45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference
19 - 22 April 2004, Palm Springs, California

AIAA 2004-1680

Adaptive Weighted Sum Method for Bi-objective Optimization

Olivier de Weck* and Il Yong Kim†
Massachusetts Institute of Technology, Cambridge, MA, 02139

This paper presents a new method that effectively determines a Pareto front for bi-objective optimization with potential application to multiple objectives. A traditional method for multiobjective optimization is the weighted sum method, which seeks Pareto optimal solutions one by one by systematically changing the weights among the objective functions. Previous research has shown that this method often produces poorly distributed solutions of a Pareto front, and that it does not find Pareto optimal solutions in non-convex regions. The proposed adaptive weighted sum method focuses on unexplored regions by changing the weights adaptively rather than by using a priori weight selections and by specifying additional inequality constraints. It is demonstrated that the adaptive weighted sum method produces well-distributed solutions, finds Pareto optimal solutions in non-convex regions, and neglects non-Pareto optimal solutions. This last point can be a potential liability of Normal Boundary Intersection, an otherwise successful multiobjective method, which is mainly caused by its reliance on equality constraints. The promise of the algorithm is demonstrated with two numerical examples and a simple structural optimization problem.

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
α	=	weighting factor
$\bar{J}_i$	=	normalized objective function
$\mathbf{J}^U$	=	utopia point
$\mathbf{J}^N$	=	nadir point
n_i	=	number of further refinements for the i th segment
l_i	=	length of the i th segment
l_{avg}	=	average length of all the segments at a stage
C	=	multiplier
$\mathbf{P}_1$, $\mathbf{P}_2$	=	end point vectors of a segment
δ_J	=	offset distance along a piecewise linearized Pareto front
δ_i	=	offset distance along the i th objective function
Δx_1, Δx_2	=	grid size

---

*Assistant Professor, Department of Aeronautics and Astronautics and Engineering Systems Division, Room 33-410, MIT, AIAA member
† Postdoctoral Associate, Department of Aeronautics and Astronautics, Room 33-409, MIT

I. Introduction

In engineering, a designer often deals with more than one objective function or design criterion, and sometimes these multiple objective functions conflict with each other. For example, one may want to maximize the performance of a system while minimizing its cost. Such kind of design problems are the subject of multiobjective optimization and can generally be formulated as a MONLP (Multiple Objective Nonlinear Program) of the form:

$$
\begin{aligned}
\min \quad & \mathbf{J}(\mathbf{x},\mathbf{p}) \\
\text{s.t.} \quad & \mathbf{g}(\mathbf{x},\mathbf{p}) \le 0 \\
& \mathbf{h}(\mathbf{x},\mathbf{p}) = 0 \\
& x_{i,LB} \le x_i \le x_{i,UB} \quad (i = 1,\ldots,n) \\
& \mathbf{J} = \begin{bmatrix} J_1(\mathbf{x}) & \cdots & J_z(\mathbf{x}) \end{bmatrix}^T \\
& \mathbf{x} = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \end{bmatrix}^T \\
& \mathbf{g} = \begin{bmatrix} g_1(\mathbf{x}) \cdots g_{m_1}(\mathbf{x}) \end{bmatrix}^T \\
& \mathbf{h} = \begin{bmatrix} h_1(\mathbf{x}) \cdots h_{m_2}(\mathbf{x}) \end{bmatrix}^T
\end{aligned}
\tag{1}
$$

where $\mathbf{J}$ is an objective function vector, $\mathbf{x}$ is a design vector, $\mathbf{p}$ is a vector of fixed parameters, $\mathbf{g}$ is an inequality constraint vector, and $\mathbf{h}$ is an equality constraint vector. In this case there are z objectives, n design variables, m_1 inequality constraints and m_2 equality constraints. Additionally, the design variables may be bounded by side constraints assuming that $x_i \in \mathbb{R}$.

After Pareto[1] introduced the concept of non-inferior solutions now called Pareto-optimal solutions, Stadler[2,3] began to apply the notion of Pareto optimality to the fields of engineering and science in the 1970s. The applications of multiobjective optimization in engineering design grew over the following decades. One of the most widely used methods for solving multiobjective optimization problems is to transform a multiple objective (vector) functions into a series of single objective (scalar) functions:

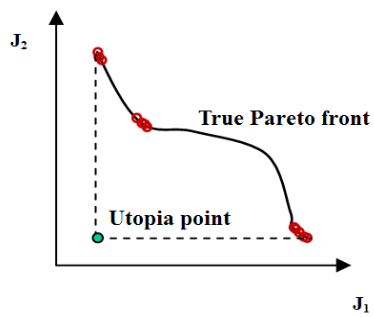
$$J_{\text{weighted sum}} = w_1 J_1 + w_2 J_2 + \cdots + w_z J_z \tag{2}$$

where w_i $(i = 1,\cdots,z)$ is a weight for the ith objective function. If $\sum w_i = 1$ and $0 \le w_i \le 1$, the weighted sum is said to be a convex combination of objectives. When an appropriate set of solutions is obtained by the single objective optimizations, the solutions can approximate the Pareto front in objective space. The weighted sum method is a traditional, popular method that parametrically changes the weights among objective functions to obtain the Pareto front. Initial work on the weighted sum method can be found in Zadeh[4] with many subsequent applications and citations. Koski[5], for example, studied the weighted sum method in the context of multicriteria truss optimization.

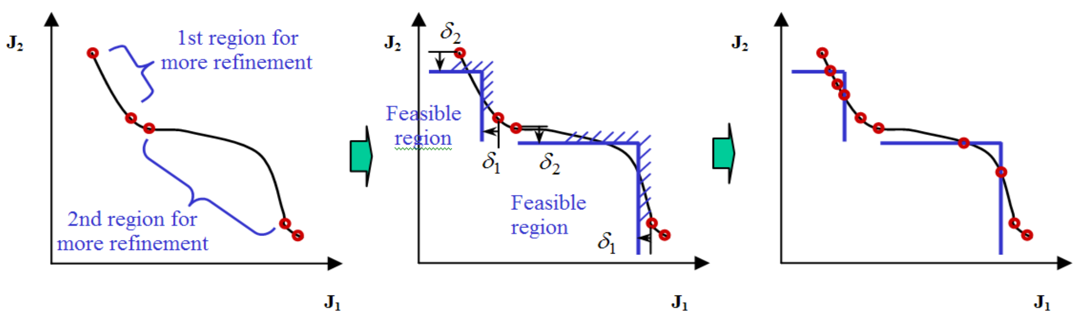
Marglin[6] developed the ε-constraint method, where one individual objective function is minimized with an upper level constraint imposed on the other objective functions[7]. Lin[8] developed the equality constraint method that minimizes objective functions one by one by simultaneously specifying equality constraints on the other objective functions. Heuristic methods are also used for multiobjective optimization; Suppapitnarm[9] applied simulated annealing to multiobjective optimization, and multiobjective optimization by Genetic Algorithms can be found in Goldberg[10], Fonseca and Fleming[11], and Tamaki et al.[12] among others. Messac and Mattson[13] used physical programming for generating a Pareto front, and they introduced the concept of s-Pareto fronts for concept selection[14]. Das and Dennis[15] proposed the NBI (Normal Boundary Intersection) method where a series of single objective optimizations is solved on normal lines to the Utopia line. The NBI method gives fairly uniform solutions and can treat problems with non-convex regions on the Pareto front. It achieves this by imposing equality constraints along equally spaced lines or hyperplanes in the multidimensional case.

As discussed in a number of studies by Messac and Mattson[13], Das and Dennis[16], and Koski[17], the traditional weighted sum approach has two main drawbacks. First, an even distribution of the weights among objective functions does not always result in an even distribution of solutions on the Pareto front. Indeed in real applications, solutions quite often appear only in some parts of the Pareto front, while no solutions are obtained in other parts. Second, the weighted sum approach cannot find solutions on non-convex parts of the Pareto front although they are non-dominated optimum solutions (Pareto optimal solutions). This is due to the fact that the weighted sum method is often implemented as a convex combination of objectives, where the sum of all weights is constant and negative weights are not allowed. Increasing the number of weights by reducing step size does not solve this problem. Eventually, this may result in selecting an inferior solution by missing important solutions on the concave regions.

Despite the drawbacks aforementioned, it is true that the weighted sum approach is extensively used because it is simple to understand and easy to implement. Also, the weight itself has a physical meaning, which reflects the relative importance among objective functions under consideration. We propose a new adaptive method, based on the weighted sum approach, for multiobjective optimization. In this approach, the weight is not predetermined, but it evolves according to the nature of the Pareto front of the problem. Starting from a large step size of the weight, a coarse representation of the solution is generated and regions where more refinement is needed are identified. The specific regions are then designated as a feasible region for suboptimization by imposing inequality constraints in the objective space. The typical weighted sum multiobjective optimization is performed in the regions. When all the regions of the Pareto front reach a pre-specified resolution, the algorithm terminates. The methodology is formulated and demonstrated for bi-objective optimization where there are two objective functions. The potential for extension to greater numbers of objectives is briefly discussed.



(a) The typical weighted sum method



(b) The procedure of the adaptive weighted sum method

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

II. Adaptive Weighted Sum Method: Fundamental Concepts

Figure 1 shows the concepts of the adaptive weighted sum method, compared with the typical weighted sum approach. The true Pareto front is represented by a solid line, and the solution points obtained by multiobjective optimization are denoted by small circles. In this example, the whole Pareto line is composed of two parts: a relatively flat convex region and a distinctly concave region. A typical way to solve the problem is to use the weighted sum method, which is stated as:

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

where J_1 and J_2 are two objective functions to be mutually minimized, $J_{1,0}$ and $J_{2,0}$ 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.

When the typical weighted sum method is used, as shown in Fig. 1 (a), most solutions concentrate near the anchor points and the inflection point , and no solutions are obtained in the concave region. The figure illustrates the two typical drawbacks of the weighted sum method:

(1) Generally, the solutions are not uniformly distributed.

(2) The weighted sum method cannot find solutions that lie in non-convex regions of the Pareto front. Increasing the number of steps of the weighting factor does not resolve this problem.

These are the main reasons that restrict the usage of the weighted sum method despite its simplicity and insight about the relative importance among objective functions. The ill-behaved nature of the method is frequently observed in realistic design optimization problems.

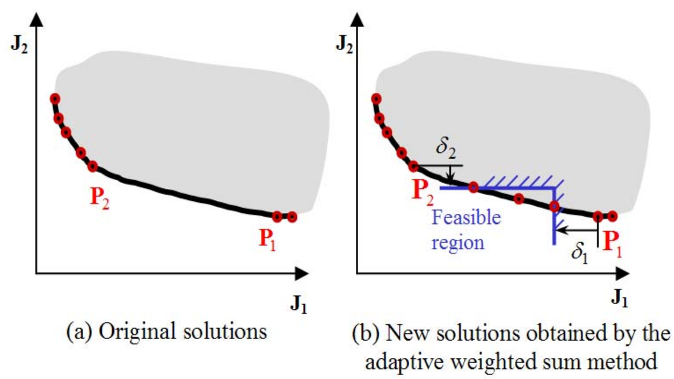


Figure 2: The adaptive weighted sum method for a convex Pareto front.

Figure 1 (b) illustrates the fundamental concepts and overall procedure of the proposed adaptive weighted sum method. It starts from a small number of divisions with a large step size of the weighting factor, α, using the traditional weighted sum method. By calculating the distances between neighboring solutions on the front in the objective space, regions for further refinement are identified. Only these regions then become the feasible regions for optimization 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. A suboptimization is solved in each of the regions, and a new solution set is identified. 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 among the entire Pareto front is one measure for the convergence. The detailed procedure is elaborated in the following section.

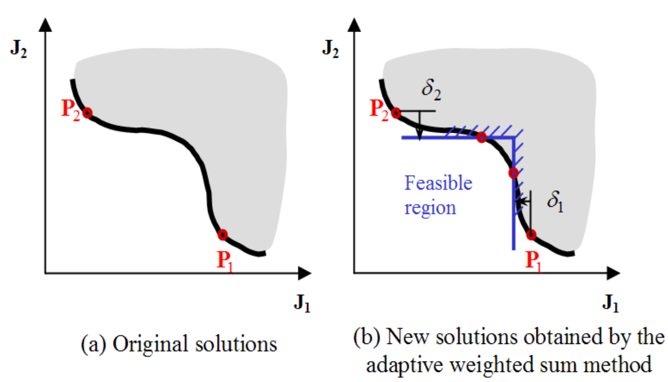


Figure 3: The adaptive weighted sum method for non-convex Pareto regions of non-dominated solutions.

The adaptive weighted sum method can effectively solve multiobjective optimization problems whose Pareto front has (i) convex regions with non-uniform curvature, (ii) non-convex regions of non-dominated solutions, and (iii) non-convex regions of dominated solutions. First, for a multiobjective optimization problem of non-uniform curvature Pareto front, most solutions obtained with the usual weighted sum method are concentrated in the region whose curvature is relatively high. Figure 2 (a) shows that very few solutions are obtained in the flat region when the usual weighted sum method is used. Because the segment length between $\mathbf{P_1}$ and $\mathbf{P_2}$ is larger than others, a feasible region for further refinement is established in the segment, in the adaptive weighted sum method. The optimization is then conducted only within this region, and more Pareto optimal solutions are obtained here. This makes the distribution of solutions more uniform, as shown in Fig. 2 (b).

In the second case of a non-convex region of non-dominated solutions, there exist Pareto optimal solutions in the region that the usual weighted sum approach cannot reach. In Fig. 3 (a), no solutions are obtained between $\mathbf{P}_1$ and $\mathbf{P_2}$ if the usual weighted sum method is used. On the other hand, the adaptive weighted sum method finds solutions because the optimization is conducted only in the non-convex region, as shown in Fig. 3 (b). The region is made by imposing inequality constraints that are offset from $\mathbf{P_1}$ and $\mathbf{P_2}$ by distances of δ_1 and δ_2 in the direction of J_1 and J_2, respectively. In this case, only two solutions are obtained at the points where the Pareto front and the inequality constraints intersect with each other.

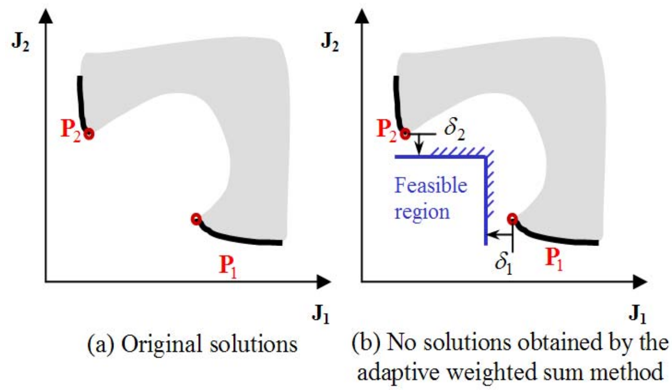


Figure 4: The adaptive weighted sum method for non-convex Pareto regions of dominated solutions.

In the third case of concave regions of dominated solutions, there are no Pareto optimal solutions in the region between $\mathbf{P}_1$ and $\mathbf{P}_2$, as shown in Fig. 4. No solution must be identified between $\mathbf{P}_1$ and $\mathbf{P}_2$ in this case. Indeed, the adaptive weighted sum method does not give optimum solutions because there is no feasible region within the imposed constraints, whereas the normal boundary intersection (NBI) method typically produces dominated solutions in this case. In summary, the adaptive weighted sum method produces evenly distributed solutions, finds Pareto optimal solutions in non-convex regions, and neglects non-Pareto optimal solutions in non-convex regions.

III. Adaptive Weighted Sum Method: Procedures

In this section, the detailed procedure of the adaptive weighted sum method is described. The description is valid for the bi-objective case.

[Step 1] Normalize the objective functions in the objective space. When $\mathbf{x}^{i*}$ is the optimal solution vector for the single objective optimization of J_i, the normalized objective function $\overline{J}_i$ is obtained as,

$$\overline{J}_i = \frac{J_i - J_i^U}{J_i^N - J_i^U}. \tag{4}$$

Here $\mathbf{J}^U$ is the utopia point and is defined as

$$\mathbf{J}^U = [J_1(\mathbf{x}^{1*}),\ J_2(\mathbf{x}^{2*})], \tag{5}$$

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

$$\mathbf{J}^N = [J_1^N,\ J_2^N] \tag{6}$$

where

$$J_i^N = \max[J_i(\mathbf{x}^{1*})\ \ J_i(\mathbf{x}^{2*})]. \tag{7}$$

[Step 2] Perform multiobjective optimization using the usual weighted sum approach with a small number of divisions, n_{initial}. We use $n_{\text{initial}} = 5 \sim 10$. The uniform step size of the weighting factor α is determined by the number of divisions:

$$\Delta\alpha = \frac{1}{n_{\text{initial}}} \tag{8}$$

By using a large step size of the weighting factor, $\Delta\alpha$, a small number of solutions is obtained.

[Step 3] Compute the lengths of the segments between all the neighboring solutions. 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.

[Step 4] Determine the number of further refinements in each of the regions. The longer the segment is, relative to the average length of all segments, the more it needs to be refined. The refinement is determined based on the relative length of the segment:

$$n_i = round\left(C\frac{l_i}{l_{avg}}\right) \quad \text{for the } i\text{th segment} \tag{9}$$

where n_i is the number of further refinements for the i th segment, l_i is the length of the i th segment, l_{avg} is the average length of all the segments, and C is a multiplier. The usual value of C is between 1 and 2. The function '$round$' rounds off to the nearest integer.

[Step 5] If n_i is less than or equal to one, no further refinement is conducted in the segment. For other segments whose number of further refinements is greater than one, go to the following step.

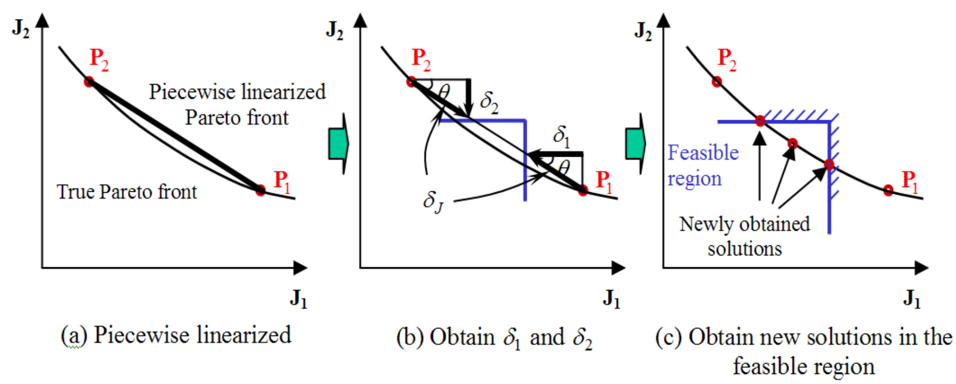


Figure 5: Determining the offset distances, δ_1 and δ_2, based on δ_J.

[Step 6] Determine the offset distances from the two end points of each segment. First, a piecewise linearized secant line is made by connecting the end points, $\mathbf{P}_1$ and $\mathbf{P}_2$ (Fig. 5 (a)). Then, the user selects the offset distance along the piecewise linearized Pareto front, δ_J. The distance δ_J determines the final density of the Pareto solution distribution, because it becomes the maximum segment length during the last phase of the algorithm.

In order to find the offset distances parallel to the objective axes, the angle θ in Fig. 5 (b) is computed as

$$\theta = \tan^{-1}\left(-\frac{P_1^1 - P_2^1}{P_1^2 - P_2^2}\right) \tag{10}$$

where P_i^1 and P_i^2 are the J_1 and J_2 positions of the end points, $\mathbf{P}_1$ and $\mathbf{P}_2$, respectively, in the objective space.

Then, δ_1 and δ_2 are determined with δ_J and θ as follows,

$$\delta_1 = \delta_J \cos\theta \quad and \quad \delta_2 = \delta_J \sin\theta. \tag{11}$$

[Step 7] Impose additional inequality constraints and conduct suboptimization with the weighted sum method in each of the feasible regions. As shown in Fig. 5 (b), the feasible region is offset from $\mathbf{P}_1$ and $\mathbf{P}_2$ by the distance of δ_1 and δ_2 in the direction of J_1 and J_2. Perform suboptimization in this region. The suboptimization problem is stated as

$$\begin{aligned} \min \quad & \alpha \bar{J}_1(x) + (1-\alpha)\bar{J}_2(x) \\ \text{s.t.} \quad & \bar{J}_1(x) \le P_1^x - \delta_1 \\ & \bar{J}_2(x) \le P_2^y - \delta_2 \\ & h(x) = 0 \\ & g(x) \le 0 \\ & \alpha \in [0,1] \end{aligned} \tag{12}$$

where δ_1 and δ_2 are the offset distances obtained in Step 6, and P_i^x and P_i^y are the x and y position of the end points. The uniform step size of the weighting factor α_i for each feasible region is determined by the number of refinements, n_i, obtained in step 4:

$$\Delta\alpha_i = \frac{1}{n_i} \tag{13}$$

The segments in which no converged optimum solutions are obtained are removed from the segment set for further refinement, because in this case these regions are non-convex and do not contain Pareto optimal solutions.

[Step 8] Compute the length of the segments between all the neighboring solutions. Delete nearly overlapping solutions. If all segment lengths are less than a prescribed maximum length, terminate the optimization procedure. If there are segments whose lengths are greater than the maximum length, go to Step 4.

IV. Numerical examples

Three numerical examples are presented in this section to demonstrate the performance of the adaptive weighted sum method. All optimizations were performed with the Sequential Quadratic Method in MATLAB.

A. Example 1: Convex Pareto front

The first example is a multiobjective optimization problem that was investigated in the context of the NBI method[15]. The problem statement is

$$\begin{aligned} \text{minimize} \quad & \begin{bmatrix} J_1 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \\ J_2 = 3x_1 + 2x_2 - \dfrac{x_3}{3} + 0.01(x_4 - x_5)^3 \end{bmatrix} \\ \text{subject to} \quad & x_1 + 2x_2 - x_3 - 0.5x_4 + x_5 = 2, \\ & 4x_1 - 2x_2 + 0.8x_3 + 0.6x_4 + 0.5x_5^2 = 0, \\ & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \le 10 \end{aligned} \tag{14}$$

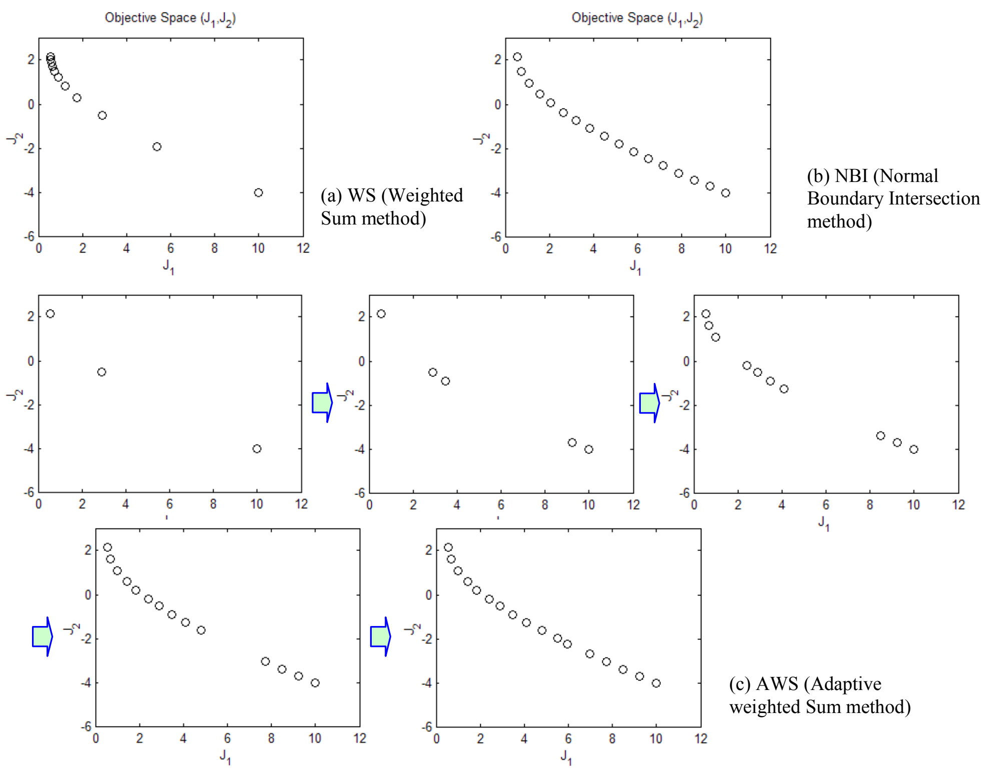


Figure 6: Results of multiobjective optimization with a convex Pareto front (example 1).

The Pareto front of this problem is convex, but the curvature is not uniform. Figure 6 (a) shows the optimal solutions obtained by the usual weighted sum method. The number of solutions on the Pareto front is 17, but most of the solutions are concentrated in the left upper region. The NBI method gives a very good approximation of the Pareto front by obtaining evenly distributed solutions, as shown in Fig. 6 (b).

The adaptive weighted sum method converges in five iterations, obtaining fairly well distributed solutions (Fig. 6 (c)). The offset distance selected on the Pareto front, δ_J, is 0.1; and the offset distances, δ_1 and δ_2, are calculated by Eq. (11). Multiplier C in Eq. (9) is 1.5. Table 1 provides a quantitative comparison of solutions in terms of computational cost (CPU time) and variance of segment lengths. The weighted sum method, the NBI method, and the adaptive weighed sum method are compared for the case of 17 solutions on the Pareto front. Although the weighted sum method has a small computational burden, its variance is very large. The NBI method has better performance both in terms of CPU time and secant length variance compared to the adaptive weighted sum method in this example. At this point it is not obvious why one might further pursue the adaptive weighted sum method. It has been observed that the NBI method usually performs better in the cases of well-conditioned multiobjective optimization problems with a convex Pareto front. However, the uniformity of the solutions obtained by the adaptive weighted sum

Table 1: Comparison of the results (example 1).

	WS	NBI	AWS
No of solutions	17	17	17
CPU time (sec)	1.71	2.43	3.83
Length variance ($\times 10^{-4}$)	266	0.23	2.3

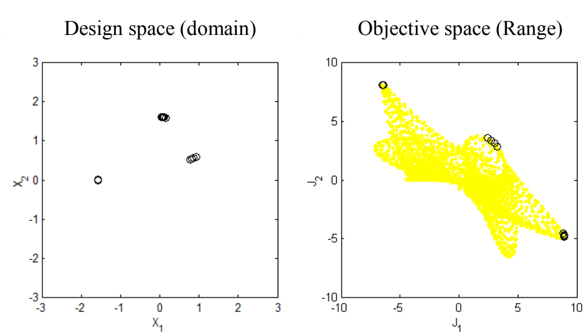


Figure 7: Results by the usual weighted sum method for multiobjective optimization with a non-convex Pareto front (example 2).

method is satisfactory according to the maximum length criterion, and the adaptive weighted sum method shows better performance in more complex problems, as demonstrated in the following example. The relatively heavy computational cost of the adaptive weighted sum approach is due to additional calculations, such as obtaining the distances between adjacent solutions and selecting segments for further refinement. This overhead will be less significant for large problems, where the cost of objective function evaluations typically dominates.

B. Example 2: Non-convex Pareto front

In the previous example, the Pareto front was convex, and the problem associated with the usual weighted sum approach was only that the solution distribution was not uniform. However, if the Pareto front is not convex, the weighted sum approach does not find concave parts, regardless of step size. In this example, a multiobjective optimization problem that has a partially non-convex Pareto front and that is not well-conditioned is considered. The problem statement[18] is

$$\text{minimize} \begin{bmatrix} 3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-x_1^2-x_2^2} - 3e^{-(x_1+2)^2-x_2^2} + 0.5(2x_1+x_2) \\ 3(1+x_2)^2 e^{-x_2^2-(-x_1+1)^2} - 10\left(-\frac{x_2}{5} + x_2^3 + x_1^5\right) e^{-x_2^2-x_1^2} - 3e^{-(2-x_2)^2-x_1^2} \end{bmatrix} \tag{15}$$

$$\text{subject to} \quad -3 \le x_i \le 3 \;, \quad i = 1, 2.$$

The solutions obtained by the usual weighted sum method are shown in Fig. 7. This figure shows the efficient designs in the design space on the left and the Pareto optimal solutions in the objective space on the right. The entire range in the objective space is obtained by a full combinatorial analysis. The difficulty in performing optimization

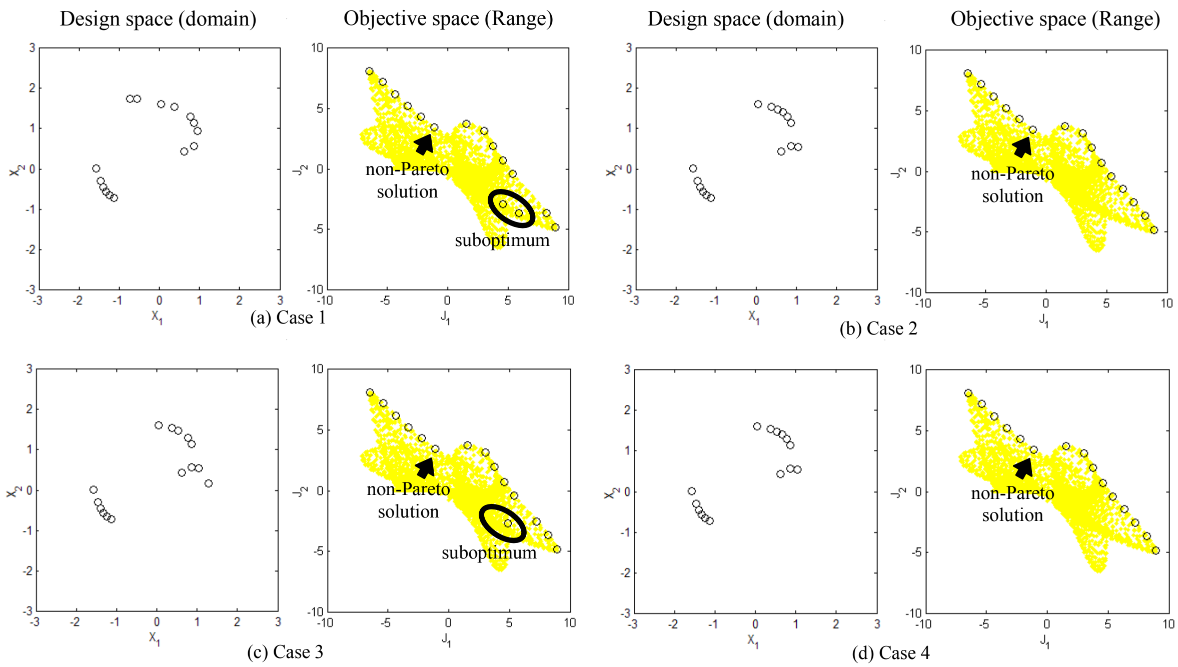


Figure 8: Results by the NBI method for multiobjective optimization with a non-convex Pareto front (example 2).

for this non-linear problem is that the convergence to an optimal solution is highly dependent on an initial starting point and determining the starting point is not straightforward. The solution dependence on the initial starting point is even more severe in the case of the NBI method and the adaptive weighted sum method than the usual weighted sum method. This is because the two methods use additional constraints and so it is difficult to find feasible regions that satisfy all the constraints. In the usual weighted sum method, three points ([1.5 0], [1 1] and [0 2]) are used as a starting point, and the best among the solutions is selected. As shown in Fig. 7, trying these three initial starting points always yields the optimum solutions for the usual weighted sum method. However, the solutions cluster around three small regions. The vast area of the two concave regions is not revealed by the traditional weighted sum method, which confirms the second drawback of the method mentioned in Section 2.

The NBI method and adaptive weighted sum method successfully find solutions on the non-convex regions. However, the solution dependence on the initial starting point is a serious concern for these methods. Hence, full combinatorial trials of initial starting points are used to better understand this issue. The domain is discretized into a grid composed of segments whose lengths are Δx_1 and Δx_2. The optimization is then started from all intersections of the grid. The best solution is then selected from among all the solutions obtained. Four different cases of the starting grid were tested for the NBI method and the adaptive weighted sum method:

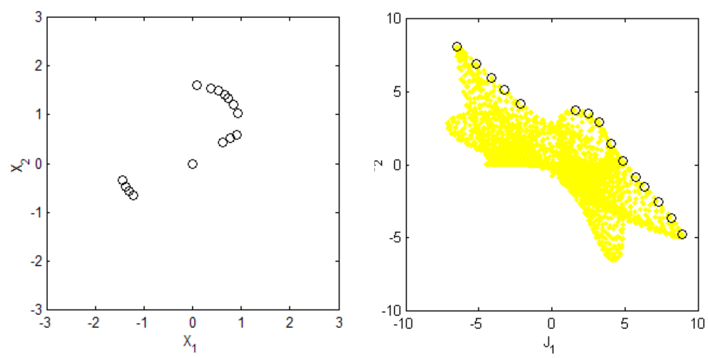


Figure 9: Results by the adaptive weighted sum method for multiobjective optimization with a non-convex Pareto front (example 2). Case 1, Case 2, Case 3 and Case 4 give the same results.

Case 1: $\Delta x_1 = \Delta x_2 = 2.0$

Case 2: $\Delta x_1 = \Delta x_2 = 1.5$

Case 3: $\Delta x_1 = \Delta x_2 = 1.0$

Case 4: $\Delta x_1 = \Delta x_2 = 0.5$

The solutions obtained by the NBI method for each of the cases are shown in Fig. 8. In all the four cases, one non-Pareto solution is obtained, which is dominated by its two neighboring solutions. Because of this problem, a Pareto filter needs to be applied for the NBI method after finding solutions. In addition, some suboptimal solutions are obtained: two suboptimal solutions for Case 1 and one suboptimal solution for Case 3. When the adaptive weighted sum method is used, on the other hand, all the solutions obtained are truly Pareto optimal, as shown in Fig. 9. Only one case is represented in the figure because the solutions are identical for all four cases. The offset distance on the Pareto front, δ_J, is 0.1. Multiplier, C, is 1.0. Note that non-Pareto optimum or suboptimal solutions are not obtained in the adaptive weighted sum method. The reason for its robustness in finding Pareto optimal solutions is that it uses inequality constraints rather than equality constraints, which makes it easier to find the feasible domain during optimization.

Table 2: Comparison of the results (example 2).

	WS	**NBI**				**AWS**			
Initial starting point case		Case 1	Case 2	Case 3	Case 4	Case 1	Case 2	Case 3	Case 4
No of solutions	15	15	15	15	15	15	15	15	15
CPU time (sec)	0.4	17.8	24.5	52.9	165.6	28.1	44.0	87.6	289.2
Length variance ($\times 10^{-4}$)	632	11	3.6	8.8	3.6	4.3	4.3	4.3	4.3
No of suboptimum solutions	0	2	0	1	0	0	0	0	0
No of non-Pareto solutions	0	1	1	1	1	0	0	0	0

This example demonstrates the advantages of the adaptive weighted sum method: (1) it finds solutions of even distribution; (2) it can find solutions on non-convex regions; (3) non-Pareto solutions on the non-convex regions are not considered as an optimum because they are not in the feasible region bounded by the additional constraints; (4) it is potentially more robust in finding optimum solutions than other methods that use equality constraints. The solution comparison of each method for this example is provided in Table 2.

C. Example 3: Three-bar truss problem

Finally, the adaptive weighted sum method is applied to a three-bar truss problem first presented by Koski[17]. Figure 10 illustrates the problem and shows the values of the parameters used. A horizontal load and a vertical load are applied at Point P, and the objective functions are the total volume of the trusses and the displacement of point P. The mathematical problem statement is

$$\begin{aligned} &\text{minimize} \quad \begin{bmatrix} \text{volume } (\mathbf{A}) \\ \Delta(\mathbf{A}) \end{bmatrix} \\ &\text{subject to} \quad \sigma_{\text{lower limit}} \le \sigma_i \le \sigma_{\text{upper limit}} \quad , i = 1,2,3 \\ &\qquad\qquad\quad A_{\text{lower limit}} \le A_i \le A_{\text{upper limit}} \quad , i = 1,2,3 \\ &\text{where} \quad \Delta = 0.25\delta_1 + 0.75\delta_2 \\ &\qquad\quad \text{and} \quad \mathbf{A} = [A_1 \ A_2 \ A_3]. \end{aligned} \tag{16}$$

The Pareto front for this example is non-convex, and the Pareto line is separated into two regions by a segment of dominated solutions, as shown in Fig. 11. The adaptive weighted sum method with 0.1 of offset δ_J is used. Multiplier, C, is 1.5. The optimization history is shown in the figure. The adaptive weighted sum method converges

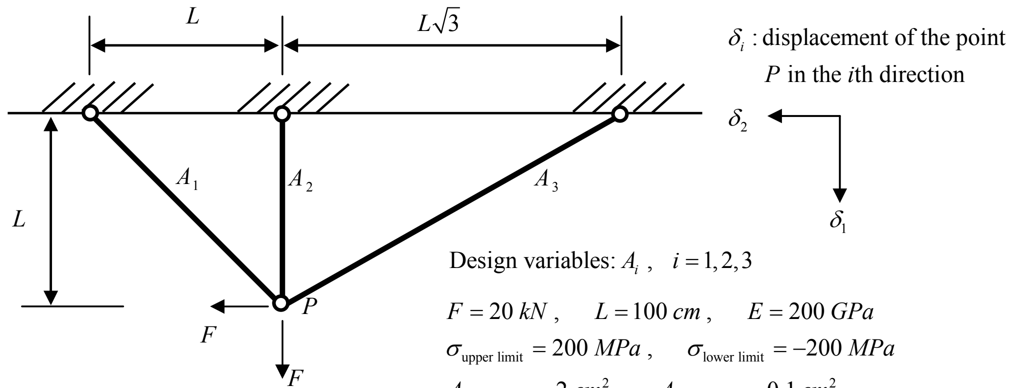


Figure 10: The three-bar truss problem[17].

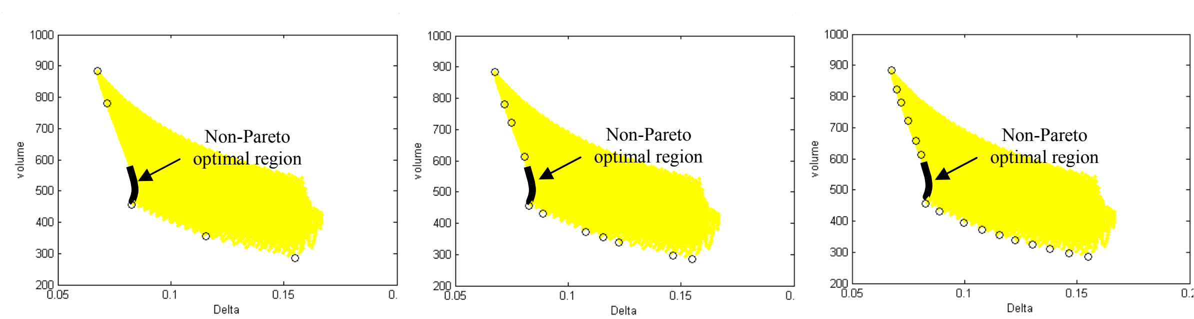
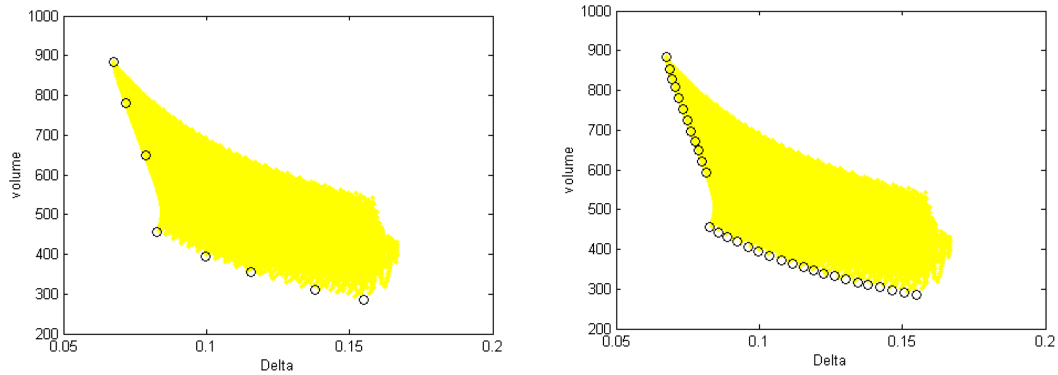


Figure 11: Optimization history by the adaptive weighted sum method (example 3). $\delta_J = 0.1$. It converges in three phases.

in three phases, and the solutions are quite evenly distributed. Note that there is no solution obtained in the non-Pareto region, without using a Pareto filter. If one changes the value of the offset distance, δ_J, the density of final solutions changes. Figure 12 shows the two results when 0.2 and 0.05 are used as the offset distance, δ_J. The adaptive weighted sum method gives 8 and 32 evenly distributed Pareto solutions for each case. Again in this example, the distribution is uniform; the Pareto optimal solutions on the non-convex region are identified; and the non-Pareto optimal solutions are ignored. The parameter δ_J is used to tune the desired density of Pareto points generated by the algorithm.



(a) $\delta_J = 0.2$ (b) $\delta_J = 0.05$

Figure 12: Solutions for different offset distances (example 3).

V. Discussion

The adaptive weighted sum method effectively approximates the Pareto front by gradually increasing the number of solutions on the front. In that sense it gradually "learns" the shape of the Pareto front and concentrates computational effort where new information can be gained most effectively. This is in contrast to other Pareto generation methods such as traditional weighted sum or NBI, which generally explore the Pareto front in a predetermined fashion. Because it adaptively determines where to divide further, the adaptive weighted sum method produces well-distributed solutions. In addition, performing optimization only in feasible regions by imposing additional inequality constraints enables the method to find Pareto solutions on non-convex regions. Because the feasible region includes only the regions of non-dominated solutions, it automatically neglects non-Pareto optimal solutions. It is potentially more robust in finding optimal solutions than other methods where equality constraints are applied.

This article does not claim superiority of the adaptive weighted sum method over other methods such as NBI in all cases. Rather the method presents itself as a potential addition to the growing suite of Pareto generators, with potential advantages for ill-conditioned problems. Further work is needed to understand the nature of this advantage in terms of starting points, imposition of inequality constraints versus equality constraints and computational cost. It must also be said that while the traditional weighted sum method has known limitations, it remains the method offering greatest transparency to non-expert users. The adaptive weighted sum approach is an effective extension of traditional weighted sum optimization, but some of the transparency is invariably hidden from the user due to the adaptive scheme. The adaptive weighted sum method needs to be applied to multidimensional multiobjective optimization problems where there are more than two objective functions. Some multiobjective optimization algorithms perform well for bivariate problems, but perform poorly in the presence of multiple objectives. It remains to be seen how well adaptive weighted sum optimization can be scaled to problems of higher dimensionality.

References

[1] Pareto, V., *Manuale di Economia Politica*, Societa Editrice Libraria, Milano, Italy, Translated into English by Schwier, A. S. 1971: *Manual of Political Economy*, Macmillan, New York, 1906.

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

[3] 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.

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

[5] Koski, J., “Multicriteria truss optimization,” *Multicriteria Optimization in Engineering and in the Sciences*, edited by Stadler, New York, Plenum Press, 1988.
[6] Marglin, S. *Public Investment Criteria*, MIT Press, Cambridge, Massachusetts, 1967.
[7] Steuer, R. E., *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley & Sons, New York, 1986.
[8] 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.
[9] Suppapitnarm, A. et al., “Design by multiobjective optimization using simulated annealing,” *International conference on engineering design ICED 99*, Munich, Germany, 1999.
[10] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
[11] Fonseca, C., and Fleming, P., “An overview of evolutionary algorithms in multiobjective optimization,” *Evolutionary Computation*, Vol. 3, 1995, pp. 1-18.
[12] Tamaki, H., Kita, H., and Kobayashi, S., “Multiobjective optimization by genetic algorithms: a review,” *1996 IEEE International Conference on Evolutionary Computation*, ICEC ’96, Nagoya, Japan, 1996.
[13] 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.
[14] Mattson, C. A., and Messac, A., “Concept Selection Using s-Pareto Frontiers,” *AIAA Journal*. Vol. 41, 2003, pp. 1190-1204.
[15] 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.
[16] 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.
[17] Koski, J., “Defectiveness of weighting method in multicriterion optimization of structures,” *Communications in Applied Numerical Methods*, Vol. 1, 1985, pp. 333-337.
[18] de Weck, O., and Willcox, K., Multidisciplinary System Design Optimization, Lecture Note, M.I.T., 2003.