

I. Y. Kim · O. L. de Weck

Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation

Received: 21 January 2005 / Revised manuscript received: 6 July 2005 / Published online: 20 December 2005
 © Springer-Verlag 2005

Abstract This paper presents an adaptive weighted sum (AWS) method for multiobjective optimization problems. The method extends the previously developed biobjective AWS method to problems with more than two objective functions. In the first phase, the usual weighted sum method is performed to approximate the Pareto surface 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 pseudonadir point and the expected Pareto optimal solutions on a piecewise planar hypersurface in the m -dimensional objective space. It is demonstrated that the method produces a well-distributed Pareto front mesh for effective visualization, and that it finds solutions in nonconvex regions. Two numerical examples and a simple structural optimization problem are solved as case studies.

Keywords NBI · AWS · Multiobjective optimization · Adaptive weighted sum · Pareto front

Nomenclature

- J** = Objective function vector
- x** = Design vector
- p** = Vector of fixed parameters
- g** = Inequality constraint vector
- h** = Equality constraint vector
- m = Number of objectives
- α_i = Weighting factor
- \bar{J}_i = Normalized objective function

- $\mathbf{J}^{\text{Utopia}}$ = Utopia point
- $\mathbf{J}^{\text{Nadir}}$ = Nadir point
- \mathbf{J}^{i*} = i th anchor point
- \mathbf{P}^j = Position vector of the j th expected solution on the hyperplane

1 Introduction

1.1 Multiobjective optimization and literature review

The goal of 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 life cycle cost for an airplane design. When there are multiple objective functions to be considered, the design problem becomes multiobjective. In this case, the usual design optimization method for a scalar objective function cannot be used.

Multiobjective optimization can be stated as follows:

$$\begin{aligned} \min \quad & \mathbf{J}(\mathbf{x}, \mathbf{p}) = [J_1 \ J_2 \ \dots \ J_m]^T \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq 0 \\ & \mathbf{h}(\mathbf{x}, \mathbf{p}) = 0 \\ & x_{i,\text{LB}} \leq x_i \leq x_{i,\text{UB}} \quad (i = 1, \dots, n) \end{aligned} \quad (1)$$

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, respectively; and $x_{i,\text{LB}}$ and $x_{i,\text{UB}}$ are the lower and upper bounds for the i th design variable, respectively. Stadler (1979, 1984) 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 scalar objective function by multiplying each objective function by a weighting factor and summing up all contributors:

$$J_{\text{weighted sum}} = w_1 J_1 + w_2 J_2 + \dots + w_m J_m \quad (2)$$

Presented as paper AIAA-2004-4322 at the 10th AIAA-ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, August 30–September 1, 2004

O. L. de Weck (✉)
 Department of Aeronautics and Astronautics,
 Engineering Systems Division, 33-410,
 Massachusetts Institute of Technology,
 Cambridge, MA 02139, USA
 e-mail: iykim@me.queensu.ca
 e-mail: deweck@mit.edu

I. Y. Kim
 Department of Mechanical Engineering,
 Queen's University, Kingston, Ontario, K7L 3N6, Canada

where w_i ($i = 1, \dots, m$) is a weighting factor for the i th objective function (which potentially can be divided by a scaling factor, i.e., $w_i = \alpha_i / sf_i$). If $\sum_{i=1}^m w_i = 1$ and $0 \leq w_i \leq 1$, then 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 systematically, and each different single objective optimization determines a different optimal solution. The solutions obtained approximate the Pareto front. Note that if there are nonunique anchor points, weights that have zero values may produce weak Pareto optimal solutions. Initial work on the weighted sum method can be found in Zadeh (1963). Koski (1988) applied the weighted sum method to structural optimization. Marglin (1967) developed the ϵ -constraint method, and Lin (1976) developed the equality constraint method. Heuristic methods are also used for multiobjective optimization: Suppaitnarm et al. (1999) applied simulated annealing to multiobjective optimization, and multiobjective optimization by genetic algorithms can be found in Goldberg (1989) and Fonseca and Fleming (1995) among others.

Das and Dennis (1998) developed the Normal Boundary Intersection (NBI) method, in which suboptimizations 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 m -dimensional problems with $m > 2$. The method can also determine Pareto optimal solutions in nonconvex regions, which the weighted sum method misses. The weak points of the NBI method are the following: (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 nondominance filtering must be used to filter out those solutions, and (3) in higher 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 (2002) and Mattson and Messac (2003) used physical programming for generating Pareto fronts for concept selection. They also developed the normal constraint method (Messac and Mattson 2004), which generates uniformly distributed solutions along the Pareto front without missing any Pareto front regions. Their method can be extended to m -dimensional problems. However, in all these methods, the way in which the Pareto front is computed is determined a priori, e.g., by predefined weights. The adaptive weighted sum (AWS) method on the other hand “learns” the shape of the Pareto front iteratively until some desired level of resolution is achieved.

1.2 Review of the adaptive weighted sum method for biobjective optimization

Figure 1 shows the fundamental concepts of the biobjective AWS method. The true Pareto front is represented by a solid line, and the solution points obtained by multiobjective optimization are denoted by small circles. The whole Pareto

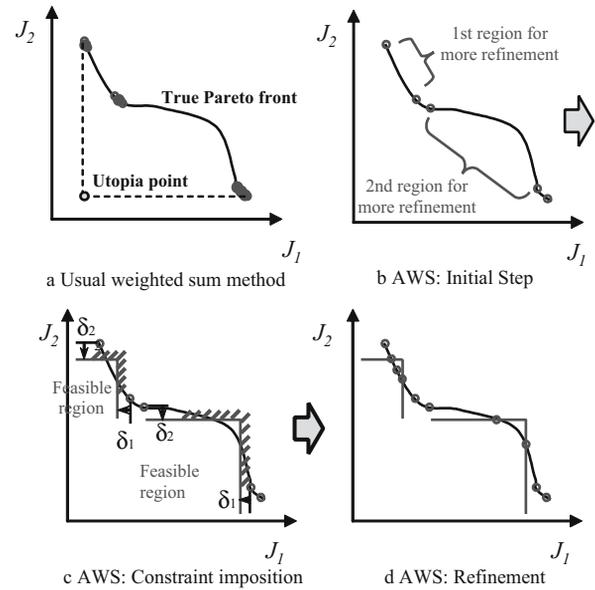


Fig. 1 The concept and procedure of the adaptive weighted sum (AWS) method (Kim and de Weck 2005)

front has two distinctively different regions: a relatively flat convex region and a concave region.

The problem may be solved by the typical weighted sum method, whose suboptimization problem is stated as:

$$\begin{aligned} \min \quad & \alpha \frac{J_1(\mathbf{x})}{sf_{1,0}(\mathbf{x})} + (1 - \alpha) \frac{J_2(\mathbf{x})}{sf_{2,0}(\mathbf{x})} \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \alpha \in [0, 1], \end{aligned} \quad (3)$$

where J_1 and J_2 are two objective functions to be mutually minimized, $sf_{1,0}$ and $sf_{2,0}$ are normalization factors for J_1 and J_2 , respectively, and α is the weighting factor which reveals the relative importance between J_1 and J_2 . By changing the weighting factor systematically (usually by a predetermined step size), we solve several suboptimization problems obtaining optimal solutions in the objective space. The optimum solution points then represent the Pareto front.

As shown in Fig. 1a, when the typical weighted sum method is used, most solutions concentrate near the anchor points and the inflection points, and no solutions are obtained in the concave region. The figure illustrates the two typical drawbacks of the weighted sum method, which were already discussed in a number of studies by Messac and Mattson (2002), Das and Dennis (1997), and Koski (1985):

1. Generally, the solutions are not uniformly distributed.
2. The weighted sum method cannot find solutions that lie in the nonconvex regions of the Pareto front; and, increasing the number of steps of the weighting factor does not resolve this problem.

Despite its intuitiveness about the relative importance between objective functions, the usage of the weighted sum

method is restricted due to these two problems. Figure 1b–d illustrates the fundamental concepts and overall procedure of the AWS method recently developed by Kim and de Weck (2005) to address these two drawbacks. The method initially determines the rough Pareto front (two anchor points in the case of a completely concave Pareto front) using the usual weighted sum method with a large step size. Based on the distances between neighboring solutions on the front in the objective space, regions for further refinement are selected. Suboptimizations are then conducted only in the selected regions by imposing additional inequality constraints in the objective space. Each region has two additional constraints that are parallel to each of the objective function axes. The constraints are constructed such that their distances from the solutions are δ_1 and δ_2 in the inward direction of J_1 and J_2 (see Fig. 1c). The suboptimization is stated as:

$$\begin{aligned} \min \quad & \alpha \frac{J_1(x)}{J_{1,0}(x)} + (1 - \alpha) \frac{J_2(x)}{J_{2,0}(x)} \\ \text{s.t.} \quad & J_1(x) \leq P_1^x - \delta_1 \\ & J_2(x) \leq P_2^y - \delta_2 \\ & \mathbf{h}(x) = 0 \\ & \mathbf{g}(x) \leq 0 \\ & \alpha \in [0, 1] \end{aligned} \quad (4)$$

where δ_1 and δ_2 are the offset distances selected by the user, P_i^x and P_i^y are the x and y positions of the i th endpoint ($i=1,2$) in each region selected for refinement in Fig. 1b, and $J_{1,0}$ and $J_{2,0}$ are scaling factors.

The suboptimization determines a new solution set in each region as shown in Fig. 1d. Again, regions for further refinement are selected by computing the distances between two adjacent solutions. The procedure is repeated until a termination criterion is met. The maximum segment length along the entire Pareto front is one measure for the convergence. Each line segment that is composed of two neighboring Pareto optimal solution points is termed a Pareto front patch. For example, the Pareto front is represented by four Pareto front patches in Fig. 1b, and after a refinement, the Pareto front is described by nine Pareto front patches. A Pareto front patch in the three-dimensional case is a surface whose corner points are Pareto optimal solutions. As in the case of a mesh in the Finite Element Method (FEM), the Pareto front patch in the three-dimensional problem may be quadrilateral (four Pareto optimal solutions) or triangular (three Pareto optimal solutions). The AWS method successfully solves multiobjective optimization problems: The AWS method produces well-distributed solutions, finds Pareto optimal solutions in nonconvex regions, and neglects non-Pareto optimal solutions. This method is different from other multiobjective optimization methods in that the AWS method does not explore the Pareto front in a predetermined fashion, but determines the Pareto front adaptively. Hence, the user does not need to select the final resolution of the Pareto front a priori without knowing its nature.

1.3 Adaptive weighted sum method for multiobjective optimization

The AWS method introduced in the previous section was only applicable to biobjective optimization problems. Therefore, we will refer to the previous technique as the “biobjective AWS method” to differentiate it from the generalized multiobjective AWS method presented here. In this paper, the biobjective AWS method is extended to multiobjective optimization problems with more than two objective functions. Unfortunately, the additional inequality constraints that are used in the biobjective AWS method are not suitable for higher dimensional multiobjective optimization. The reason is that the Pareto front patches 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 AWS method, additional equality constraints that connect the pseudonadir point and the expected locations of Pareto optimal solutions on the piecewise linearized hyperplane in the objective space are introduced. Suboptimizations 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 front. The goal of adaptive refinement is to have not only uniformly distributed solutions but also solutions that form a well-shaped mesh layout for effective visualization.

2 Multiobjective adaptive weighted sum method: fundamental concepts

The philosophy of the AWS method is to adaptively refine the Pareto front. In the first stage, the method determines a rough shape of the Pareto front. By estimating the size of each Pareto front patch, the regions for further refinement in the objective space are determined. In the subsequent stage, only these regions are specified as feasible domains for suboptimization by assigning additional constraints. In the biobjective AWS method, a Pareto front patch is a line segment that connects

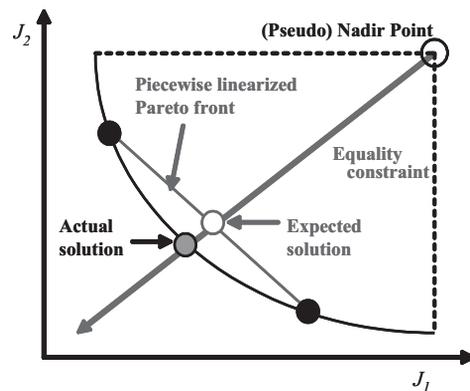


Fig. 2 AWS method for multidimensional problems (two-dimensional representation)

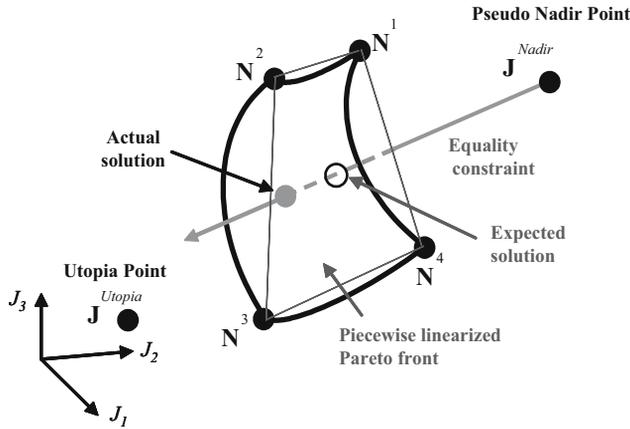


Fig. 3 AWS method for multidimensional problems (three-dimensional representation)

two neighboring solutions, and the feasible domain for further exploration is determined by specifying two inequality constraints.

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 endpoints, because the Pareto front is a two-dimensional curve, and there are always only two endpoints for each Pareto front segment.

However, in higher dimensional cases, the Pareto front becomes a surface (for three objective functions) or a hyper-surface (for more than three objective functions); it is difficult to impose constraints such that suboptimizations are performed only in a Pareto front patch selected for further refinement. 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. In this case, it is even more difficult to impose inequality constraints for suboptimization of further refinement. Indeed, the problems encountered then resemble adaptive remeshing in the FEM, but in higher dimensions. While there has been extensive research conducted in this field for decades, it is

also important to note that FEM remeshing techniques can be applied only to three-dimensional problems.

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 using inequality constraints. Adding equality constraints increases the likelihood of entrapment in local minima, but facilitates the adaptive procedure by simplifying mesh 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 convex regions of the Pareto front is determined by using the usual weighted sum method. Any nonconvex region of the Pareto front is not found in this stage of the usual weighted sum method, but they are determined in subsequent optimizations. Pareto front patches are then identified, and patches for further refinement are selected on the basis of the patch size. Additional equality constraints are introduced such that suboptimization is conducted only in the selected patches. Figure 2 shows the concept of the multiobjective AWS method with equality constraints for multiobjective optimization.

In the biobjective AWS method, feasible regions for further refinement are defined by two inequality constraints (Fig. 1); however, in the multiobjective AWS method, one or several equality constraints that connect the pseudonadir point and expected solutions on the piecewise linearized Pareto front are specified. 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 in the objective space, allows us to have great control over the position of new Pareto optimal solutions obtained, and this simplifies adaptive refinement. As shown in Fig. 3, 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. The position vector of the vertex of the Pareto front patch is denoted by N^i . It is important to note that all vertices N^i are Pareto optimal (nondominated) and are found in previous iterations. When solutions are obtained in all patches that are selected for further refinement, a new set of Pareto front patches is identified, and a patch-size evaluation is performed to determine where

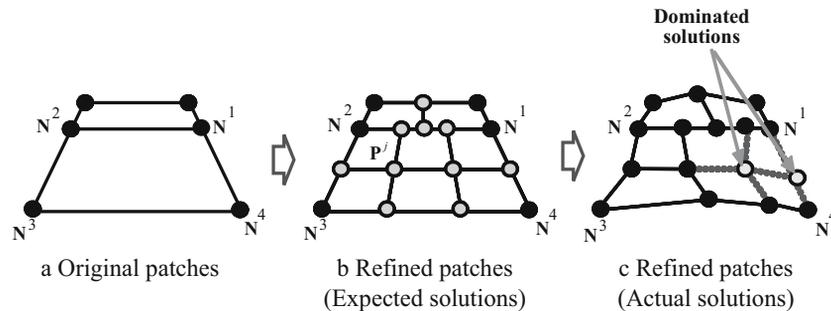


Fig. 4 Adaptive refinement procedure

sum method is used. The Euclidean 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. Values between 0.01 and 0.05 in the normalized objective space (9) are recommended for ϵ .

3.4 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 add complexity in programming, 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 nondominated 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.

3.5 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. Because the lower patch is larger, it is refined more than the upper one in the figure.

The size of a patch is identified by two dimensions for quadrilateral patches of the three-dimensional problem: the average length of two line edges that face each other. Each patch then has two averaged lengths, e.g., the averaged length of $\overline{N^1N^2}$ and $\overline{N^3N^4}$ the average length of $\overline{N^2N^3}$ and $\overline{N^1N^4}$ in Fig. 4. Depending on the orientation of the vector sum of the two facing edges, each corresponding averaged length is classified into the horizontal-type length (parallel to the plane of J_1 and J_2) or the vertical-type length (parallel to the J_3 axis). As an example, the averaged length of line $\overline{N^1N^2}$ and $\overline{N^3N^4}$ in Fig. 4 is regarded as the horizontal-type length. The length of each patch is compared to the mean of the lengths of all patches, and the refinement level is determined based on the relative length in each direction. In each mesh, the locations of expected solutions are determined by interpolation, and suboptimizations are conducted along the lines that connect the pseudonadir point (6) and the expected solutions. The actual solutions may be different from the expected solutions, and there can be dominated solutions, which must be deleted by a Pareto filter.

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

$$\mathbf{P}^j = \beta_1 \mathbf{N}^1 + \beta_2 \mathbf{N}^2 + \beta_3 \mathbf{N}^3 + \beta_4 \mathbf{N}^4, \quad \beta_i \in [0, 1], \quad (13)$$

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

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

$$\bar{\mathbf{P}}_i^j = \frac{P_i^j - J_i^{\text{Utopia}}}{J_i^{\text{Nadir}} - J_i^{\text{Utopia}}} \quad (14)$$

where P_i^j is the coordinate of the j th expected solution on the piecewise linearized (hyper-)plane.

3.6 Step 6

Impose an additional equality constraint for each expected solution and conduct a suboptimization with the weighted sum method. For the j th normalized expected solution, $\bar{\mathbf{P}}^j$, the suboptimization problem is defined as

$$\begin{aligned} & \text{minimize} \quad \mathbf{w} \cdot \bar{\mathbf{J}}(\mathbf{x}) \\ & \text{subject to} \quad \frac{(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{\text{Nadir}}) \cdot (\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{\text{Nadir}})}{|\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{\text{Nadir}}| |\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{\text{Nadir}}|} = 1 \\ & \quad \bar{\mathbf{h}}(\mathbf{x}) = 0 \\ & \quad \bar{\mathbf{g}}(\mathbf{x}) \leq 0 \end{aligned} \quad (15)$$

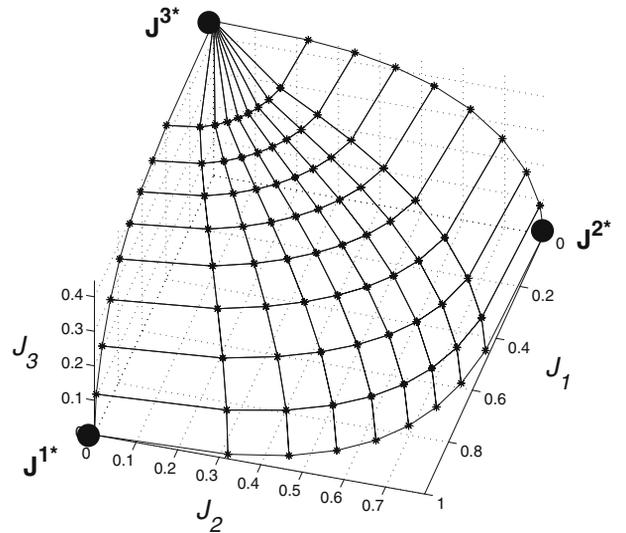


Fig. 6 Pareto front obtained by the usual weighted sum method for example 1

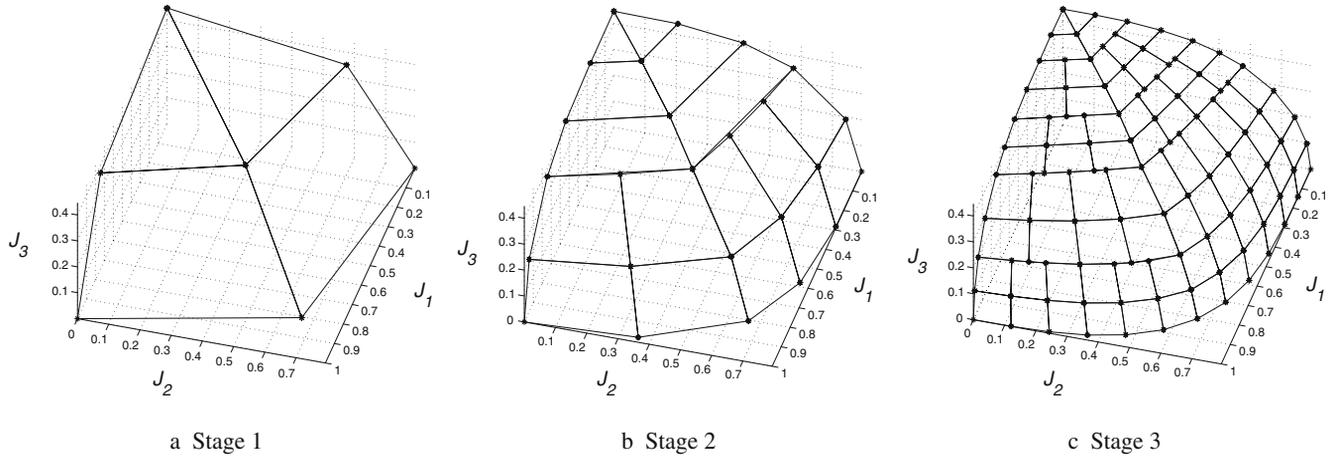


Fig. 7 Pareto front obtained by the multiobjective AWS method for example 1

where $\mathbf{w} = -(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{\text{Nadir}})$ 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 $\bar{\mathbf{J}}^{\text{Nadir}}$ is a vector whose components are one, i.e., $\bar{\mathbf{J}}^{\text{Nadir}} = (1, 1, \dots, 1)$.

The equality constraint containing the dot product in (15) forces the two vectors $\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{\text{Nadir}}$ and $\bar{\mathbf{J}}(\mathbf{x}) - \bar{\mathbf{J}}^{\text{Nadir}}$ to be collinear in the objective space. This constraint therefore ensures that the solution is obtained only along the line $\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{\text{Nadir}}$, which connects the expected solution on the piecewise linearized plane and the pseudonadir point. The objective function $-(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{\text{Nadir}}) \cdot \bar{\mathbf{J}}(\mathbf{x})$ is a scalar function to be minimized, determining the solution that is nearest to the utopia point in the direction of $-(\bar{\mathbf{P}}^j - \bar{\mathbf{J}}^{\text{Nadir}})$.

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

The subproblem introduces an equality constraint (line constraint in the objective space), which is often difficult to satisfy. The location of the expected Pareto optimal solution in the objective space can be precisely computed as illustrated in Fig. 5, but what we actually need for optimization is its corresponding design vector in the design space, which is not known. A method to guess the initial design vector for each expected Pareto solution is to use the same interpolation functions (13) using the four corner points in the design space, which have been already determined in the previous stage.

3.7 Step 7

Perform Pareto filtering In the biobjective AWS method, non-Pareto optimal solutions are automatically rejected, hence the filtering is not needed. In the multiobjective AWS 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.

The Pareto filter compares each solution against all other Pareto optimal solutions in the same stage with respect to all objective function values. The new solution compared becomes a nondominated solution if and only if no other solution performs better in terms of all objective functions (no other solution has a smaller value in all objective functions in the minimization problem).

3.8 Step 8

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.

4 Numerical examples

Three numerical examples are presented in this section to demonstrate the performance of the multiobjective AWS method. All examples are three-dimensional problems. Sequential Quadratic Programming (SQP) in MATLAB is used as the optimization algorithm.

Table 1 Number of Pareto front patches, nondominated solutions, and dominated solutions in each stage of example 1

Stage	1	2	3
Number of Pareto front patches	4	16	72
Number of nondominated solutions	7	25	102
Number of dominated solutions	0	0	0

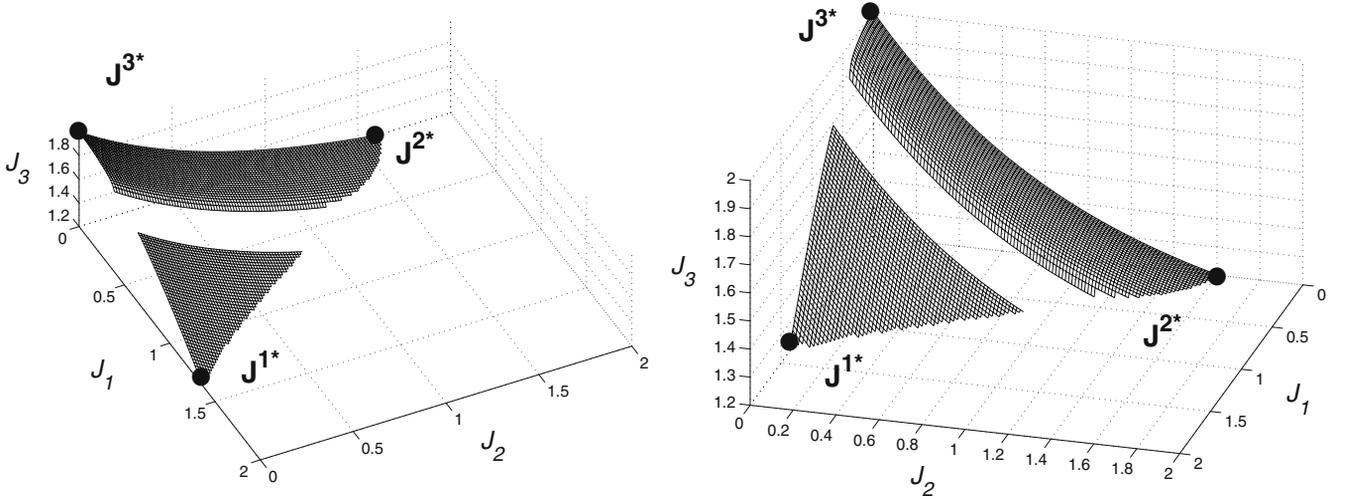


Fig. 8 Pareto front profile for example 2

4.1 Example 1: convex Pareto front

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

$$\text{maximize } [J_1 \ J_2 \ J_3]^T$$

$$\text{subject to } x_1^4 + 2x_2^3 + 5x_3^2 \leq 1$$

$$J_1 = x_1$$

$$J_2 = x_2$$

$$J_3 = x_3$$

$$x_i \geq 0 \quad (i = 1, 2, 3). \quad (16)$$

The Pareto front of this problem is convex, but the curvatures are different in the three axes. Figure 6 shows the Pareto front obtained by the usual weighted sum method. The step sizes for the two weighting factors in (12), $\Delta\alpha_1$ and $\Delta\alpha_2$, are one ninth. 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 patches on the top along the J^3 axis are slender, and the patches near to the two anchor points J^{1*} and J^{2*} 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 AWS 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 obtained by the usual weighted sum method in Fig. 6, the mesh shape and size are quite

uniform. Note that some patches do not share corner points (Pareto solutions) with their neighboring patches.

Because the Pareto front in this example is convex, all solutions obtained by the adaptive multiobjective optimization are nondominated solutions. Table 1 shows the numbers of Pareto front patches, nondominated solutions, and dominated solutions in each stage.

Note that the AWS method produces not only uniformly distributed solutions but also solutions that form a well-shaped mesh layout. Multiobjective optimization is conducted 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. However, in higher dimensions, 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,

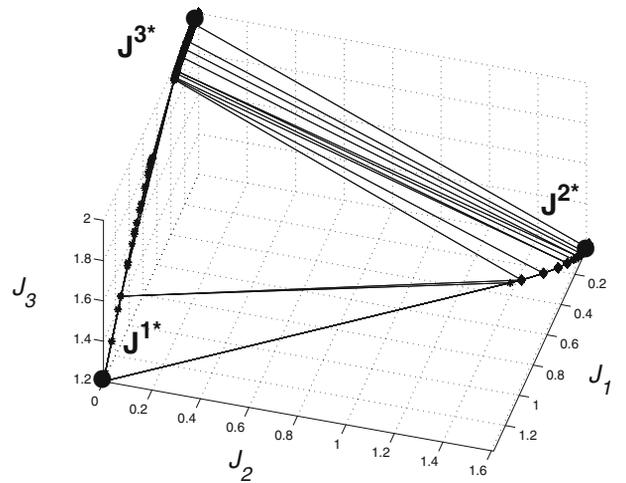


Fig. 9 Pareto front obtained by the usual weighted sum method for example 2

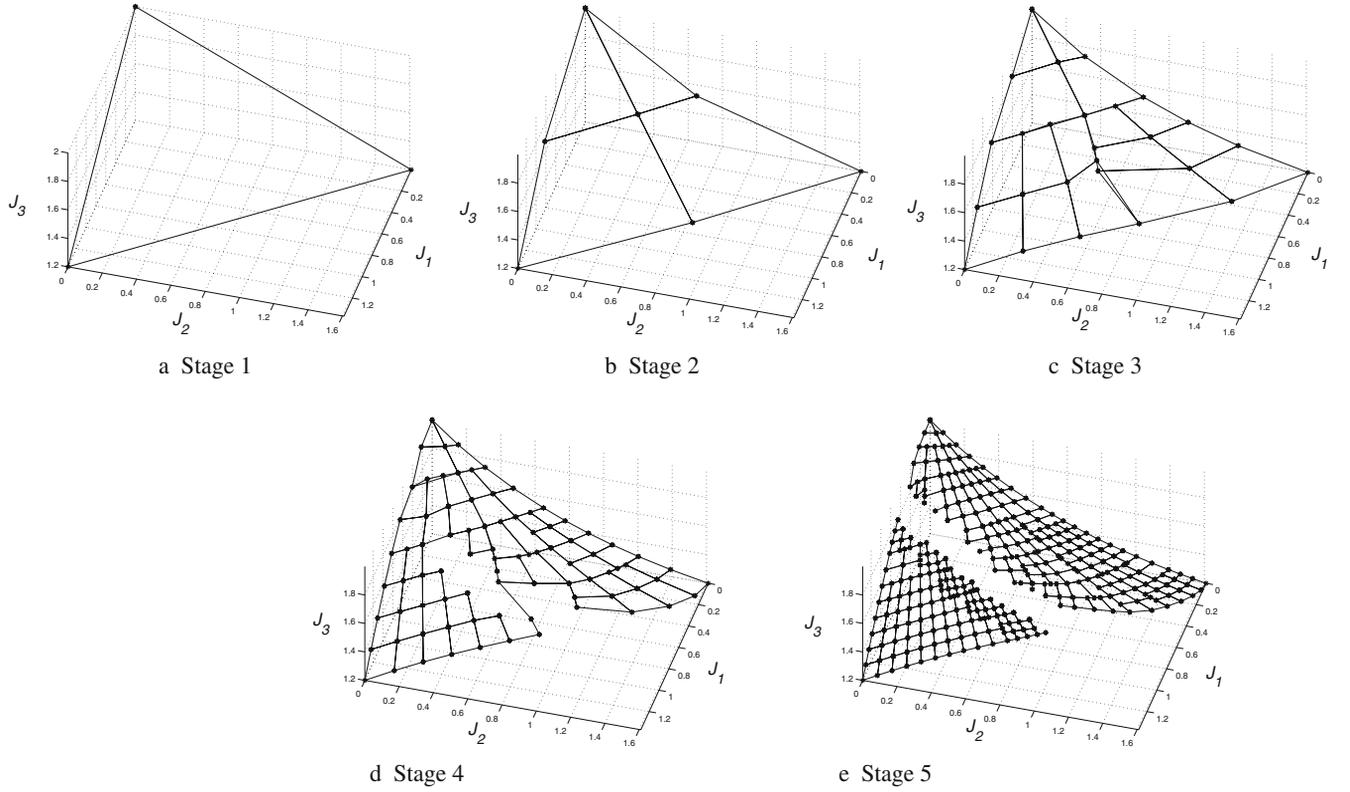


Fig. 10 Pareto front obtained by the multiobjective AWS method for example 2

and it is very easy to visualize the Pareto front obtained—at least in three objective dimensions at-a-time.

4.2 Example 2: nonconvex 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 nonconvex regions and is disconnected due to dominated solutions. The problem statement is

$$\text{maximize } [J_1 \ J_2 \ J_3]^T$$

$$\text{subject to } -\cos x_1 - e^{-x_2} + x_3 + 0.5e^{-200(x_1-0.5)^2} \leq 0$$

$$J_1 = x_1$$

$$J_2 = x_2$$

$$J_3 = x_3$$

$$0 \leq x_1 \leq \pi$$

$$x_2 \geq 0$$

$$x_3 \geq 1.2$$

(17)

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 edge curves: The curve between J^1* and J^3* is convex with a gap due to a dominated solution region, but the other two curves are not convex. The overall surface of the Pareto front is nonconvex with a middle region that looks like a valley.

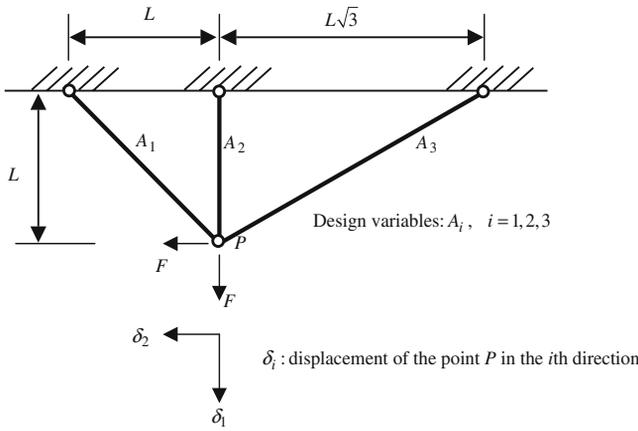
The usual weighted sum method is used to determine the Pareto front with a 20×20 mesh. As appears in Fig. 9, solutions are obtained only on the convex curve between J^1* and J^3* and on the short convex curve segment near J^2* . Dominated solutions are not obtained, but most of the Pareto front, which is nonconvex, is not revealed by this method.

The multiobjective AWS 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. 10a). In the second stage, the overall shape of the nonconvex Pareto front is found; a result which cannot be obtained no mat-

Table 2 Number of Pareto front patches, nondominated solutions, and dominated solutions in each stage of example 2

Stage	1	2	3	4	5
Number of Pareto front patches	1	4	16	50	270
Number of nondominated solutions	3	7	26	83	214
Number of dominated solutions	0	0	0	6	44

Figure 8 shows the Pareto front of this problem from two different viewpoints, which was generated with MATLAB



$F = 20 \text{ kN}$, $L = 100 \text{ cm}$, $E = 200 \text{ GPa}$
 $\sigma_{\text{upper limit}} = 200 \text{ MPa}$, $\sigma_{\text{lower limit}} = -200 \text{ MPa}$
 $A_{\text{upper limit}} = 2 \text{ cm}^2$, $A_{\text{lower limit}} = 0.1 \text{ cm}^2$

Fig. 11 Three-bar problem (Koski 1985)

ter how many solutions are found by the usual weighted sum method. Until the third stage, all solutions obtained are nondominated solutions and are used to construct the Pareto front. From the fourth stage on, however, dominated solutions (non-Pareto optimal solutions) are obtained by the multiobjective AWS method, because the equality constraint used cannot differentiate dominated solutions from nondominated ones. A Pareto filtering step is conducted in each stage (Step 7). 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. It can be seen that the Pareto front in stage 5 in Fig. 10e is the same as the one obtained by a full factorial analysis in Fig. 8. The number of dominated solutions, which were detected by a Pareto filter and removed from the Pareto front, is six in stage 4 and 44 in stage 5 as shown in Table 2.

4.3 Example 3: three-bar problem

Finally, the multiobjective AWS method is applied to the three-bar problem, see Koski (1985). The geometry and material properties are presented in Fig. 11. The upper end of

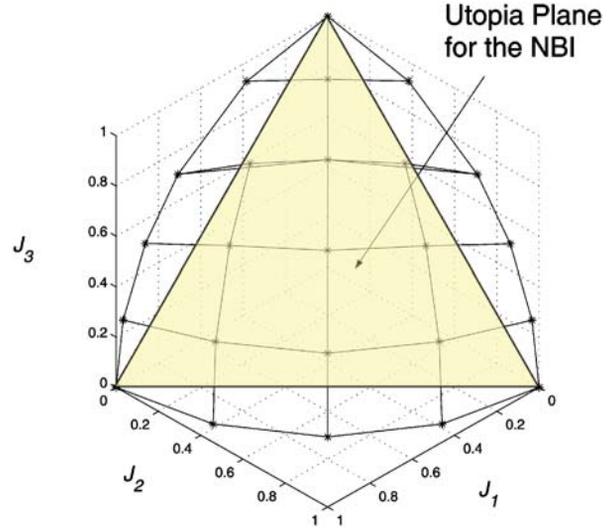


Fig. 13 Comparison of the AWS method and the NBI method

each bar is fixed, and the horizontal and vertical loads F 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 as follows:

$$\begin{aligned} &\text{minimize } [\text{Volume}(A) \ \sigma_1(A) \ \sigma_3(A)]^T \\ &\text{subject to } -200 \text{ MPa} \leq \sigma_i(A) \leq 200 \text{ MPa} \ (i = 1, 2, 3) \\ &\quad 0.1 \text{ cm}^2 \leq A_i \leq 2 \text{ cm}^2 \ (i = 1, 2, 3) \end{aligned} \tag{18}$$

where the design variable A_i is the cross-sectional area of the i th truss. The Pareto front of this problem is nonconvex, and some part of the objective space is dominated. Figure 12 shows the results obtained using the multiobjective AWS method. For more effective 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 pseudonadir point is the origin where three reference planes in the figure meet. In the first stage, the approximate Pareto front is represented by six patches. The multiobjective AWS method is then applied, determining more refined Pareto front meshes until convergence is

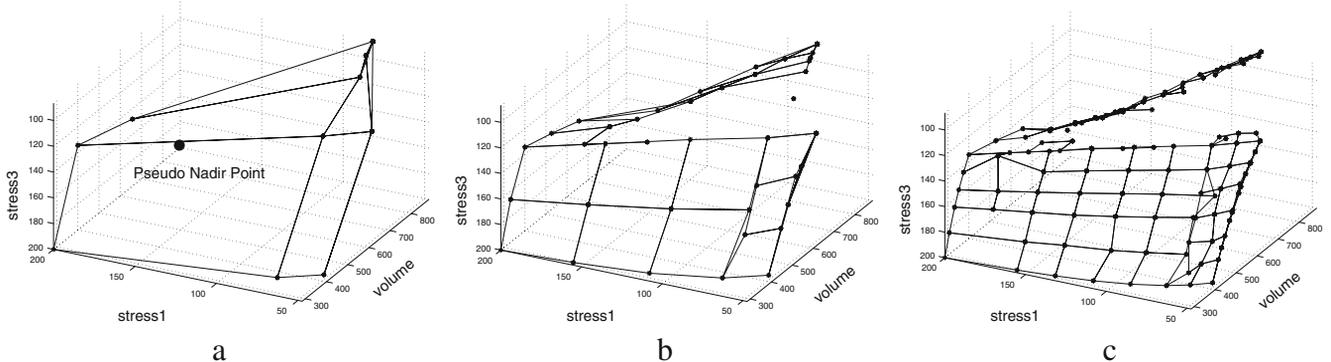


Fig. 12 Pareto front obtained by the multiobjective AWS method for example 3

reached in the third stage. There is an open region on the top surface of the box-like Pareto front; although solutions are determined in the region by the AWS method, they are discarded by the Pareto filter. It is clear from the figure that stress 1 and stress 3 exhibit a trade-off with respect to volume, i.e., the volume may decrease only at the expense of an increase in stress 1 or stress 3. The trade-off between stress 1 and stress 3 can be observed in the top-right portion of the Pareto front in stage 3 of the figure.

5 Discussion

The multiobjective AWS method effectively solves multiobjective optimization problems with more than two objective functions. In the biobjective AWS method, which is applicable only to optimizations with two objective functions, inequality constraints are used to specify regions for further refinement. In the multiobjective AWS method, on the other hand, equality constraints are used, and the method is scalable to m -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 three important issues in using the multiobjective AWS method. First, adaptive refinement is conducted only within the footprint of the Pareto front that is determined by the usual weighted sum method in the first stage. Pareto front regions that are located within this first-stage Pareto front approximation will be found in subsequent stages, although those regions are not detected by the usual weighted sum method in the first stage. However, Pareto front regions that stay out of the footprint of the first-stage Pareto front would not be discovered in the following adaptive refinement stages. In general, the usual weighted sum method finds most parts of the convex Pareto front reliably and quickly. This is, in addition to its adaptivity, a distinctive advantage of the AWS 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 AWS method (based on the first-stage usual weighted sum method) and the utopia plane for the NBI method. The utopia plane is defined as a hyperplane on which all anchor points lie. 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, in addition to adaptive refinement, is the fact that equality constraints in AWS are imposed radially from the pseudonadir point rather than normally to the utopia plane defined by the anchor points as is done in NBI.

Second, equality constraints are generally difficult to be satisfied as previously discussed in Step 6 in “Multiobjective adaptive weighted sum method: procedure.” Optimization tools are often good at finding a (local) optimal solution if they start 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 in the design space lies on or near the feasible domain that satisfies the equality constraint. It is often difficult to find such an initial design. One may have to try many initial designs near to the guessed one (which is explained in Step 6), and this can be computationally expensive.

Third, if the multiobjective optimization problem has a very irregularly shaped Pareto front, it would be difficult to represent the Pareto front using only quadrilateral patches. As in the case of adaptive meshing in the FEM, it would be more effective to use a combination of triangular and quadrilateral meshes, or triangular meshes alone. For this implementation, more study will be needed, e.g., where to use a triangular mesh and where to use a quadrilateral mesh, how to determine the relative size of the different types of meshes for further refinement, how to refine a patch with triangular and quadrilateral meshes, etc.

A convex Pareto front example, a nonconvex Pareto front with a dominated region, and a three-bar problem were solved successfully by the multiobjective AWS method. The advantages over the usual weighted sum method—uniform distribution and the ability to determine a nonconvex 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 complex 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. A 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 nonunique anchor solutions can be treated. Furthermore, visualization challenges for patch representations in cases with more than three objectives ($m > 3$) will be investigated.

References

- Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Struct Optim* 14:63–69
- Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM J Optim* 8:631–657
- Fonseca C, Fleming P (1995) An overview of evolutionary algorithms in multiobjective optimization. *Evol Comput* 3:1–18
- Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading
- Kim IY, de Weck OL (2005) Adaptive weighted sum method for bi-objective optimization: Pareto front generation. *Struct Multidiscipl Optim* 29:149–158
- Koski J (1985) Defectiveness of weighting method in multicriterion optimization of structures. *Commun Appl Numer Methods* 1:333–337

- Koski J (1988) Multicriteria truss optimization. In: Stadler W (ed) *Multicriteria optimization in engineering and in the sciences*. Plenum, New York
- Lin J (1976) Multiple objective problems: Pareto-optimal solutions by method of proper equality constraints. *IEEE Trans Automat Contr* 21:641–650
- Marglin S (1967) *Public investment criteria*. MIT, Cambridge, MA
- Mattson CA, Messac A (2003) Concept selection using s-Pareto frontiers. *AIAA J* 41:1190–1204
- Messac A, Mattson CA (2002) Generating well-distributed sets of Pareto points for engineering design using physical programming. *Optim Eng* 3:431–450
- Messac A, Mattson CA (2004) Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA J* 42:2101–2111
- Stadler W (1979) A survey of multicriteria optimization, or the vector maximum problem. *Jota* 29:1–52
- Stadler W (1984) Applications of multicriteria optimization in engineering and the sciences (A Survey). In: Zeleny M (ed) *Multiple criteria decision making—past decade and future trends*. JAI, Greenwich
- Suppakitnarm A et al (1999) Design by multiobjective optimization using simulated annealing. *International conference on engineering design ICED 99*, Munich, Germany
- Zadeh L (1963) Optimality and non-scalar-valued performance criteria. *IEEE Trans Automat Contr* 8:59–60