor, Post Office Box 120, School of Aeronautics, Number 127 West Youyi Road; zhangkeshi@nwpu.edu.cn.

†Ph.D., Post Office Box 754, School of Aeronautics, Number 127 West Youyi Road; hanzh@nwpu.edu.cn.

‡Professor, Post Office Box 120, School of Aeronautics, Number 127 West Youyi Road; lwjnwpu@126.com.

§Professor, Post Office Box 754, School of Aeronautics, Number 127 West Youyi Road; wpsong@nwpu.edu.cn.

method within the MDO framework. This kind of method can be called a multi-objective MDO method.

Integrating the multi-objective optimization method within the CSSO framework is an effective way to develop the multi-objective MDO method. Huang and Bloebaum [1,13] and Parashar and Bloebaum [14] have extended CSSO to solve multi-objective MDO problems, including the multi-objective Pareto CSSO (MOPCSSO), the multi-objective range CSSO (MORCSSO), the multi-objective target CSSO (MOTCSSO), and the multi-objective genetic algorithm CSSO (MOGACSSO) methods. In the MOPCSSO method, the constraint method is integrated within the CSSO framework [1]. In the MORCSSO and MOTCSSO methods, the concept of designer preference is introduced [13]. In the MOGACSSO method, the genetic algorithm is combined with CSSO and in the hope of improving the computational efficiency [14].

The goal of this research is to develop a multi-objective CSSO method that will provide a relatively uniformly spaced, widely distributed Pareto front. A uniformly and widely distributed Pareto front is very helpful in the preliminary design, as it gives designers an overall understanding of all possible scenarios. As a multi-objective optimization method, the adaptive weighted sum (AWS) method is able to evenly distribute the Pareto front and overcome the drawbacks of the ordinary weighted sum method [15,16]. The AWS method is also able to generate any points in the concave region of the Pareto front. It is expected that a Pareto front of high quality will be acquired by integrating the AWS method into CSSO. In this paper, a novel integration of the AWS method within the CSSO framework is presented. The CSSO and AWS methods are briefly reviewed, followed by a description of the proposed AWS-based CSSO (AWSCSSO) method. Issues pertaining to the modification when AWS is applied in the bilevel framework are discussed. The AWSCSSO method is applied to a test problem with a convex Pareto front, a test problem with a nonconvex Pareto front, and an MDO problem of the conceptual design of a subsonic passenger aircraft. The results are discussed.

II. Concurrent Subspace Optimization and Adaptive Weighted Sum Background

A. Concurrent Subspace Optimization Method

A variant of the CSSO method described in [1] is used in this paper. The following MDO problem with two coupled subsystems is taken as an example:

$$\begin{aligned} &\text{Min } F(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) \\ &\text{s.t. } \mathbf{G}_1(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) \le \mathbf{0} \\ &\mathbf{G_2}(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) \le \mathbf{0} \\ &\mathbf{Y}_1 = f_1(\mathbf{X}, \mathbf{Y}_2) \\ &\mathbf{Y}_2 = f_2(\mathbf{X}, \mathbf{Y}_1) \end{aligned} \tag{1}$$

where $\mathbf{X}$ is the design variable vector, $\mathbf{Y}_i$ is the coupled variable vector of subsystem i, and $\mathbf{G}_i$ are constraints provided by the subsystem i. For Eq. (1), the mathematical models of subspace optimization for the modified CSSO can be written as follows:

$$\begin{aligned} &\text{Suboptimization 1} \\ &\text{Min } F(\mathbf{X}_1) \\ &\text{s.t. } C_1(\mathbf{X}_1) \le C_1^0(1 - r_1^1) \\ &\hat{C}_2(\mathbf{X}_1) \le C_2^0(1 - r_1^2) \\ &\text{Suboptimization 2} \\ &\text{Min } F(\mathbf{X}_2) \\ &\text{s.t. } C_1(\mathbf{X}_2) \le C_1^0(1 - r_2^1) \\ &\hat{C}_2(\mathbf{X}_2) \le C_2^0(1 - r_2^2) \end{aligned} \tag{2}$$

where $\mathbf{X}_i$ is the design variable vector of subspace i. The responsibility coefficients, r_k^p, represent the responsibility of the design variables in the kth subspace toward satisfying the constraints of pth subspace. The r_k^p are calculated with the sensitivity information. By means of the KS function [17], the cumulative constraint, C_i, which represents the constraints of subsystem i, is expressed as

$$C_i = \frac{1}{\rho} \ell_n \left(\sum_{k=1}^{m} \exp[\rho g_{ik}(\mathbf{X})] \right) \tag{3}$$

where $[g_{i1}, g_{i2}, \ldots, g_{im}] = \mathbf{G}_i$ and ρ is a positive value. In the modified CSSO [1], ρ is set to be a small value at the beginning and is increased in the process of optimization.

The flowchart of the modified CSSO method is shown in Fig. 1. In the first stage, based on initial design variables, the system analysis is performed. If the initial scenario does not satisfy the constraints, a minimization will be needed to reduce the initial infeasibility of the constraints as much as possible. In the subsequent stage, all the sensitivities are computed using the GSE. The GSE is a method for computing sensitivity derivatives of state (output) variables with respect to independent (input) variables for complex, internally coupled systems, while avoiding the cost and inaccuracy of finite differencing performed on the entire system analysis. The GSE will not be discussed here due to limited space, but the details of it can be found in [18]. Based on the sensitivity information, the impact of the design variables upon each subspace can be analyzed, and the design variables will be allocated to the subspace of the greatest impact. In the third stage, the subspace optimizations are performed concurrently. There is no system optimization in this method.

B. Adaptive Weighted Sum Method

The fundamental philosophy of the AWS method is to adaptively refine the Pareto front [15]. In the first stage, the ordinary weighted sum method is performed to approximate a rough profile of the Pareto front quickly, and a mesh of Pareto front patches is identified. In each Pareto front patch, the feasible domain for further exploration is determined by assigning additional constraints. In the subsequent stage, the weighted sum method is performed in these feasible domains to obtain more Pareto optimal solutions. When new Pareto optimal solutions are obtained, the Pareto front patch size estimation is again performed to determine regions for further refinement. These steps are repeated until a termination criterion is met.

The AWS method is first developed for biobjective optimization problems [15]. The principle of the biobjective AWS method is shown in Fig. 2. The line segment connecting two adjacent Pareto

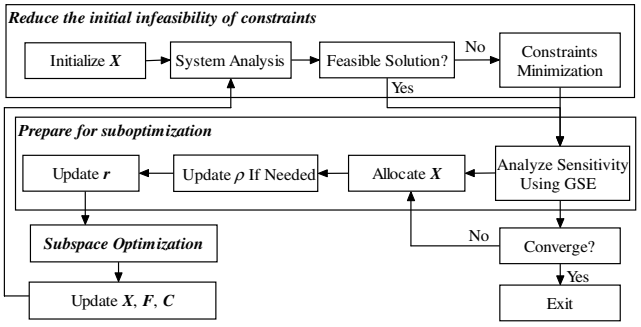


Fig. 1 Flowchart of the modified CSSO method.

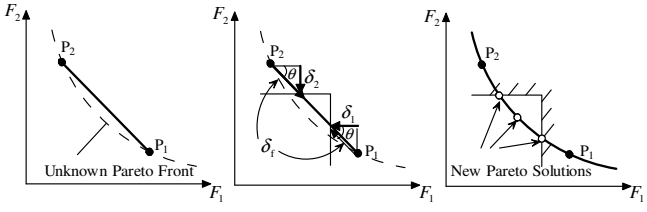


Fig. 2 Principle of biobjective adaptive weighted sum method.

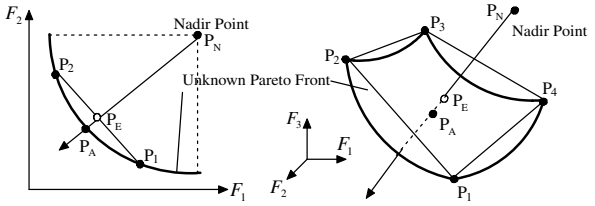


Fig. 3 Principle of the adaptive weighted sum method: a) biobjective, and b) multiobjective.

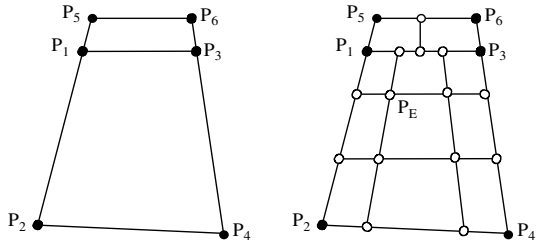


Fig. 4 Patches of AWS method: a) original, and b) refined.

points, P_1P_2, constructs a Pareto front patch. The feasible domain for further exploration is determined by specifying two additional inequality constraints ($F_1 \leq F_1(P_1) - \delta_1$ and $F_2 \leq F_2(P_2) - \delta_2$).

In multi-objective cases, as the Pareto front becomes a surface or a hyper-surface with an arbitrary shape, it is difficult to restrict optimizations in a selected Pareto front patch with just two inequality constraints. Then, the biobjective AWS method is extended to the problems with more than two objective functions [16]. The principle of the multi-objective AWS method is shown in Fig. 3. The linearized piecewise surface, $P_1P_2P_3P_4$, constructs a Pareto front patch, as shown in Fig. 3b. The mesh nodes are called expected solutions and denoted as P_E in Fig. 3. The line, P_NP_E, is expressed as an additional equality constraint and then appended to define feasible regions for further refinement. This method can also be used to solve biobjective optimization problems, as shown in Fig. 3a.

The multi-objective optimization is performed by the multi-objective AWS method in the following steps.

1) Stage = 1: Normalize the objective functions. When $\mathbf{X}^{i*}$ is the optimal solution vector for the single-objective optimization of the ith objective function J_i, the utopia point and pseudonadir point are defined as

$$J^{\text{utopia}} = [J_1(\mathbf{X}^{1*}), J_2(\mathbf{X}^{2*}), \ldots, J_m(\mathbf{X}^{m*})] \tag{4}$$

$$J^{\text{nadir}} = [J_1^{\text{nadir}}, J_2^{\text{nadir}}, \ldots, J_m^{\text{nadir}}] \tag{5}$$

where $J_i^{\text{nadir}} = \max[J_i(\mathbf{X}^{1*}) \quad J_i(\mathbf{X}^{2*}) \quad \cdots \quad J_i(\mathbf{X}^{m*})]$, and m is the number of objective functions. The point in objective space, J_i, can be normalized as

$$\bar{J}_i = \frac{J_i - J^{\text{nadir}}}{J^{\text{utopia}} - J^{\text{nadir}}} \tag{6}$$

2) Perform multi-objective optimization using the ordinary weighted sum method with a large step size of weighting factor to specify a rough profile of the Pareto front. The weighted single-objective function of m objective functions is determined as

$$J_{\text{total}}^m = \alpha_{m-1} J_{\text{total}}^{m-1} + (1 - \alpha_{m-1}) J_m, \qquad J_{\text{total}}^1 = J_1, \qquad \alpha_i \in [0, 1] \tag{7}$$

where α_i is the ith weighting factor. The uniform step size of the weighting factor is determined by the user.

3) Stage = Stage + 1: Mesh the Pareto front patches. After deleting nearly overlapping Pareto points, the adjacent Pareto points are connected to construct Pareto front patches with a special shape (quadrilateral patches are suggested in [16]). Then these patches are meshed according to their size. The larger the patch is, the more it needs to be refined. The example of refinement is shown in Fig. 4, in which a patch is composed of four nodes in three-dimensional objective space. Because the lower patch is larger, it is refined more than the upper one. In each mesh, the locations of expected solutions are determined as

$$P_E = \beta_1 P_1 + \beta_2 P_2 + \beta_3 P_3 + \beta_4 P_4 \qquad \beta_i \in [0, 1] \tag{8}$$

where P_i is the position vector of the node P_i of a Pareto front patch, and β_i is the weighting factor for interpolation.

4) After the location vectors for all expected solutions are computed, the optimization is conducted along the line that connects the pseudonadir point and the expected solution. As expressed in Fig. 3, each line connecting the expected solution P_E with the pseudonadir point P_N represents an equality constraint for the suboptimization. The equality constraint is defined as

$$\frac{(\bar{\mathbf{F}}_E - \bar{\mathbf{F}}_N) \cdot (\bar{\mathbf{F}}(\mathbf{X}) - \bar{\mathbf{F}}_N)}{|\bar{\mathbf{F}}_E - \bar{\mathbf{F}}_N||\bar{\mathbf{F}}(\mathbf{X}) - \bar{\mathbf{F}}_N|} = 1 \tag{9}$$

where $\bar{\mathbf{F}}_E$, $\bar{\mathbf{F}}_N$, and $\bar{\mathbf{F}}(\mathbf{X})$ are the normalized position vector of nodes P_E, P_N, and the current design point X, respectively.

For different expected solutions, the different equality constraints are constructed. With an additional constraint, the suboptimization is performed with the weighted sum method.

5) Perform Pareto filtering and remove those pseudo-Pareto points. This step is especially necessary for the Pareto front with a discontinuity.

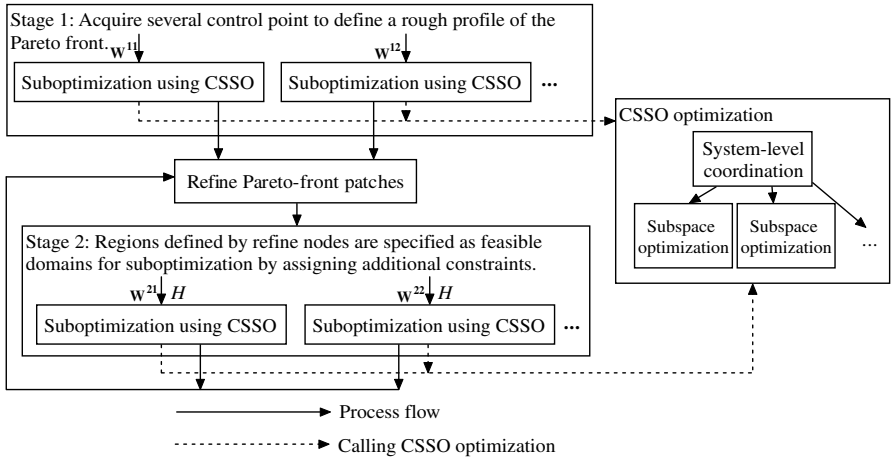


Fig. 5 The framework of the AWSCSSO method.

6) Append the newly obtained solutions to the initial ones and return to step 3. These steps are repeated until the termination criterion is met.

III. Adaptive Weighted Sum Concurrent Subspace Optimization Method Development

To obtain better distributed solutions for multi-objective MDO problems, a novel integration of the AWS method within the CSSO framework is proposed. In this section, the proposed AWSCSSO method is described, and issues pertaining to the modification when AWS is applied in the bilevel framework are discussed.

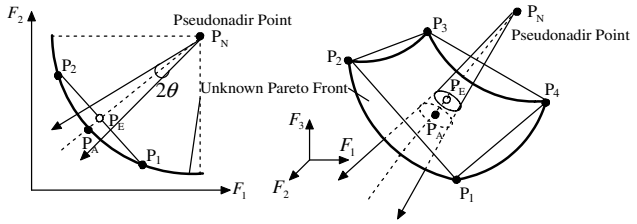


Fig. 6 AWSCSSO method for multidimensional problems: a) 2-D representation, and b) 3-D representation.

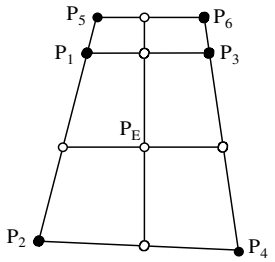


Fig. 7 Refined patches of the AWSCSSO method.

A. Adaptive Weighted Sum Concurrent Subspace Optimization Method

The motivation to develop the AWSCSSO method is to be able to use an AWS-based strategy to solve the multi-objective MDO problem, with the intent of obtaining a more uniformly spaced, more widely distributed, and smoother Pareto front. Mathematically, for minimization problems the general form for the multi-objective MDO can be represented as follows:

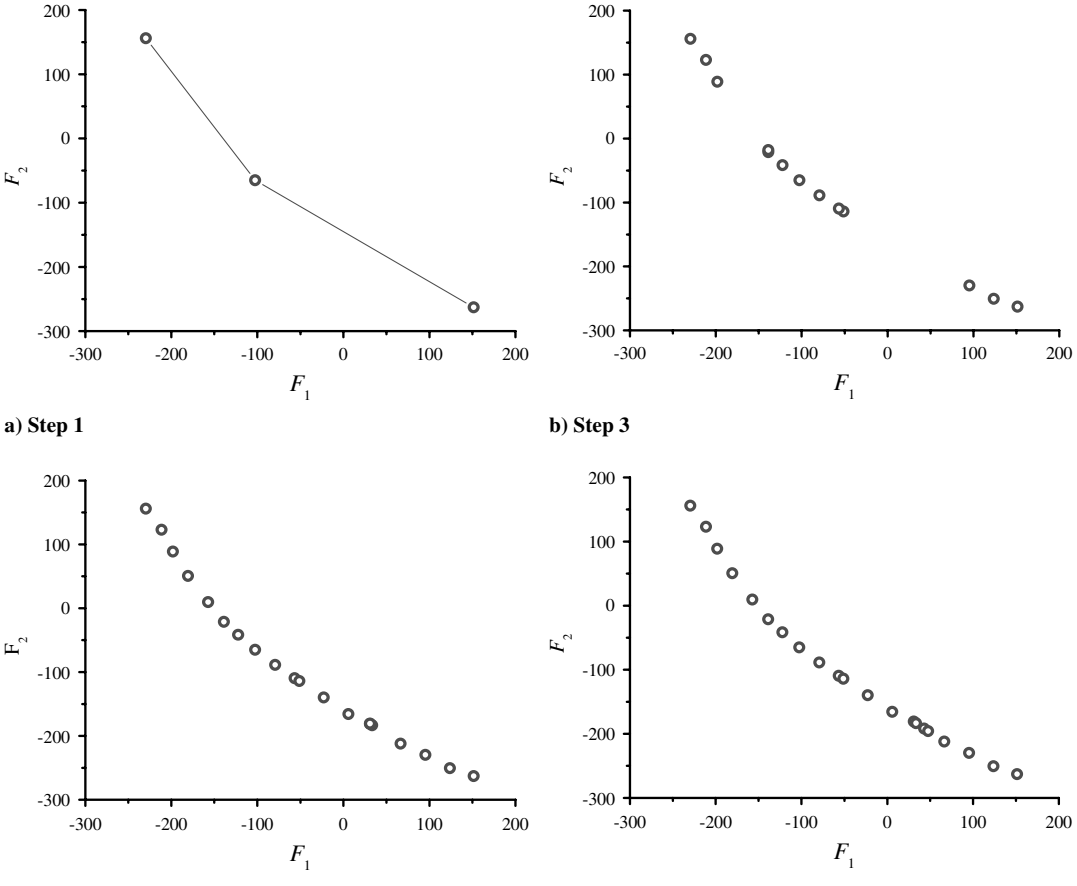
$$\begin{aligned} &\text{Min } \mathbf{F}(\mathbf{X}, \mathbf{Y}_1, \ldots, \mathbf{Y}_N) \\ &\text{s.t. } \mathbf{G}_i(\mathbf{X}, \mathbf{Y}_1, \ldots, \mathbf{Y}_N) \le \mathbf{0} \\ &\mathbf{Y}_i = f_i(\mathbf{X}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{i-1}, \mathbf{Y}_{i+1}, \ldots, \mathbf{Y}_N) \\ &i = 1, 2, \ldots, N \end{aligned} \tag{10}$$

where $\mathbf{F}$ is the objective function vector that is composed of two or more objectives, $\mathbf{G}_i$ is the constraint vector provided by subsystem i, $\mathbf{Y}_i$ is the coupling vector of subsystem i, and $\mathbf{X}$ is the design vector. The objective function and the constraints can be expressed by the function of $\mathbf{X}$ and $\mathbf{Y}_i$.

The procedure of solving the Pareto front by AWSCSSO is similar to the AWS method. As an example, the AWSCSSO method for a generic biobjective problem, with subsystems 1 and 2, is stated in the following paragraphs. In the same way, AWSCSSO can be also applied to the multi-objective problem with three or more subsystems.

1) In the first stage, a rough profile of the Pareto front is determined.

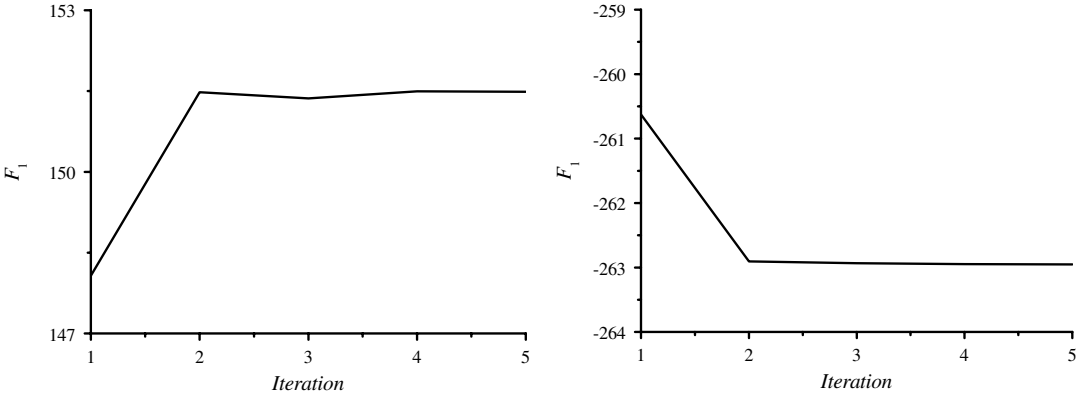
The CSSO method is adopted in the single-objective optimization for each objective function, and the normalization of objective function is performed using Eq. (6). Then, with a large step size of the



a) Step 1 **b) Step 3**

c) Step 5 **d) Step 7**

Fig. 8 Pareto front obtained by using AWSCSSO.



a) Objective function 1 **b) Objective function 2**

Fig. 9 Convergence history of one Pareto point obtained in stage 1.

weighting factor, the usual weighted sum method is used in CSSO to approximate the Pareto front quickly. The subspace optimization for AWSCSSO can be expressed as

$$\begin{aligned}
&\text{Suboptimization 1}\\
&\text{Min } W_1F_1(\mathbf{X}_1,\mathbf{Y}_1,\hat{\mathbf{Y}}_2)+W_2\hat{F}_2(\mathbf{X}_1)\\
&\text{s.t. } C_1(\mathbf{X}_1,\mathbf{Y}_1,\hat{\mathbf{Y}}_2)\le C_1^0(1-r_1^1)\\
&\hat{C}_2(\mathbf{X}_1)\le C_2^0(1-r_1^2)\\
&\mathbf{Y}_1=f_1(\mathbf{X}_1,\hat{\mathbf{Y}}_2)\\
&\text{Suboptimization 2}\\
&\text{Min } W_1\hat{F}_1(\mathbf{X}_2)+W_2F_2(\mathbf{X}_2,\hat{\mathbf{Y}}_1,\mathbf{Y}_2)\\
&\text{s.t. } \hat{C}_1(\mathbf{X}_2)\le C_1^0(1-r_2^1)\\
&C_2(\mathbf{X}_2,\hat{\mathbf{Y}}_1,\mathbf{Y}_2)\le C_2^0(1-r_2^2)\\
&\mathbf{Y}_2=f_2(\mathbf{X}_2,\hat{\mathbf{Y}}_1,\hat{\mathbf{Y}}_3)
\end{aligned}\tag{11}$$

where the value with the symbol ˆ above is a linearly approximated one, C_1 and C_2 are cumulative constraints of $\mathbf{G}_1$ and $\mathbf{G}_2$, respectively, and r_k^p represents the responsibility assigned to the kth subsystem for reducing the violation of C_p. The value with superscript 0 corresponds to the starting point $\mathbf{X}_0$. W_1 and W_2 are weighting factors for the objective functions F_1 and F_2, respectively.

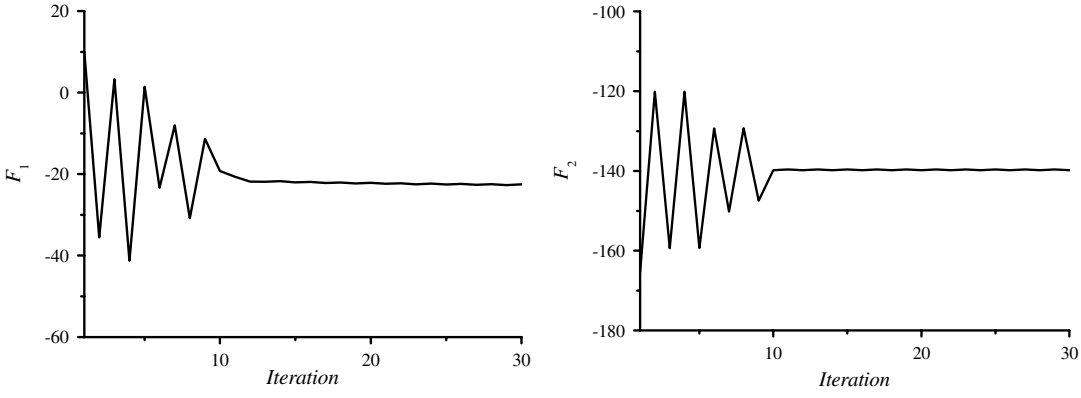
By estimating the size of each Pareto patch, the refined regions in the objective space are determined. The refinement of the Pareto front patch will be discussed in detail in Sec. III.B.

2) In the subsequent stage, only these regions are specified as feasible domains for the suboptimization problems with additional constraints. Each Pareto front patch is then refined by imposing additional equality constraints that connect the pseudonadir point and the expected Pareto optimal solutions on a piecewise planar surface in the objective space (as shown in Fig. 3). Suboptimizations are defined by imposing the additional constraint H to Eq. (11):

$$\begin{aligned}
&\text{Suboptimization 1}\\
&\text{Min } W_1F_1(\mathbf{X}_1,\mathbf{Y}_1,\hat{\mathbf{Y}}_2)+W_2\hat{F}_2(\mathbf{X}_1)\\
&\text{s.t. } C_1(\mathbf{X}_1,\mathbf{Y}_1,\hat{\mathbf{Y}}_2)\le C_1^0(1-r_1^1)\\
&\hat{C}_2(\mathbf{X}_1)\le C_2^0(1-r_1^2)\\
&\mathbf{Y}_1=f_1(\mathbf{X}_1,\hat{\mathbf{Y}}_2)\\
&H(\mathbf{X}_1,\mathbf{Y}_1,\hat{\mathbf{Y}}_2)\le 0\\
&\text{Suboptimization 2}\\
&\text{Min } W_1\hat{F}_1(\mathbf{X}_2)+W_2F_2(\mathbf{X}_2,\hat{\mathbf{Y}}_1,\mathbf{Y}_2)\\
&\text{s.t. } \hat{C}_1(\mathbf{X}_2)\le C_1^0(1-r_2^1)\\
&C_2(\mathbf{X}_2,\hat{\mathbf{Y}}_1,\mathbf{Y}_2)\le C_2^0(1-r_2^2)\\
&\mathbf{Y}_2=f_2(\mathbf{X}_2,\hat{\mathbf{Y}}_1,\hat{\mathbf{Y}}_3)\\
&H(\boldsymbol{X}_2,\boldsymbol{Y}_1,\hat{\boldsymbol{Y}}_3)\le 0
\end{aligned}\tag{12}$$

The additional equality constraint is relaxed in AWSCSSO, which is different from AWS. If the additional equality constraint is directly imposed on the optimization of subspaces, the convergence can hardly be achieved due to the nconsistencies between the subspace designs. The additional constraint will be discussed in detail in Sec. III.B.

Figure 5 shows the framework of AWSCSSO. In Fig. 5, W^{1i} and W^{2i} are the weighting factors in stages 1 and 2, respectively; H is the additional constraint. The optimization problem is performed in two stages in the AWSCSSO method. In the first stage, the Pareto front is approximated quickly with large step size of the weight factors. The optimization problems of this stage are defined in Eq. (11). In the subsequent stage, by calculating the distances between neighboring



a) Objective function 1 **b) Objective function 2**

Fig. 10 Convergence history of one Pareto point obtained in stage 2.

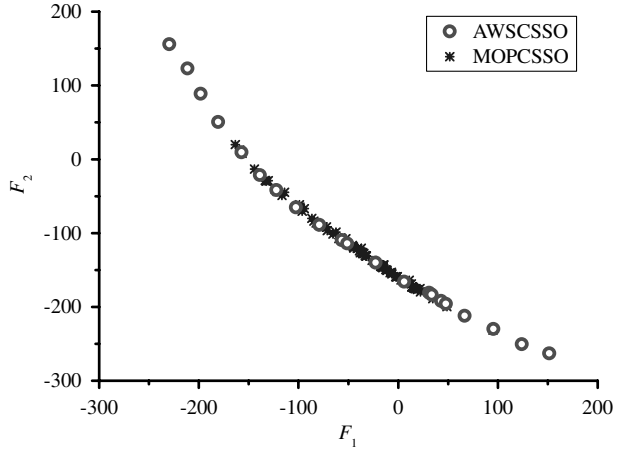


Fig. 11 Comparison of Pareto front obtained by using AWSCSSO and MOPCSSO.

solutions on the Pareto front in objective space, the refined regions are identified and the refined mesh is formulated. Only these regions then become the feasible regions for optimization by imposing additional constraints in the objective space. The optimization problems of this stage are defined in Eq. (12). The different locations of new Pareto points are defined by the different additional constraints. Optimization is performed in each of the regions, and the new solution set is acquired. Being an MDO problem, the optimization is performed by the CSSO method.

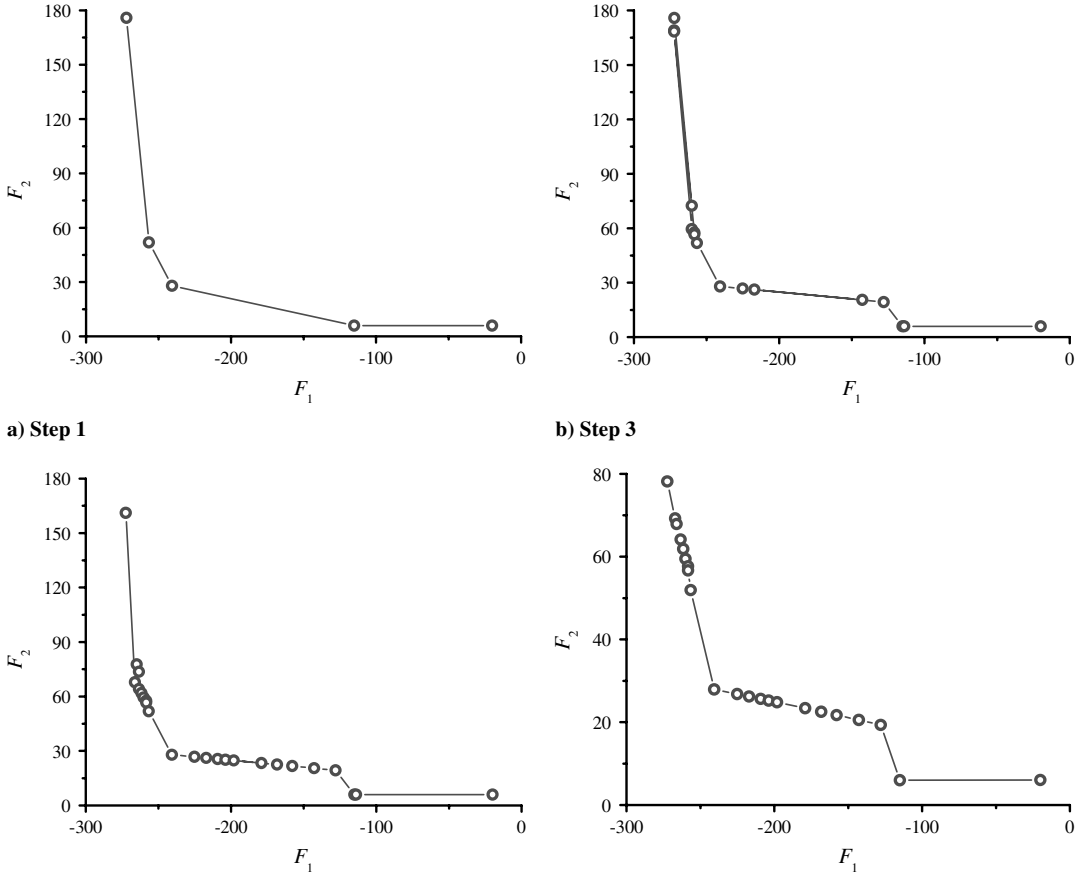
B. Key Points

To obtain better distributed solutions, two modifications are made to the AWSCSSO method. On the one hand, the additional equality constraint in suboptimization for each expected solution is relaxed because it causes slow convergence of the suboptimization. The probability of entrapment in local minima can also be decreased. On the other hand, the mesh of the Pareto front patches is modified due to the low efficiency of the original mesh. These two modifications will be discussed in detail here.

1) The additional equality constraint in suboptimization for each expected solution is relaxed.

The concept of the AWSCSSO method with relaxed equality constraints for multi-objective problems is shown in Fig. 6. After relaxation of the equality constraint, the search is in the region with cone angle 2θ instead of along the line $\overline{P_N P_E}$. In Fig. 6, P_1, P_2, P_3, and P_4 are Pareto points obtained in former steps, which determine a rough profile of the Pareto front. P_N is called the pseudonadir point [15,16], the location of which in objective space is computed using Eq. (5). Line segment $\overline{P_1 P_2}$ and quadrilateral patch $\overline{P_1 P_2 P_3 P_4}$ are the piecewise linearized Pareto front. If the Pareto front is known, the extension of the line segment $\overline{P_N P_E}$ will intersect with the Pareto front, and the point of intersection is the actual solution, P_A. It means that the Pareto point P_A could be acquired if the optimization is performed along $\overline{P_N P_E}$.

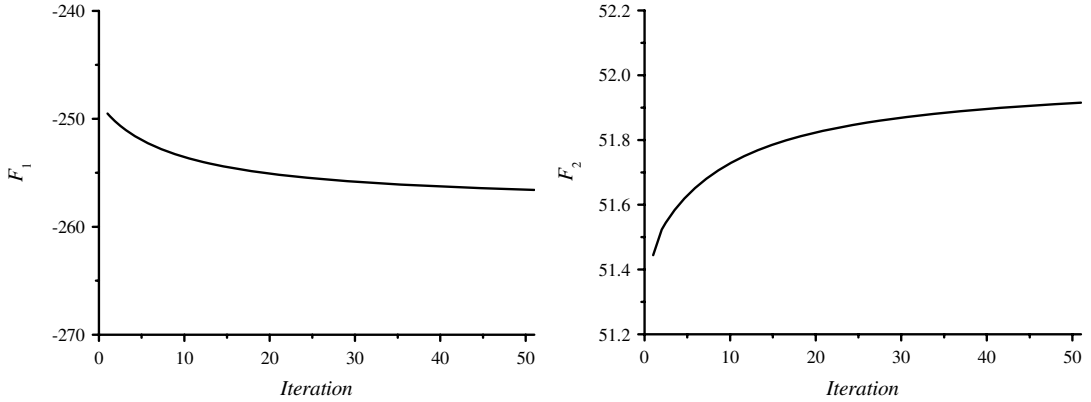
However, the additional equality constraint in the AWS method is not fit for the AWSCSSO method. On the one hand, because any solution that lies on the additional equality constraint is feasible, the probability of entrapment in the local minima is increased by the equality constraint. On the other hand, the equality constraints can hardly be satisfied within the bilevel optimization framework due to the disagreement between the different subspaces. The bilevel framework of the AWSCSSO method is designed to promote disciplinary autonomy. The design of the different subspaces is inconsistent. The interdisciplinary compatibility could be achieved through repeated subspace optimizations and system-level coordination. In a conventional all-at-once optimization, the slow convergence is often caused by equality constraints. In a bilevel framework, the equality constraints are more difficult to satisfy



a) Step 1 **b) Step 3**

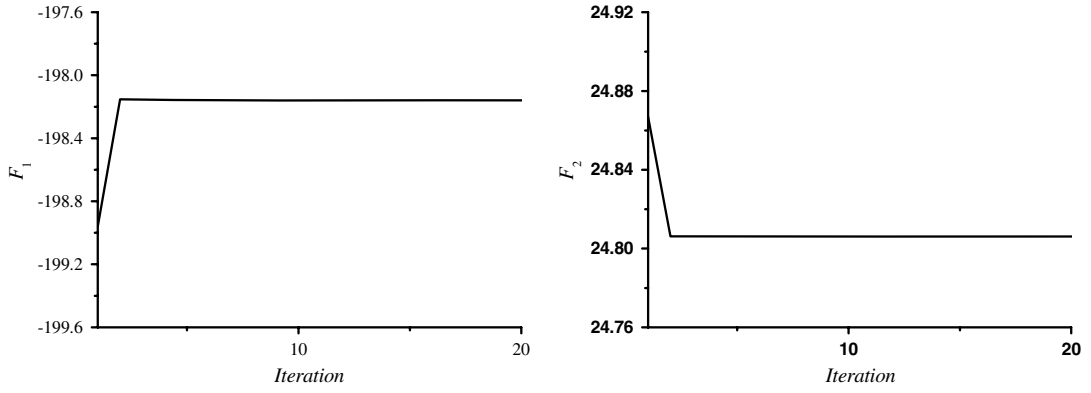
c) Step 6 **d) Step 8**

Fig. 12 Pareto front obtained by using AWSCSSO.



a) Objective function1 **b) Objective function 2**

Fig. 13 Convergence history of one Pareto point obtained in stage 1.



a) Objective function 1 **b) Objective function 2**

Fig. 14 Convergence history of one Pareto point obtained in stage 2

because the perfect compatibility of the subspace is impossible to realize. AWSCSSO optimization cannot converge to the expected point or not converge at all. Therefore, a special relax technique is proposed to solve this problem in this paper, as expressed in Eq. (13):

$$H = \frac{(\bar{\mathbf{F}}_E - \bar{\mathbf{F}}_N) \cdot (\bar{\mathbf{F}}(\mathbf{X}) - \bar{\mathbf{F}}_N)}{|\bar{\mathbf{F}}_E - \bar{\mathbf{F}}_N||\bar{\mathbf{F}}(\mathbf{X}) - \bar{\mathbf{F}}_N|} + L \leq 0 \tag{13}$$

where L is the adaptive relax factor that is less than 1. The left term of Eq. (13) is the cosine value of the included angle between vectors $\mathbf{P}_N\mathbf{P}_E$ and $\mathbf{P}_N\mathbf{P}_X$ (see Fig. 6). When θ is larger than zero, the relax value is (1-L). Once θ reaches the ideal value of zero, the relax value becomes zero and Eq. (13) is the same as Eq. (9). Whatever θ is, $\cos\theta$ can never be more than 1. And so $\cos\theta$ need only be limited by the lower bound L. If L is too close to 1, AWSCSSO can hardly converge, as Eq. (13) is almost an equality constraint. If L is too far from 1, the Pareto points are difficult to distribute uniformly, as the equality constraints are relaxed excessively. In AWSCSSO, L is set to be increased with the rise of the distribution density of the Pareto points. In Fig. 6, Eq. (9) is shown as line $\overline{P_N P_E P_A}$, whereas Eq. (13) is shown as the taper region with line $\overline{P_N P_E P_A}$ as the centerline.

2) The layout for the mesh refinement is modified.

It is shown in Fig. 4 that in the AWS method the layout for mesh refinements in each of the Pareto front patches is determined by the size of the patches. The larger the patch is, the more it needs to be refined. However, if the layout for the mesh refinement of the AWS method is used in AWSCSSO, the optimizations with different

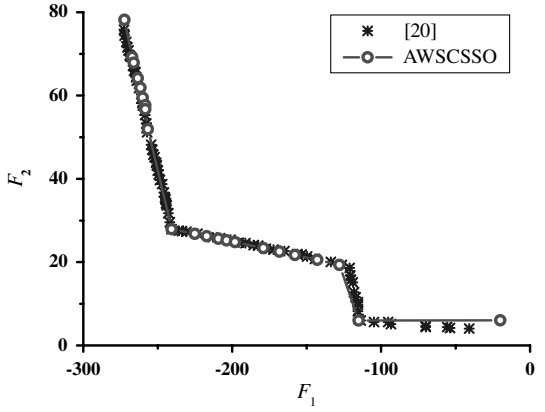


Fig. 15 Comparison of Pareto front obtained by using AWSCSSO and [20].

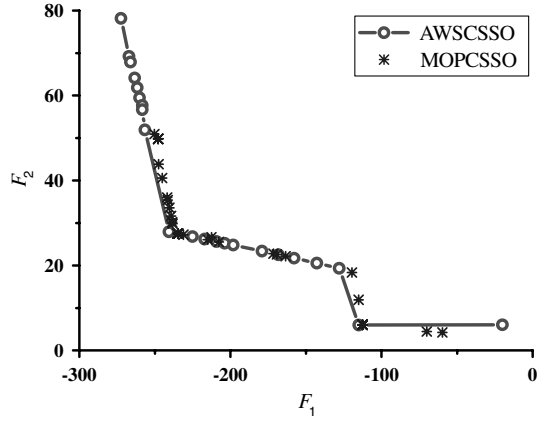


Fig. 16 Comparison of Pareto front obtained by using AWSCSSO and MOPCSSO.

Table 1 Design variables

Design variable	Symbol	Lower limit	Upper limit
Wing area, m^2	S	111.48	232.26
Aspect ratio	AR	9.5	10.5
Design gross weight, 10^3 kg	W_{dg}	63.504	113.400
Installed thrust, 10^3 kg	T_i	12.587	24.948

equality constraints defined by different refined nodes P_E in the neighboring position may converge to overlapping solutions. In this case, the computational efficiency will be reduced. Thus, in AWSCSSO the Pareto front patch could not be refined too much in one step. An example of the mesh refinement in AWSCSSO is shown in Fig. 7, in which hollow points represent the newly refined node P_E (expected solution) and solid points represent the four initial nodes that define the patch. As shown in Fig. 7, the quadrilateral patch is given as an example. If the line segment that connects two neighboring nodes of the patch is too long, it is divided into only two equal line segments. The central point becomes the new refined node. These refined nodes are connected to form a refined mesh. Then the suboptimizations in Eq. (12) are performed using different additional constraints for different refined nodes, and the new Pareto points are obtained. In the next step, according to the prescribed density of the Pareto points, the Pareto front patch that is too large will be refined again in the same way. In subsequent steps, the refinement and suboptimizations are repeated until the number of Pareto points does not increase anymore.

IV. Adaptive Weighted Sum Concurrent Subspace Optimization Method Validation

The proposed method is validated and preliminarily demonstrated with three MDO problems: 1) a numerical MDO test problem with a convex Pareto front, which is from the NASA Langley Research Center MDO Test Suite; 2) a test problem with a nonconvex Pareto front, which is not easily solved; and 3) a conceptual design of a subsonic passenger aircraft, which consists of two objectives, four design variables, five coupling behavior variables, and seven constraints in aerodynamics and weight discipline.

A. Example 1: Convex Pareto Front

This problem is taken from a test problem in [1,19]. This is a test problem available in the NASA Langley Research Center MDO Test Suite. It has two objectives, F_1 and F_2, to be minimized. It consists of 10 inequality constraints, 4 coupled state variables, and 10 design variables in 2 coupled subsystems. The mathematical model is not listed here for concision and the reader can refer to test problem 1 in [1,19].

The steps performed to obtain the Pareto front in objective space by AWSCSSO are shown in Fig. 8. In step 1, a rough profile of the Pareto front is determined. The mathematical model of this stage corresponds to Eq. (11). From steps 2 to 7, the Pareto front is gradually refined. The mathematical model of these stages corresponds to Eq. (12). Taking one of the Pareto points obtained in step 1 as an example, the convergence history of the objective function is shown in Fig. 9. The convergence history of another Pareto point obtained in steps 2–7 is shown in Fig. 10.

It can be seen that the convergence of AWSCSSO is good in stage 1 (i.e., step 1) and is not so good in stage 2 (i.e., steps 2–7) due to the additional constraints.

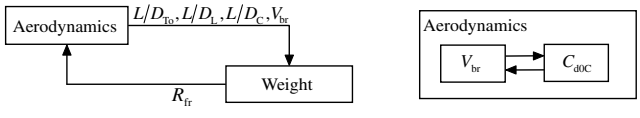


Fig. 17 Dataflow between and in the subspaces.

The comparison of the solution obtained by MOPCSSO [1] and by the proposed AWSCSSO is shown in Fig. 11. It can be concluded that for the problem with a convex Pareto front a uniformly spaced, widely distributed, and smooth Pareto front can be obtained by the proposed method in this paper. When using MOPCSSO, the whole range of the Pareto front has not been captured, as locations of the Pareto points to some extent depend on where the starting points lie.

B. Example 2: Nonconvex Pareto Front

This problem [20] consists of two objective functions, six design variables, and six constraints. Two objectives, F_1 and F_2, need to be minimized. The model problem is defined as

$$
\begin{aligned}
&\text{Min } F_1(\boldsymbol{x}) = -(25(x_1-2)^2 + (x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2 \\
&\quad + (x_5-1)^2) \\
&\text{Min } F_2(\boldsymbol{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \\
&\text{s.t. } c_1(\boldsymbol{x}) = x_1 + x_2 - 2 \geq 0 \\
&\qquad c_2(\boldsymbol{x}) = 6 - x_1 - x_2 \geq 0 \\
&\qquad c_3(\boldsymbol{x}) = 2 + x_1 - x_2 \geq 0 \\
&\qquad c_4(\boldsymbol{x}) = 2 - x_1 + 3x_2 \geq 0 \\
&\qquad c_5(\boldsymbol{x}) = 4 - (x_3-3)^2 - x_4 \geq 0 \\
&\qquad c_6(\boldsymbol{x}) = (x_5-3)^2 + x_6 - 4 \geq 0 \\
&\qquad 0 \leq x_1, x_2, x_6 \leq 10, 1 \leq x_3, x_5 \leq 5, 0 \leq x_4 \leq 6
\end{aligned}
\tag{14}
$$

The problem in Eq. (14) is divided into two subspaces. F_1 and constraints c_1–c_4 belong to subspace 1, whereas F_2 and constraints c_5–c_6 belong to subspace 2. The procedure to obtain the Pareto front in objective space by AWSCSSO is shown in Fig. 12. In step 1, a rough profile of the Pareto front is determined. From steps 2 to 8, the Pareto front is gradually refined. Because the point [−272.20, 175.89] obtained in the first step is a pseudo-Pareto point, the left segment of the rough profile of the Pareto front is not correct. As more and more Pareto points are obtained as the optimization continues, the pseudo-Pareto point is recognized and removed. That is why the final Pareto front is slightly different from that obtained in the initial steps. Taking one of the Pareto points obtained in stage 1 (step 1) as an example, the convergence history of objective functions is shown in Fig. 13. The convergence history of another Pareto point obtained in stage 2 (steps 2–8) is shown in Fig. 14. To validate the effectiveness and correctness of the AWSCSSO method, the Pareto front computed using the AWSCSSO method is compared with the one computed using multi-objective evolutionary algorithms [20], as shown in Fig. 15. Apparently, the Pareto front of our method is in good agreement with that of [20]. However, the solutions are not so uniformly spaced as those in example 1. There may be two reasons for this:

1) Because Pareto points lie on a nonlinear constraint surface, an optimization algorithm may have more difficulty finding a spread of solutions across the entire Pareto front.

2) After relaxation, the additional constraint becomes a taper region in the objective space. The possibility of the optimization being premature is largely reduced. However, if the taper region is too small, the optimization will possibly be premature and the pseudo-Pareto points will appear. Furthermore, the disagreement between subspaces and the special difficulty of solving the problem in Eq. (14) will possibly cause the optimization to be premature.

The comparison of the Pareto front obtained by AWSCSSO and by MOPCSSO is shown in Fig. 16. It is concluded that, for the problem with a nonconvex Pareto front, the more uniformly spaced, more widely distributed, and smoother Pareto front is also obtained by the AWSCSSO method.

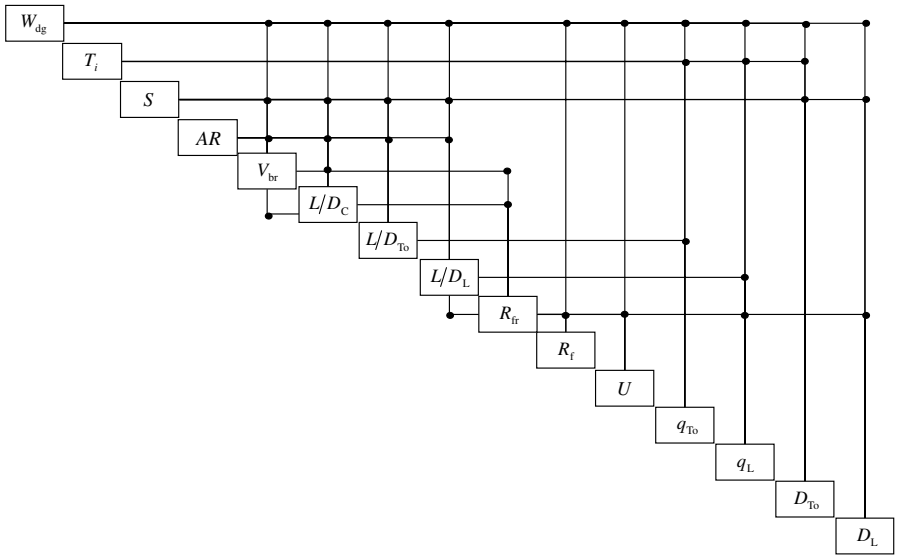


Fig. 18 Dataflow between the state variables and design variables.

C. Example 3: Conceptual Design of a Subsonic Passenger Aircraft

The mathematical model of this problem is defined as

$$\begin{aligned}
&\text{Max } U\\
&\text{Max } L/D_C\\
&\text{s.t. } C_{d0L} \le 0.2, \qquad C_{d0C} \le 0.02\\
&\quad R_f \ge 1\\
&\quad q_{\text{To}} \ge 0.027, \qquad q_L \ge 0.024\\
&\quad D_{\text{To}} \le 1981, \qquad D_L \le 1371
\end{aligned} \tag{15}$$

The objective functions in Eq. (15) are to maximize the useful load fraction (U) and lift-to-drag ratio for the cruising condition (L/D_C). The constraints in Eq. (15) are as follows:

1) The drag coefficient for the takeoff condition and landing condition (C_{d0L}) is no more than 0.2 and for the cruising condition (C_{d0C}) is no more than 0.02.

2) The overall fuel balance coefficient (R_f) is no less than 1.

3) The achievable climb gradient for the takeoff condition (q_{To}) is greater than 0.027 and for the landing condition (q_L) is greater than 0.024.

4) The takeoff field length (D_{To}) is less than 1981 m and the landing field length (D_L) is less than 1371 m.

The overall fuel balance coefficient is defined as the ratio of the fuel weight required for the mission to that available for the mission. The design variables are listed in Table 1.

Two disciplines, aerodynamics and weight, are considered in this problem.

The data flow between and in subsystems is analyzed in Fig. 17, in which L/D_{To}, L/D_L, and L/D_C are the lift-to-drag ratios for the takeoff, landing, and cruising conditions, respectively; V_{br} is the cruise velocity with the longest range; R_{fr} is the fuel weight fraction required for the mission; and C_{d0C} is the zero-lift drag coefficient for the cruising condition.

The two disciplines, aerodynamics and weight, are coupled. When the state variables in aerodynamics such as the cruise velocity with the longest range, lift coefficients, zero-lift drag coefficients, skin-friction drag coefficients, and the lift-to-drag ratio are computed, some state variables in weight such as R_{fr} should be known. Similarly, when the state variables in weight such as the useful load fraction, the overall fuel balance coefficient, the achievable climb gradient on takeoff and landing, the takeoff field length, and the landing field length are computed, some state variables in aerodynamics such as L/D_{To}, L/D_L, L/D_C, and V_{br} should also be provided. In the aerodynamics discipline, V_{br} is coupled with C_{d0C}. The data flow between state variables and design variables can be seen in Fig. 18. Many details of the equations in the aerodynamic and weight discipline models are described in the Appendix. The full description of them can be found in [21,22].

The procedure of obtaining the Pareto front by AWSCSSO is shown in Fig. 19. In step 1, a rough profile of the Pareto front is determined. From steps 2 to 9, the Pareto front is gradually refined. Each solution on the Pareto front is obtained using CSSO with iterative subspace optimizations. Taking one of the Pareto points obtained in stage 1 (step 1) as an example, the convergence history of objective function is shown in Fig. 20. The convergence history of another Pareto point obtained in stage 2 (steps 2–9) can be seen in Fig. 21. Taking one of the optimal designs as an example, the values of the design variables are $S = 232.3$ m^2, $AR = 10.5$, $W_{\text{dg}} = 113.4 \times 10^3$ kg, and $T_i = 16.75 \times 10^3$ kg. The performance parameters of the aircraft in optimal design are $C_{d0C} = 0.01777$, $L/D_C = 21.05$, $V_{\text{br}} = 183.43$ m/s, $q_{\text{To}} = 0.03303$, $q_L = 0.08804$, $D_{\text{To}} = 1823$ m, and $D_L = 1086$ m. Two conclusions can be made from these results:

1) The AWSCSSO method is primarily proved to be applicable to aircraft conceptual design.

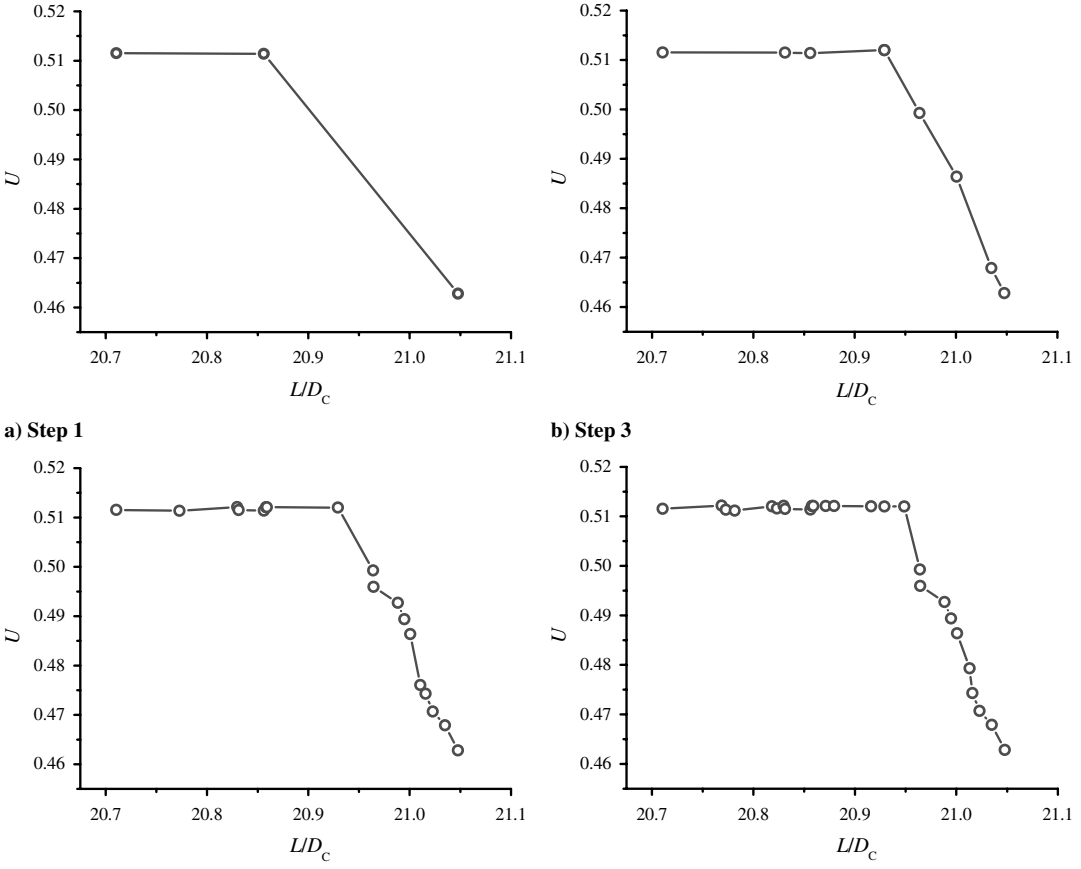
2) The distribution of Pareto points is not as uniform as expected.

These results are still very encouraging in general. The nonuniformity may be due to the additional constraint that changes the location of the expected solution. Further study is still needed on how to achieve the balance between uniformity and convergence.

V. Conclusions

Because of the multi-objective nature of most MDO problems, this work focuses on developing an effective multi-objective MDO method. By using the AWSCSSO method, a uniformly spaced, widely distributed, and smooth Pareto front can be obtained for the design of large-scale, complex engineering systems such as aircraft. Two numerical examples and an aircraft conceptual design problem are used to validate the AWSCSSO method. Several remarks can be concluded as follows. First, AWSCSSO is effective at and applicable to solving multi-objective MDO problems. For test problems and aircraft conceptual design problems, it provides the entire Pareto front. Second, AWSCSSO is a promising method for obtaining a uniformly spaced, widely distributed, and smooth Pareto front for multi-objective MDO problems.

As it is well known that it is difficult to acquire uniform, widely distributed Pareto points in a bilevel optimization framework, future work will focus on further improving the solution quality and also on testing it for more realistic engineering design problems.



a) Step 1 b) Step 3

c) Step 6 d) Step 9

Fig. 19 Pareto front obtained by using AWSCSSO.

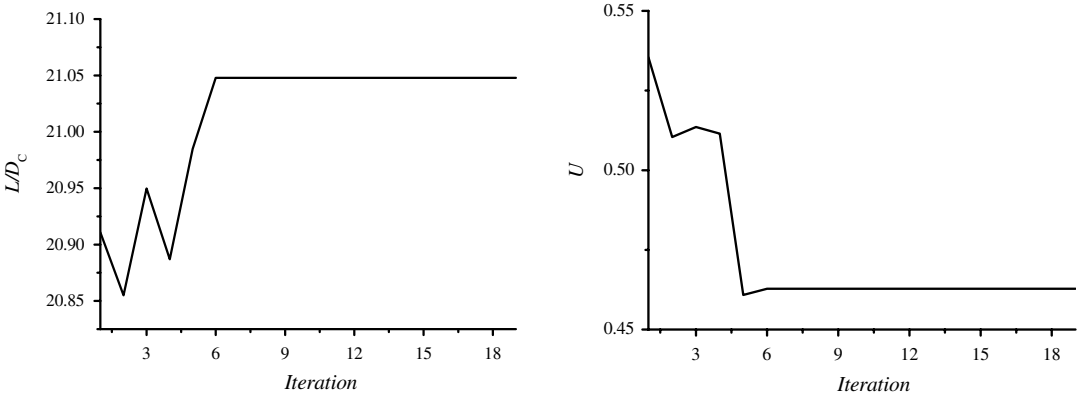


Fig. 20 Convergence history of one Pareto point obtained in stage 1.

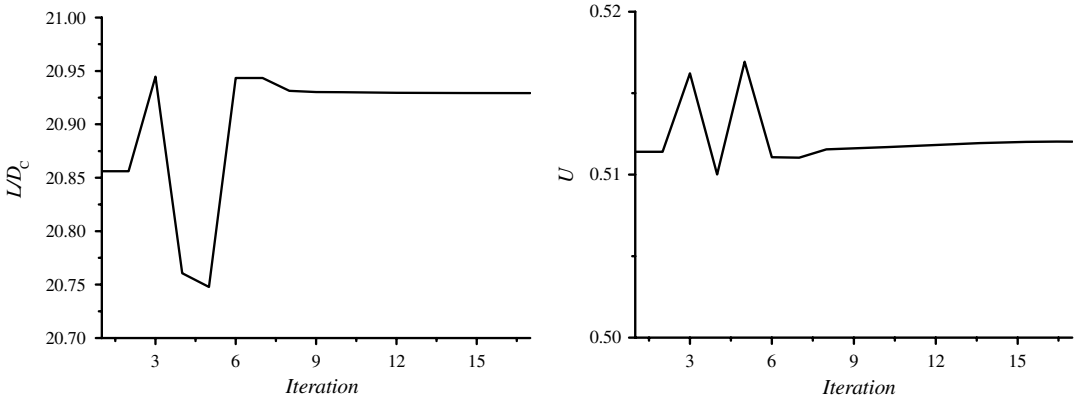


Fig. 21 Convergence history of one Pareto point obtained in stage 2.

Appendix: Mathematical Model of Example 3

The given parameters of example 3 are listed in Table A1.

The analysis models of the aerodynamic and weight subsystems are stated as follows.

I. Aerodynamic Subsystem Model

a) Fuselage diameter

$$d_f = 0.558(1.318(N_P/l_f) + 1)$$

Body wetted surface ratio

$$S_{f\text{wet}} = \frac{\pi d_f l_f}{S} \cdot \left(1 - 2\frac{d_f}{l_f}\right)^{2/3} \cdot \left(1 + \left(\frac{d_f}{l_f}\right)^2\right)$$

b) Velocity in the cruising condition

$$V_{\text{br}} = \sqrt{\frac{2W_{\text{dg}}}{\rho_C S\sqrt{\frac{C_{d0C}}{k}}}}$$

Oswald factor

$$e = 0.96(1 - (d_f/B)^2)$$

Quadratic drag polar

$$k = 1/(\pi \cdot AR \cdot e)$$

c) Lift coefficient

$$C_{L\text{To}} = \frac{W_{\text{dg}}}{\frac{1}{2}\rho_0 V_0^2 S}$$

if in takeoff.

$$C_{LL} = \frac{W_{\text{dg}}(1 - R_{\text{fr}})}{\frac{1}{2}\rho_0 V_0^2 S}$$

if in landing.

d) Skin friction coefficient

$$C_f = 0.455\left(\log_{10}\left(\frac{V \cdot l}{\mu}\right)\right)^{-2.58}$$

$l = (S/B)$ for wing and $l = l_f$ for body.

e) Zero-lift drag coefficient

$$C_{d0} = C_{d0w} + C_{d0f} + \Delta C_{d0}$$

Wing contribution

$$C_{d0w} = 1.1C_{fw}(1 + 1.2(t/c) + 100(t/c)^4) \cdot S_{\text{wet}}$$

Body contribution

$$C_{d0f} = C_{ff}(1 + 0.0025(d_f/l_f) + 100(d_f/l_f)^3) \cdot S_{f\text{wet}}$$

Incremental drag coefficient

$$\Delta C_{d0} = 0.005$$

f) Lift-to-drag ratio, landing and takeoff

$$\frac{L}{D} = \frac{C_L}{C_{d0} + kC_L^2}$$

Lift-to-drag ratio, cruising

$$\frac{L}{D} = \frac{1}{2\sqrt{kC_{d0C}}}$$

II. Weight Subsystem Model

a) Useful load fraction

$$U = 1.1(1 - 0.95R_{fw}) + (W_{\text{pay}}/W_{\text{dg}})$$

Fuel weight fraction

$$R_{fw} = \exp\left(-\frac{C \cdot R}{L/D_C \cdot V_{\text{br}}}\right)$$

b) Fuel weight required for mission

$$R_{\text{fr}} = 1.1(1 - 0.95R_{fw})$$

c) Overall fuel balance

$$R_f = R_{\text{fa}}/R_{\text{fr}}$$

Fuel weight available

$$R_{\text{fa}} = 1 - \frac{W_{\text{pay}}}{W_{\text{dg}}} - \frac{W_{\text{fix}}}{W_{\text{dg}}} - \frac{W_{\text{empty}}}{W_{\text{dg}}}$$

Empty weight

$$W_{\text{empty}} = W_{\text{dg}}\left(\frac{0.912}{W_{\text{dg}}^{0.0638}} + \frac{0.37644T_i^{0.9881}}{W_{\text{dg}}}\right)$$

Table A1 Given parameters

Give parameters	Symbol	Value
Number of passengers	N_P	188
Range, K	R	5374
Cruising altitude, Km	H	10.668
Maximum lift coefficient	$C_{L\max}$	2.6
Payload (cargo and passengers), kg	W_{pay}	18144
Fix equipment weight, kg	W_{fix}	499
Number of engines	N_e	3
Airfoil thickness-to-chord ratio	t/c	0.12
Density, sea level (takeoff, landing), kg/m^3	ρ_0	1.225
Density, cruising, kg/m^3	ρ_C	0.3801
Kinematic viscosity, sea level (takeoff, landing), m^2/s	μ_0	0.1449×10^{-4}
Kinematic viscosity, cruising, m^3/s	μ_C	0.3772×10^{-4}
Velocity, sea level (takeoff, landing), m/s	V_0	67.056
Horizontal tail area, m^2	S_{HT}	$0.2S$
Vertical tail area, m^2	S_{VT}	$0.1S$
Fuselage length, m	l_f	45
Engine specific fuel consumption, 1/h	C	0.70

d) Achievable climb gradient, takeoff

$$q_{\mathrm{To}} = \frac{-1}{L/D_{\mathrm{To}}} + \left(1 - \frac{1}{N_e}\right)\frac{T_i}{W_{\mathrm{dg}}}$$

Achievable climb gradient, landing

$$q_L = \frac{-1}{L/D_L} + \left(1 - \frac{1}{N_e}\right)\frac{T_i}{W_{\mathrm{dg}}(1 - R_{\mathrm{fr}})}$$

e) Takeoff field length

$$D_{\mathrm{To}} = \frac{1.3047 \cdot W_{\mathrm{dg}}^2}{C_{L\max} T_i S} + 12.0\sqrt{\frac{W_{\mathrm{dg}}}{C_{L\max} S}}$$

Landing field length

$$D_L = 121.92 + \frac{7.3664 \cdot W_{\mathrm{dg}}(1 - R_{\mathrm{fr}})}{C_{L\max} S}$$

Acknowledgments

This research has benefited greatly from the support of the National Natural Science Foundation of China under grant 10702055, the Ph.D. Programs Foundation of Ministry of Education of China under grant 20070699047, and the China Postdoctoral Science Foundation under grant 20070410383.

References

[1] Huang, C. H., and Bloebaum, C. L., "Multi-Objective Pareto Concurrent Subspace Optimization for Multidisciplinary Design," *AIAA Journal*, Vol. 45, No. 8, 2007, pp. 1894–1906. doi:10.2514/1.19972

[2] Sobieszczanski-Sobieski, J., "A Step from Hierarchic to Non-Hierarchic Systems," NASA CP-3031, 1989.

[3] Bloebaum, C. L., "Formal and Heuristic System Decomposition in Structural Optimization," NASA CR-4413, 1991.

[4] Renaud, J. E., and Gabriele, G. A., "Second Order Based Multidisciplinary Design Optimization Algorithm Development," *Advances in Design Automation*, Vol. 65, No. 2, 1993, pp. 347–357.

[5] Renaud, J. E., and Gabriele, G. A., "Improved Coordination in Non-Hierarchic System Optimization," *AIAA Journal*, Vol. 31, No. 12, 1993, pp. 2367–2373. doi:10.2514/3.11938

[6] Renaud, J. E., and Gabriele, G. A., "Approximation in Non-Hierarchic System Optimization,". *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 198–205. doi:10.2514/3.11967

[7] Sellar, R. S., Batill, S. M., and Renaud, J. E., "Response Surface Based, Concurrent Subspace Optimization for Multidisciplinary System Design," AIAA Paper 96-0714, 1996.

[8] Tappeta, R. V., and Renaud, J. E., "Multiobjective Collaborative Optimization," *Journal of Mechanical Design*, Vol. 119, No. 3, Sept. 1997, pp. 403–411. doi:10.1115/1.2826362

[9] McAllister, C. D., Simpson, T. W., and Yukesh, M., "Goal Programming Applications in Multidisciplinary Design Optimization," AIAA Paper 2000-4717, 2000.

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

[11] Orr, S. A., and Hajela, P., "Genetic Algorithm Based Collaborative Optimization of a Tiltrotor Configuration," AIAA Paper 2005-2285, 2005.

[12] Aute, V., and Azarm, S. A., "Genetic Algorithms Based Approach for Multidisciplinary Multiobjective Collaborative Optimization," AIAA Paper 2006-6953, 2006.

[13] Huang, C. H., and Bloebaum, C. L., "Incorporation of Preferences in Multi-Objective Concurrent Subspace Optimization for Multidisciplinary Design," AIAA Paper 2004-4548, 2004.

[14] Parashar, S., and Bloebaum, C. L., "Multi-Objective Genetic Algorithm Concurrent Subspace Optimization (MOGACSSO) for Multidisciplinary Design," AIAA Paper 2006-2047, 2006.

[15] Kim, I. Y., and de De Weck, O. L., "Adaptive Weighted-sum Method for Bi-Objective Optimization: Pareto Front Generation," *Structural and Multidisciplinary Optimization*, Vol. 29, No. 2, Feb. 2005, pp. 149–158. doi:10.1007/s00158-004-0465-1

[16] Kim, I. Y., and De Weck, O. L., "Adaptive Weighted Sum Method for Multiobjective Optimization," AIAA Paper 2004-4322, 2004.

[17] Kreisselmeier, G., Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *Regelungstechnik*, Vol. 27, No. 3, 1979, pp. 76–79.

[18] Sobieszczanski-Sobieski, J., "Sensitivity of Complex, Internally Coupled Systems," *AIAA Journal*, Vol. 28, No. 1, 1990, pp. 153–160. doi:10.2514/3.10366

[19] Huang, C. H., "Development of Multi-Objective Concurrent Subspace Optimization and Visualization Methods for Multidisciplinary Design," Ph.D. Dissertation, The State University of New York, New York, 2003.

[20] Deb, K., Pratap, A., and Meyarivan, T., "Constrained Test Problems for Multi-Objective Evolutionary Optimization," *1st International Conference on Evolutionary Multi-Criterion Optimization*, Springer–Verlag, London, 2001, pp. 284–298.

[21] Lewis, K., "An Algorithm for Integrated Subsystem Embodiment and System Synthesis," Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, Georgia, August 1997.

[22] Lewis, K., and Mistree, F., "Designing Top-Level Aircraft Specifications: A Decision-Based Approach to a Multiobjective, Highly Constrained Problem," AIAA Paper 1995-1431, 1995.

T. Zang
Associate Editor