10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
30 August - 1 September 2004, Albany, New York

AIAA 2004-4322

Adaptive Weighted Sum Method for Multiobjective Optimization

Il Yong Kim[*]
Queen's University, Kingston, Ontario, K7L 3N6, Canada

Olivier de Weck[†]
Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139, USA

This paper presents an adaptive weighted sum method for multiobjective optimization problems. The authors developed the bi-objective adaptive weighted sum method, which determines uniformly-spaced Pareto optimal solutions, finds solutions on non-convex regions, and neglects non-Pareto optimal solutions. However, the method could solve only problems with two objective functions. In this work, the bi-objective adaptive weighted sum method is extended to problems with more than two objective functions. In the first phase, the usual weighted sum method is performed to approximate Pareto surfaces quickly, and a mesh of Pareto front patches is identified. Each Pareto front patch is then refined by imposing additional equality constraints that connect the pseudo nadir point and the expected Pareto optimal solutions on a piecewise planar surface in the objective space. It is demonstrated that the method produces a well-distributed Pareto front mesh for effective visualization and finds solutions in non-convex regions. Two numerical examples and a simple structural optimization problem are solved as case studies.

Nomenclature

$\mathbf{J}$	=	objective function vector
$\mathbf{x}$	=	design vector
$\mathbf{p}$	=	vector of fixed parameters
$\mathbf{g}$	=	inequality constraint vector
$\mathbf{h}$	=	equality constraint vector
m	=	number of objectives
α_i	=	ith weighting factor
$\bar{J}_i$	=	normalized objective function
$\mathbf{J}^{Utopia}$	=	utopia point
$\mathbf{J}^{Nadir}$	=	nadir point
$\mathbf{J}^{i*}$	=	ith anchor point $\mathbf{J}^{i*}$
$\mathbf{P}^{j}$	=	position vector of the jth expected solution on the piecewise linearized plane

I. Introduction

Design optimization is to seek the best design that minimizes the objective function by changing design variables while satisfying design constraints. During design optimization one often needs to consider several design criteria or objective functions simultaneously. For example, we may want to maximize range and payload mass while trying to minimize lifecycle cost for an airplane design. When more than one design objective is associated, the design

[*] Assistant Professor, Department of Mechanical and Materials Engineering, McLaughlin Hall 305, Queen's University.

[†] Assistant Professor, Department of Aeronautics and Astronautics and Engineering Systems Division, Room 33-410, MIT, AIAA senior member.

problem becomes multiobjective, in which case the usual design optimization for a scalar objective function cannot be used.

Multiobjective optimization can be stated as follows:

$$\begin{aligned} \min \quad & \mathbf{J}(\mathbf{x}, \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq 0 \\ & \mathbf{h}(\mathbf{x}, \mathbf{p}) = 0 \\ & x_{i,LB} \leq x_i \leq x_{i,UB} \quad (i = 1, \ldots, n) \ , \end{aligned} \tag{1}$$

where the objective function vector $\mathbf{J}$ is a function of design vector $\mathbf{x}$ and a fixed parameter vector $\mathbf{p}$; $\mathbf{g}$ and $\mathbf{h}$ are inequality and equality constraints; and $x_{i,LB}$ and $x_{i,UB}$ are the lower and upper bounds for the ith design variable, respectively.

Stadler[1,2] applied the notion of Pareto optimality to the fields of engineering and science in the 1970s. The most widely-used method for multiobjective optimization is the weighted sum method. The method transforms multiple objectives into an aggregated objective function by multiplying each objective function by a weighting factor and summing up all weighted objective functions:

$$J_{\text{weighted sum}} = w_1 J_1 + w_2 J_2 + \cdots + w_m J_m \tag{2}$$

where w_i $(i = 1, \cdots, m)$ is a weighting factor for the ith objective function (potentially also dividing each objective by a scaling factor, i.e. $w_i = \alpha_i / sf_i$). If $\sum_{i=1}^{m} w_i = 1$ and $0 \leq w_i \leq 1$, the weighted sum is said to be a convex combination of objectives. Each single objective optimization determines one particular optimal solution point on the Pareto front. The weighted sum method then changes weights systemically, and each different single objective optimization determines a different optimal solution. The solutions obtained approximate the Pareto front. Initial work on the weighted sum method can be found in Zadeh[3]. Koski[4] applied the weighted sum method to structural optimization. Marglin[5] developed the ε-constraint method, and Lin[6] developed the equality constraint method. Heuristic methods are also used for multiobjective optimization: Suppapitnarm[7] applied simulated annealing to multiobjective optimization, and multiobjective optimization by Genetic Algorithms can be found in Goldberg[8], and Fonseca and Fleming[9] among others.

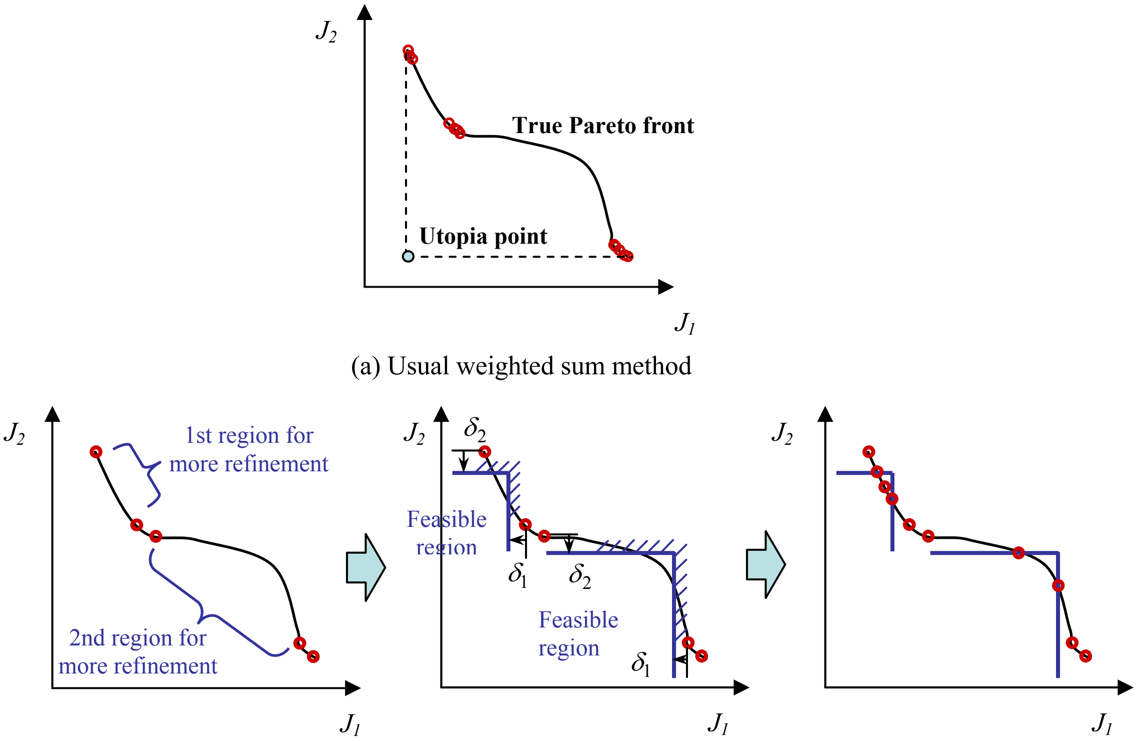
Das and Dennis[10] developed the NBI (Normal Boundary Intersection) method, in which sub-optimizations are performed on normal lines to the utopia hyperplane that is defined and bounded by all anchor points. The NBI method produces well-distributed solutions, and it is easily scalable to n-dimensional problems. The method can also determine Pareto optimal solutions in non-convex regions, which the weighed sum method misses. The weak points of the method are (1) in highly nonlinear problems, it is hard to obtain optimal solutions due to equality constraints, (2) non-Pareto optimal solutions (dominated solutions) are also obtained, and a Pareto filtering must be used to filter out those solutions, and (3) in high dimensional problems (more than two objective functions), the projection of the utopia plane does not cover the entire Pareto front, and some Pareto front regions are not discovered by this method. Messac and Mattson[11,12] used physical programming for generating Pareto fronts for concept selection. They also developed the normal constraint method[13], which generates uniformly distributed solutions along the Pareto front without missing any Pareto front regions. The method can be extended to n-dimensional problems.

The well-known drawbacks of the weighted sum method, as discussed in a number of studies[11,14,15], are that (1) often the optimal solution distribution is not uniform, and that (2) more seriously, optimal solutions in non-convex regions are not detected. The adaptive weighted sum (AWS) method[16] was developed recently by the authors to address these two drawbacks. By imposing additional inequality constraints in the usual weighted sum method, the optimization is performed only in a newly-defined feasible region where more exploration is needed. The adaptive weighted sum method successfully solves multiobjective optimization problems: the AWS method produces well-distributed solutions, finds Pareto optimal solutions in non-convex regions, and neglects non-Pareto optimal solutions. The AWS method, however, was previously only applicable to bi-objective optimization problems. Therefore, we will refer to the previous technique as the "bi-objective adaptive weighted sum method" to differentiate it from the generalized multiobjective adaptive weighted sum method presented here.

In this work, the bi-objective adaptive weighted sum method is extended to multiobjective optimization problems with more than two objective functions. Unfortunately, the additional inequality constraints that are used in the bi-objective adaptive weighted sum method are not suitable for higher-dimensional multiobjective optimization. The reason is that the Pareto-front patches (extension of segments in two dimensions) to be constructed by additional constraints can be arbitrarily-shaped hyperplanes in multidimensional problems, and it is difficult to define feasible regions for refinement by inequality constraints alone. In the multiobjective adaptive weighted sum method, additional equality constraints are introduced that connect the pseudo-nadir point and the expected locations of Pareto optimal solutions on the piecewise linearized plane in the objective space. Sub-optimizations for further refinement are conducted along these lines (equality constraints) determining solutions near desired positions, which leads to a well-distributed mesh representation of the Pareto.

II. Multiobjective Adaptive Weighted Sum Method: Fundamental Concepts

The fundamental philosophy of the adaptive weighted sum method is to adaptively refine the Pareto front. In the first stage, the method determines a rough profile of the Pareto front. By estimating the size of each Pareto patch (line segment in the case of two-dimensional problems), the regions for further refinement in the objective space are determined. In the subsequent stage, only these regions are specified as feasible domains for sub-optimization by assigning additional constraints. In the bi-objective adaptive weighted sum method, the feasible domain for further exploration is determined by specifying two inequality constraints. The usual weighted sum method is then performed as sub-optimization in these feasible domains obtaining more Pareto optimal solutions. When a new set of Pareto optimal solutions are determined, the Pareto patch size estimation is again performed to determine the regions for further refinement. These steps are repeated until a termination criterion is met. Figure 1 compares the typical weighted sum method and the bi-objective adaptive weighted sum method for a sample problem that has a relatively flat region and a non-convex region.



(a) Usual weighted sum method

(b) Procedure of the bi-objective adaptive weighted sum method

Figure 1. The concept and procedure of the adaptive weighted sum method.

We found that the inequality constraints as boundaries for constructing feasible regions are not suitable for optimization problems with more than two objective functions. Feasible regions for further refinement in the two-dimensional case can be defined easily by laying two inequality constraints that are parallel to each of the axes with prescribed offset distances from the end points, because the Pareto front is a two-dimensional curve, and there are always only two end points for each Pareto-front segment.

In higher-dimensional cases, however, the Pareto front becomes a surface (for three objective functions) or a hypersurface (for more than three objective functions), and it becomes difficult to impose constraints such that sub-optimizations are performed only in a selected Pareto-front patch and to adaptively refine the patches. This is because Pareto-front patches may have arbitrary shapes, and the number of edges for each Pareto-front patch may vary. In addition, when the number of vertices is larger than the dimension of the objective space, all vertices or their connecting edges may not lie in the same (hyper-) plane, and it becomes even more difficult to impose constraints for sub-optimization of further refinement and to perform adaptive refinement in the following stages. Indeed, the problems encountered then resemble adaptive remeshing in the Finite Element Method, but in higher dimensions. There has been very extensive research conducted in this field for decades, and it remains to be seen whether the sophisticated and sometimes complicated adaptive remeshing techniques of the FEM are applicable to Pareto front generation. It is also important to note that the FEM techniques can be applied only to three-dimensional problems.

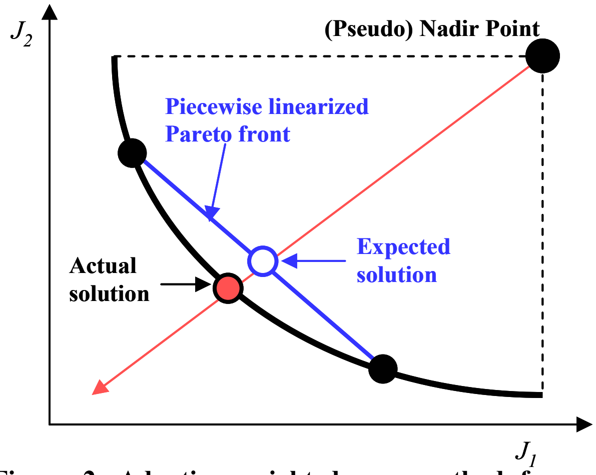


Figure 2. Adaptive weighted sum method for multidimensional problems (2-D representation).

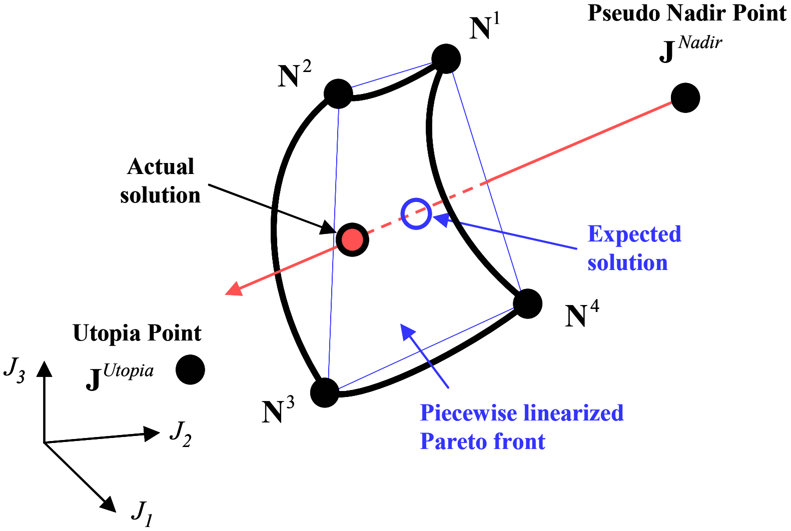


Figure 3. Adaptive weighted sum method for multidimensional problems (3-D representation).

In this work, we adopt equality constraints to define feasible regions for further refinement, which is more robust for obtaining well-distributed solutions in multidimensional problems than inequality constraints. Adding equality constraints increases the likelihood of entrapment in local minima, but also facilitates the adaptive procedure by simplifying patch refinement. Although Pareto front patches of any shape can be used, we demonstrate the method with quadrilateral patches with applications to three-dimensional problems in this paper.

In the first stage, the approximate shape of the Pareto front is determined by using the usual weighted sum method. Pareto front patches are then identified, and patches for further refinement are selected on the basis of the patch size. Sub-optimization is performed only in the selected patches by specifying additional equality constraints. Figure 2 shows the concept of the multiobjective adaptive weighted sum method with equality constraints for multiobjective optimization. In the bi-objective adaptive weighted sum method, feasible regions for further refinement are defined by two inequality constraints (Fig. 1), but in the multiobjective adaptive weighted sum method, one or several equality constraints are specified that connect the pseudo nadir point and expected solutions on the piecewise linearized Pareto front. Actual solutions obtained will be on the equality constraint line, but they may be located in different positions from the expected solutions as shown in the figure.

The equality constraint, which is represented by a line, allows us to have great control over the position of new Pareto optimal solutions obtained, and this simplifies adaptive refinement. In the three-dimensional case, the Pareto front becomes a surface, and the linearized Pareto front patch is represented by four line segments that connect four vertices, as shown in Fig. 3. When solutions are obtained in all patches for further refinement, a new set of Pareto front patches are identified, and a patch-size evaluation is performed to determine where to further refine. These steps are repeated until a termination criterion is met. The complete and detailed procedure is presented in the following section.

III. Multiobjective Adaptive Weighted Sum Method: Procedures

In this section, we detail the procedure for implementing the multiobjective adaptive weighted sum method.

[Step 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 $\mathbf{J}^U$ is defined as

$$\mathbf{J}^{Utopia} = [J_1(\mathbf{x}^{1*}) \;\; J_2(\mathbf{x}^{2*}) \;\; \cdots \;\; J_m(\mathbf{x}^{m*})], \tag{3}$$

and the pseudo nadir point $\mathbf{J}^{Nadir}$ is defined as

$$\mathbf{J}^{Nadir} = [J_1^{Nadir} \;\; J_2^{Nadir} \;\; \cdots \;\; J_m^{Nadir}] \tag{4}$$

where m is the number of objective functions or the dimension of the objective space, and each component J_i^{Nadir} is determined by

$$J_i^{Nadir} = \max[J_i(\mathbf{x}^{1*}) \;\; J_i(\mathbf{x}^{2*}) \;\; \cdots \;\; J_i(\mathbf{x}^{m*})]. \tag{5}$$

The ith anchor point $\mathbf{J}^{i*}$ is defined as

$$\mathbf{J}^{i*} = [J_1(\mathbf{x}^{i*}) \;\; J_2(\mathbf{x}^{i*}) \;\; \cdots \;\; J_m(\mathbf{x}^{i*})]. \tag{6}$$

Now the normalized objective function $\bar{J}_i$ is obtained as

$$\bar{J}_i = \frac{J_i - J_i^{Utopia}}{J_i^{Nadir} - J_i^{Utopia}}. \tag{7}$$

[Step 2] Perform multiobjective optimization using the usual weighted sum approach with a small number of divisions, n_{initial}.

For three objective functions, the weighted single objective function J_{Total} is obtained as

$$\begin{aligned} J_{Total} &= \alpha_2\left[\alpha_1 J_1 + (1-\alpha_1)J_2\right] + (1-\alpha_2)J_3 \\ &= \alpha_1\alpha_2 J_1 + (1-\alpha_1)\alpha_2 J_2 + (1-\alpha_2)J_3, \qquad \alpha_i \in [0,1] \end{aligned} \tag{8}$$

where α_i is the ith weighting factor. As a general form, the weighted single objective function of m objective functions, J_{Total}^m, is determined by

$$J_{Total}^m = \alpha_{m-1} J_{Total}^{m-1} + (1-\alpha_{m-1})J_m \;\;, \qquad m \geq 2 \tag{9}$$

where $J_{Total}^1 \equiv J_1$.Note that *m-1* weighting factors are needed to explore an *m*-dimensional objective space.

The uniform step size of the ith weighting factor α_i is determined by the number of initial divisions along the ith objective dimension:

$$\Delta\alpha_i = \frac{1}{n_{\text{initial, i}}}, \qquad i = 1, \cdots, m-1 \tag{10}$$

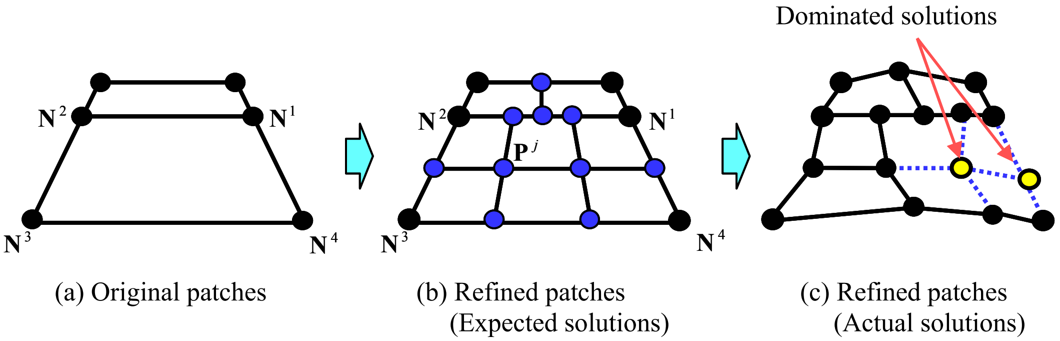
where m is the number of objective functions.

In this work, we use the same step size for all weighting factors. There is a scheme that systemically determines each weighting factor and helps produce well-distributed solutions[10]. In the adaptive weighted sum method, however, the usual step size strategy Eq. (10) can be used because this approximate multiobjective optimization is conducted only once, after which adaptive refinement is conducted.

[Step 3] Delete nearly overlapping solutions. It occurs often that several nearly identical solutions are obtained when the weighted sum method is used. The Euclidian distances between these solutions are nearly zero, and among these, only one solution is needed to represent the Pareto front. In the computer implementation, if the distance among solutions in the objective space is less than a predetermined distance (ε), then all solutions except one are deleted.

[Step 4] Identify Pareto-front patches. Patches of any shape can be used, but in this work we use quadrilateral patches in three-dimensional problems. Four Pareto-optimal solutions become the four nodes of each patch, and edges are line segments that connect two neighboring nodes of each patch. Constructing and maintaining meshes on the Pareto front may be tedious, but there are two advantages of using a mesh, which will be discussed in detail in the following sections: (1) patches play the role of primitives for further refinement for subsequent stages, as will be seen in Step 5, and (2) if only non-dominated solution points are displayed, it is difficult to visualize and interpret the shape of the Pareto front. A mesh representation makes it very easy to visualize the Pareto surface as in the case of finite element meshes.

[Step 5] Stage = Stage + 1. Determine the layout for further refinements in each of the Pareto-front patches. The larger the patch is, the more it needs to be refined. Figure 4 shows an example of refinement, in which a patch is composed of four nodes in three dimensional objective space, as will be studied in this paper. Because the lower patch is larger, it is refined more than the upper one. In each mesh, the locations of expected solutions are determined by interpolation, and sub-optimizations are conducted along the lines that connect the pseudo nadir point and the expected solutions. The actual solutions may be different from expected solutions, and there can be dominated solutions, which must be deleted by a Pareto filter.



(a) Original patches

(b) Refined patches (Expected solutions)

(c) Refined patches (Actual solutions)

Figure 4. Adaptive refinement procedure.

The position vector of the jth expected solution on the piecewise linearized plane ($\mathbf{P}^j$) is obtained as the weighed sum of the four vectors of the nodal solutions as

$$\mathbf{P}^j = \beta_1\mathbf{N}^1 + \beta_2\mathbf{N}^2 + \beta_3\mathbf{N}^3 + \beta_4\mathbf{N}^4 \quad , \qquad \beta_i \in [0,1] \tag{11}$$

where $\mathbf{N}^i$ is the position vector the ith node of a Pareto-front patch (Fig. 4(b)), and β_i is a weighting factor for interpolation.

The normalized vector of $\mathbf{P}^j$ is obtained as

$$\bar{P}_i^j = \frac{P_i^j - J_i^{Utopia}}{J_i^{Nadir} - J_i^{Utopia}} \tag{12}$$

where P_i^j is the ith coordinate of the jth expected solution on the piecewise linearized (hyper-)plane. The refinement level, which is represented by the step size of weighting factors, is determined based on the relative average length of the patch in each direction.

[Step 6] Impose an additional equality constraint for each expected solution and conduct a sub-optimization with the weighted sum method. For the jth normalized expected solution, $\bar{\mathbf{P}}^j$, the sub-optimization problem is defined as

$$\begin{aligned} &\text{minimize} && \mathbf{w}_i \cdot \bar{\mathbf{J}}(\mathbf{x}) \\ &\text{subject to} && \frac{(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir}) \cdot (\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{Nadir})}{\left|\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir}\right| \left|\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{Nadir}\right|} = 1 \\ & && \bar{\mathbf{h}}(\mathbf{x}) = 0 \\ & && \bar{\mathbf{g}}(\mathbf{x}) \le 0 \end{aligned} \tag{13}$$

where $\mathbf{w}_i = -(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir})$ is a vector of weighting factors, and $\bar{\mathbf{h}}(\mathbf{x})$ and $\bar{\mathbf{g}}(\mathbf{x})$ are normalized equality and inequality constraint vectors. Note that the normalized nadir point $\bar{\mathbf{J}}^{Nadir}$ is a vector whose components are one, i.e. $\bar{\mathbf{J}}^{Nadir} = (1, 1, \cdots, 1)$.

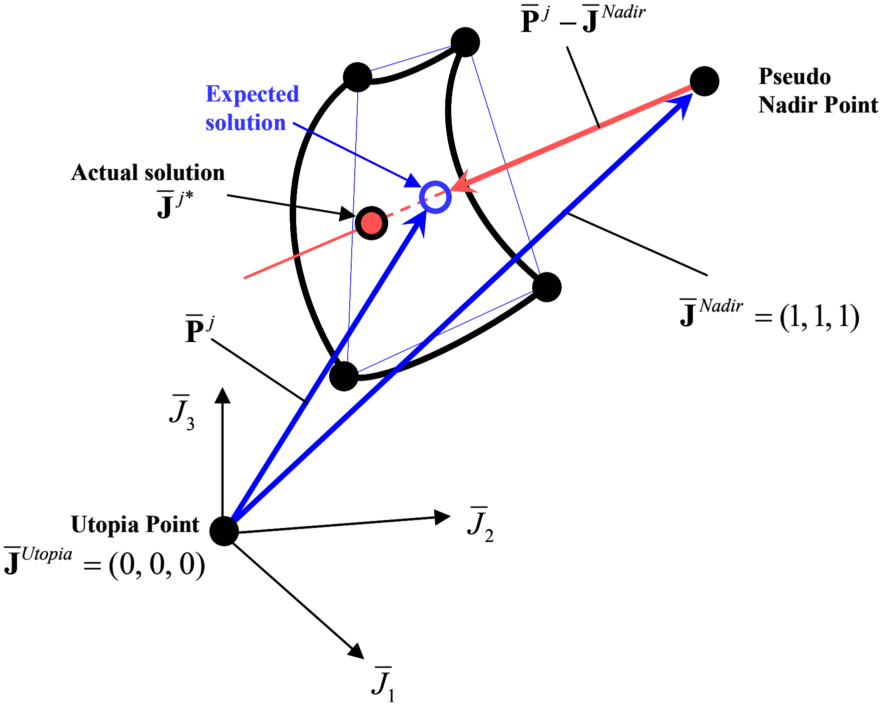


Figure 5. Configuration of an additional equality constraint for refinement (3-D representation).

The equality constraint $\left((\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir})\cdot(\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{Nadir})\right)/\left(\left|\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir}\right|\left|\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{Nadir}\right|\right) = 1$ makes the two vectors $\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir}$ and $\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{Nadir}$ be collinear in the objective space. This constraint therefore ensures that the solution is obtained only along the line $\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir}$, which connects the expected solution on the piecewise linearized plane and the pseudo nadir point. The objective function $-(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir})\cdot\bar{\mathbf{J}}(\mathbf{x})$ is a scalar function to be minimized determining the solution that is nearest to the utopia point in the direction of $-(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir})$.

The actual solution obtained for the jth normalized expected solution, $\bar{\mathbf{P}}^{j*}$, would be different from the expected solution. In Figure 5, the origin of the vector $\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{Nadir}$ is actually $(0, 0, 0)$ but moved for better visualization.

[Step 7] Perform Pareto filtering. In the bi-objective adaptive weighted sum method, non-Pareto optimal solutions are automatically rejected, so the filtering is not needed. In the multiobjective adaptive weighted sum method, however, any solution that lies on the equality constraint is feasible, and non-Pareto optimal solutions may be obtained. In each step, it is necessary to perform Pareto filtering to obtain the true Pareto front.

[Step 8] Delete overlapping solutions. Identify Pareto-front patches with all Pareto optimal solutions including newly obtained solutions in the previous steps. If a termination criterion is met, stop; otherwise go to Step 5. Several types of termination criteria may be used: (1) the number of stages reaches a prescribed number; (2) the size of largest Pareto-front patch falls below a prescribed value; (3) the standard deviation among the sizes of all Pareto-front patches falls below a prescribed value. In this work, the maximum number of stages is used as the termination criterion.

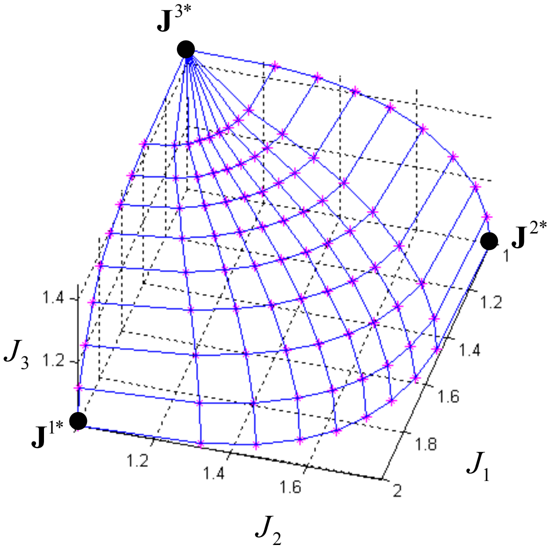


Figure 6. Pareto front obtained by the usual weighted sum method for Example 1.

IV. Numerical Examples

Three numerical examples are presented in this section to demonstrate the performance of the multiobjective adaptive weighted sum method. All examples are three-dimensional problems. Sequential Quadratic Programming (SQP) in MATLAB is used for every optimization.

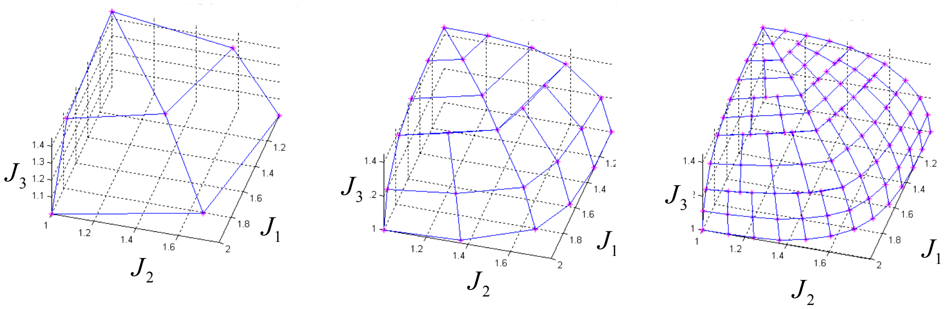
A. Example 1: Convex Pareto Front

The first example is a multiobjective maximization problem whose Pareto front is convex. The problem statement is

$$\begin{aligned} &\text{maximize } [J_1\ J_2\ J_3]^T \\ &\text{subject to } \ x_1^4 + 2x_2^3 + 5x_3^2 \le 1 \\ &\qquad\qquad J_1 = x_1 \\ &\qquad\qquad J_2 = x_2 \\ &\qquad\qquad J_3 = x_3 \\ &\qquad\qquad x_i \ge 0\ \ (i = 1, 2, 3). \end{aligned} \tag{14}$$

The Pareto front of this problem is convex, but the curvatures are different in three axes. Figure 6 shows the Pareto front obtained by the usual weighted sum method. The step sizes for the two weighting factors in Eq. (8), $\Delta\alpha_1$ and $\Delta\alpha_2$, are 1/9. The number of Pareto optimal solutions on the front is 100 with ten solutions coincident on the top vertex. These solutions represent a 9×9 Pareto-front mesh. The patch size varies greatly according to the position: the meshes on the top are slender, and the meshes near to the two anchor points $\mathbf{J}^{1*}$ and $\mathbf{J}^{3*}$ on the bottom are relatively large and slender while the patches in the middle region of the front are small and nearly square.

Figure 7 shows the three stages of the Pareto front evolution produced by the multiobjective adaptive weighted sum method. In the first stage, the usual three-dimensional weighted sum method with $\Delta\alpha_1 = \Delta\alpha_2 = 0.5$ is performed obtaining a coarse 2×2 Pareto-front mesh. Based on this initial Pareto front, the relative size of each Pareto-front patch is estimated, and adaptive refinement is performed. Note that the Pareto front is not symmetric in any direction, and this is the reason that mesh refinement in the second stage is asymmetric. The number of Pareto front patches is 72 in the third stage. Contrary to the Pareto front representation by the usual weighted sum method in Fig. 6, the mesh shape and size are quite uniform.



(a) Stage 1 **(b) Stage 2** **(c) Stage 3**

Figure 7. Pareto front obtained by the multiobjective adaptive weighted sum method for Example 1.

In this work, the goal of adaptive refinement is to have not only uniformly distributed solutions but also solutions that form a well-shaped mesh layout. Multiobjective optimization is conducted in order to present trade-off information to engineers such that best decisions can be made. In the two-dimensional case, it is not difficult to interpret a Pareto front that is represented only by solution points. In higher dimensions, however, the point-based representation is often hard to interpret. By maintaining meshes as in this work, further adaptive refinement considering the mesh size can be performed systematically, and it is very easy to visualize the Pareto front obtained – at least in three objective dimensions at-a-time.

B. Example 2: Non-convex Pareto Front with Dominated Solutions

In the previous example, the Pareto front was convex, and the problem associated with the usual weighted sum method was only that we could not obtain evenly-distributed solutions, or the mesh layout was not uniform. In this example, we solve a multiobjective problem whose Pareto front has non-convex regions and is disconnected due to dominated solutions. The problem statement is

$$\begin{aligned}
&\text{maximize} \;\; [J_1 \; J_2 \; J_3]^T \\
&\text{subject to} \;\; -\cos x_1 - e^{-x_2} + x_3 \le 0 \\
&\qquad\qquad J_1 = x_1 \\
&\qquad\qquad J_2 = x_2 \\
&\qquad\qquad J_3 = x_3 \\
&\qquad\qquad 0 \le x_1 \le \pi \\
&\qquad\qquad x_2 \ge 0 \\
&\qquad\qquad x_3 \ge 1.2 \;\; .
\end{aligned} \tag{15}$$

Figure 8 shows the Pareto front of this problem including the dominated solution region from two different viewpoints, which was generated with MATLAB executing a full factorial evaluation (which would not be feasible for higher dimensional design spaces). The boundary of the Pareto front is composed of three curves: the curve between $\mathbf{J}_1^*$ and $\mathbf{J}_3^*$ is convex with a gap due to a dominated solution region, but the other two curves are not convex. There is a dominated region in the middle, which looks like a valley. The usual weighted sum method is used to determine the Pareto front with a 20×20 mesh. As appears in Fig. 9, solutions are obtained only on the convex curve between $\mathbf{J}^{1*}$ and $\mathbf{J}^{3*}$ and on the short convex curve segment near $\mathbf{J}^{2*}$. Dominated solutions are not obtained, however most of the Pareto front, which is non-convex, is not revealed by this method.

The multiobjective adaptive weighted sum method is performed to find the Pareto front adaptively. In the first stage, the usual weighted sum method with $\Delta\alpha_1 = \Delta\alpha_2 = 1$ determines the three anchor points forming an approximate Pareto front (Fig. 10). In the second stage, the overall shape of the non-convex Pareto front is found; a result which cannot be obtained no matter how many solutions are found by the usual weighted sum method. Because equality constraints are used, dominated solutions (non-Pareto optimal solutions) are obtained by the multiobjective adaptive weighted sum method. A Pareto filtering step is conducted in each stage. In the final stage, the Pareto front in its "entirety" is determined. We can clearly see a dominated region in the middle, and the mesh representation makes it easy to interpret the surface.

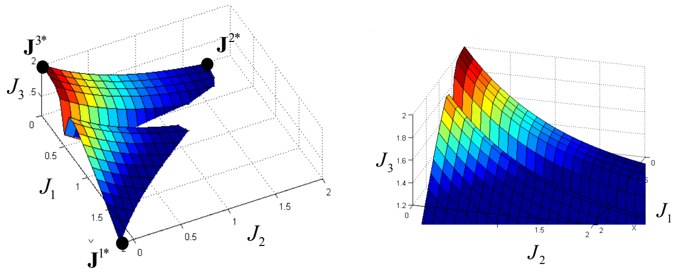


Figure 8. Pareto front profile (including the dominated region) for Example 2.

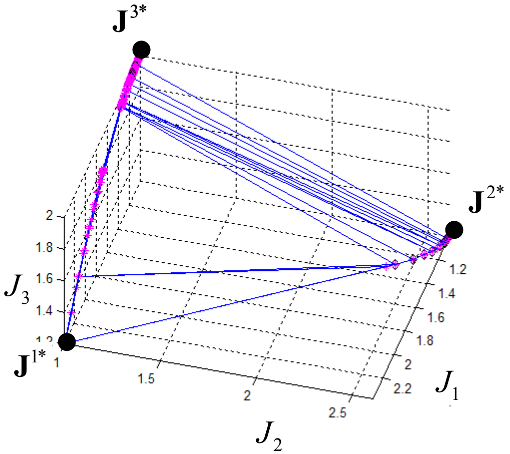
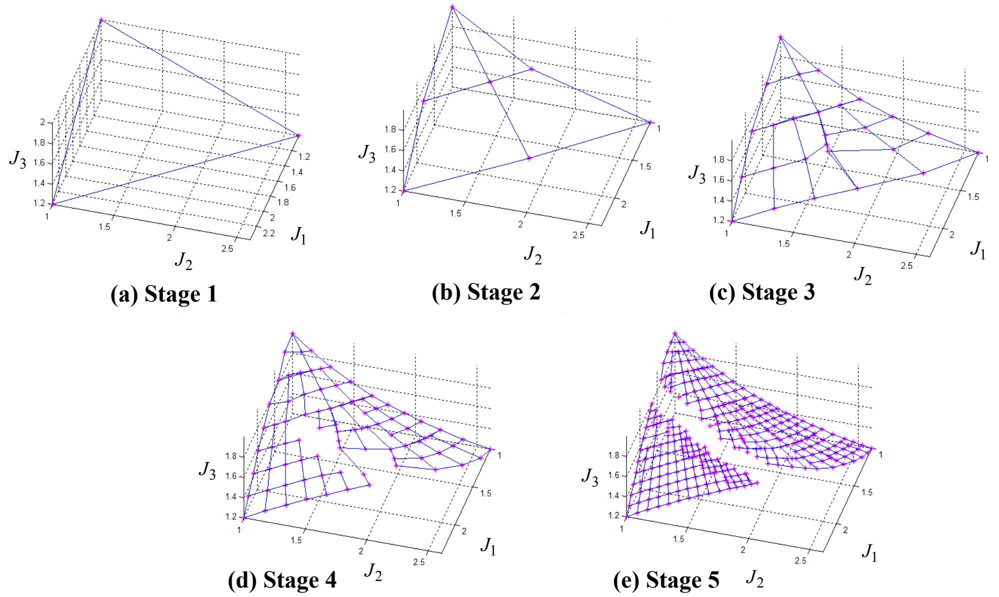


Figure 9. Pareto front obtained by the usual weighted sum method for Example 2.



(a) Stage 1

(b) Stage 2

(c) Stage 3

(d) Stage 4

(e) Stage 5

Figure 10. Pareto front obtained by the multiobjective adaptive weighted sum method for Example 2.

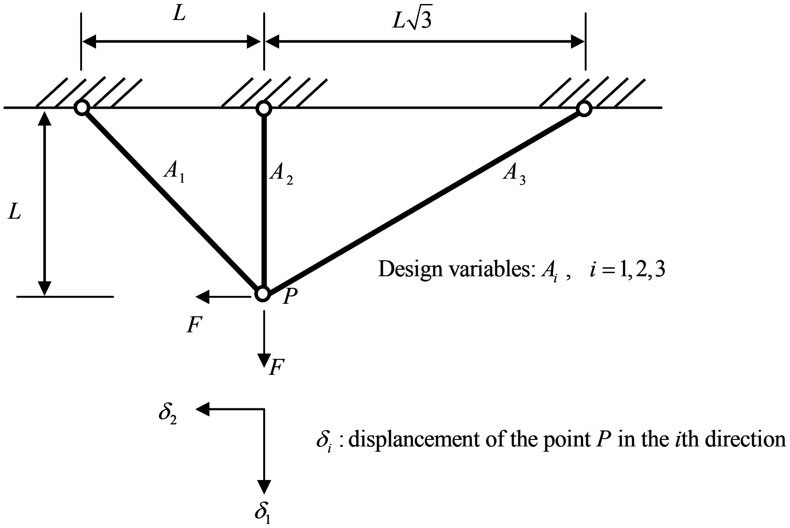
C. Example 3: Three-bar Problem

Finally, the multiobjective adaptive weighted sum method is applied to the three-bar problem[15]. The geometry and material properties are presented in Fig. 11. The upper end of each bar is fixed, and a horizontal and vertical load are applied at Point P. The objective functions to be minimized are the total volume, stress in Truss 1, and stress in Truss 3. The mathematical problem statement is

$$\begin{aligned} &\text{minimize} \;\; [\text{Volume}(\mathbf{A}) \; \sigma_1(\mathbf{A}) \; \sigma_3(\mathbf{A})]^T \\ &\text{subject to} \;\; -200 \text{ MPa} \le \sigma_i(\mathbf{A}) \le 200 \text{ MPa} \; (i=1,2,3) \\ &\qquad\qquad\;\; 0.1 \text{ cm} \le A_i \le 2 \text{ cm} \;\; (i=1,2,3) \end{aligned} \tag{16}$$

where the design variable A_i is the cross-sectional area of the ith truss.

The Pareto front of this problem is non-convex, and some part of the objective space is dominated. Figure 12 shows the results obtained with the multiobjective adaptive weighted sum method. For better visualization, the graph is inverted, i.e. all three objective functions are multiplied by minus one (it is often easier to interpret the Pareto surface when we view it from outside the feasible range rather than from the inside). The pseudo nadir point is the origin where three reference planes in the figure meet. In the first stage, the approximate Pareto front is represented by 6 patches. The multiobjective adaptive weighted sum method is then applied determining more refined Pareto front meshes until convergence is reached in three stages.

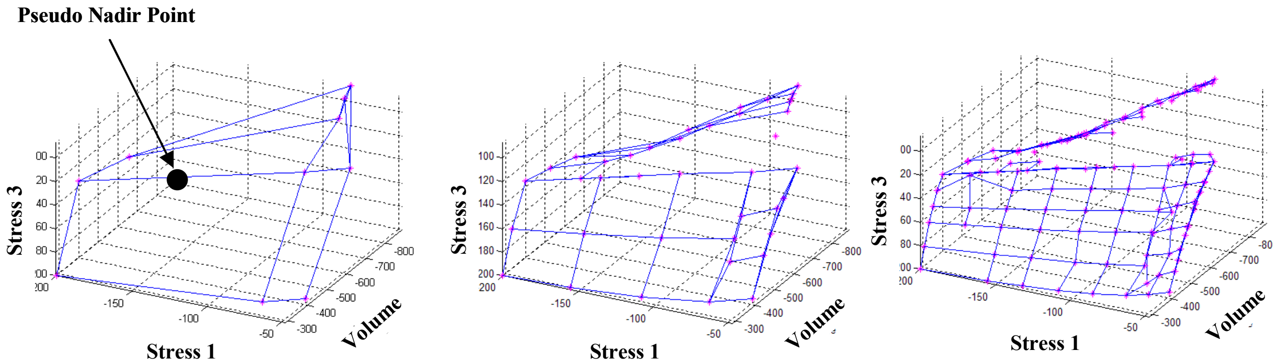

L
$L\sqrt{3}$
A_1
A_2
A_3
L
Design variables: A_i , $i=1,2,3$
P
F
F
δ_2
δ_i : displancement of the point P in the ith direction
δ_1

$F = 20\,kN$, $L = 100\,cm$, $E = 200\,GPa$

$\sigma_{\text{upper limit}} = 200\,MPa$, $\sigma_{\text{lower limit}} = -200\,MPa$

$A_{\text{upper limit}} = 2\,cm^2$, $A_{\text{lower limit}} = 0.1\,cm^2$

Figure 11. Three-bar problem.



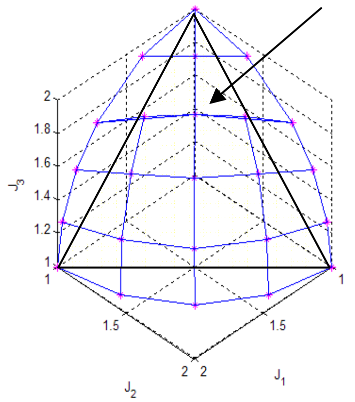
(a) Stage 1 **(b) Stage 2** **(c) Stage 3**

Figure 12. Pareto front obtained by the multiobjective adaptive weighted sum method for Example 3.

V. Discussions

The multiobjective adaptive weighted sum method effectively solves multiobjective optimization problems with more than two objective functions. In the bi-objective adaptive weighted sum method, which is applicable to only optimizations with two objective functions, inequality constraints are used to specify regions for further refinement. In the multiobjective adaptive weighted sum method, on the other hand, equality constraints are used and the method is scalable to *n*-dimensional problems. The equality constraints allow us to decide where to obtain additional solutions, and this makes the Pareto front mesh well conditioned.

There are two important issues in using the multiobjective adaptive weighted sum method. First, adaptive refinement is conducted only within the Pareto front that is determined by the usual weighted sum method in the first stage. If the first stage optimization misses some regions on the Pareto front, they are not discovered in the following adaptive refinement stages. In general, the usual weighted sum method finds most parts of the Pareto front reliably and quickly. This is, in addition to its adaptivity, a distinctive advantage of the adaptive weighted sum method, which utilizes the usual weighted sum method in the first stage, over the NBI method. Figure 13 shows a Pareto surface that is obtained by the adaptive weighted sum method (based on the first stage usual weighted sum method) and the utopia plane for the NBI method. The view-direction is rotated such that it is normal to the utopia plane. As can be seen in the figure, the NBI method cannot determine the three regions that are not covered by the normal projection of the utopia plane. The main difference between AWS and NBI, aside from adaptive refinement, is the fact that equality constraints in AWS are imposed radially from the pseudo-Nadir point rather than normal to the utopia plane defined by the anchor points as is done in NBI. Second, equality constraints are generally difficult to be satisfied. Optimizers are often good at finding a (local) optimal solution if it starts from within a feasible region, but optimization may fail to find a feasible solution if the initial design is far from the feasible region. When a line constraint (equality constraint) is specified in the objective space, we should make sure the initial design lies on or near the line. It is often difficult to find such an initial design, however, and we have to try many initial designs, which is computationally expensive.



Utopia plane for the NBI

Figure 13. Comparison of the AWS method and the NBI method.

A convex Pareto front example, non-convex Pareto front with dominated regions, and a three-bar problem are solved successfully by the multiobjective adaptive weighted sum method. The advantages over the usual weighted sum method – uniform distribution and the ability to determine non-convex Pareto front – are presented by the examples. Adaptivity and the ability to find the entire Pareto front are the merits of this method that the NBI method does not have. This method will be applied to problems with practical applications and complicated Pareto fronts as further work. This includes benchmarking AWS against NBI and other methods in terms of uniformity, completeness and computational effort for Pareto front generation. One typical case in which the usual weighted sum method fails to discover all parts of the Pareto front is when the anchor points are not unique (in the two-dimensional case, an anchor solution may be a line segment, and in the three-dimensional case, it can be a plane). The usual weighted sum method for the first stage will also be improved such that problems with non-unique anchor solutions can be treated. Furthermore, visualization challenges for patch representations in cases with more than three objectives (m>3) will be investigated.

References

[1] Stadler, W., "A Survey of Multicriteria Optimization, or the Vector Maximum Problem," *Journal of Optimization Theory and Applications*, Vol. 29, 1979, pp. 1-52.

[2] Stadler, W., "Applications of Multicriteria Optimization in Engineering and the Sciences (A Survey)," *Multiple Criteria Decision Making – Past Decade and Future Trends*, edited by M. Zeleny, JAI Press, Greenwich, Connecticut, 1984.

[3] Zadeh, L., "Optimality and Non-Scalar-Valued Performance Criteria," IEEE Transactions on Automatic Control, AC-8, 59, 1963.

[4] Koski, J., "Multicriteria truss optimization," *Multicriteria Optimization in Engineering and in the Sciences*, edited by Stadler, New York, Plenum Press, 1988.

[5] Marglin, S. *Public Investment Criteria*, MIT Press, Cambridge, Massachusetts, 1967.

[6] Lin, J., "Multiple objective problems: Pareto-optimal solutions by method of proper equality constraints," *IEEE Transactions on Automatic Control*, Vol. 21, 1976, pp. 641-650.

[7] Suppapitnarm, A., Seffen, K. A., Parks, G. T., and Clarkson, P. J., "A simulated annealing algorithm for multiobjective optimization," Engineering Optimization, Vol. 33, No. 1, 2000. pp. 59-85.

[8] Goldberg, D. E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.

[9] Fonseca, C., and Fleming, P., "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, Vol. 3, 1995, pp. 1-18.

[10] Das, I., and Dennis, J. E., "Normal-Boundary Intersection: A New Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems," *SIAM Journal on Optimization*, Vol. 8, No. 3, 1998, pp. 631-657.

[11] Messac, A., and Mattson, C. A., "Generating Well-Distributed Sets of Pareto Points for Engineering Design using Physical Programming," *Optimization and Engineering*, Kluwer Publishers, Vol. 3, 2002, pp. 431-450.

[12] Mattson, C. A., and Messac, A., "Concept Selection Using s-Pareto Frontiers," *AIAA Journal*. Vol. 41, 2003, pp. 1190-1204.

[13] Messac, A. and Mattson, C. A., "Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier," AIAA Journal, Vol. 42, 2004.

[14] Das, I., and Dennis, J. E., "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Structural Optimization*, Vol. 14, 1997, pp. 63-69.

[15] Koski, J., "Defectiveness of weighting method in multicriterion optimization of structures," *Communications in Applied Numerical Methods*, Vol. 1, 1985, pp. 333-337.

[16] Kim, I.Y., and de Weck, O., "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Structural and Multidisciplinary Optimization*, 2004 (in press).