10th World Congress on Structural and Multidisciplinary Optimization
May 19 - 24, 2013, Orlando, Florida, USA

A gradient-based multiobjective optimization technique using an adaptive weighting method

Kazuhiro Izui, Takayuki Yamada and Shinji Nishiwaki

Department of Mechanical Engineering and Science, Kyoto University, Kyoto Japan
{izui, takayuki, shinji}@me.kyoto-u.ac.jp

1. Abstract
While various multiobjective optimization methods based on metaheuristic techniques have been proposed, these methods still encounter difficulties when handling many variables, or numerous objectives and constraints. This paper proposes a new aggregative gradient-based multiobjective optimization method for obtaining a Pareto-optimal solution set. In this method, the objective functions and constraints are evaluated at multiple points in the objective function space, and design variables at each point are updated using information aggregatively obtained from all other points. In the proposed method, a multiobjective optimization problem is converted to a single objective optimization problem using a weighting method, with weighting coefficients adaptively determined by solving a linear programming problem. A sequential linear programming technique is used to update the design variables, since it allows effective use of design sensitivities that can be easily obtained in many engineering optimization problems. Several numerical examples illustrate the effectiveness the proposed method.
2. Keywords: Design optimization, Multiobjective optimization, Gradient-based optimization, Adaptive weighting coefficient

3. Introduction

Obtaining an entire set of Pareto-optimal solutions can offer designers a clear picture of the trade-off relationships among the conflicting objective functions, and various multiobjective optimization methods have been proposed to obtain Pareto-optimal solution sets. In particular, multiobjective optimization methods based on metaheuristic techniques such as genetic algorithms [1][2][3], and particle swarm optimization [4][5], have been extensively studied. However, constraints cannot be explicitly handled in such methods, and the algorithms or objective functions must be modified to implement constraint handling [6][7][8]. Furthermore, metaheuristic-based methods are inefficient when searching for fine-tuned solutions once a nearly global optimum is found, since the algorithms do not include design sensitivities. Metaheuristic techniques are also not well-suited for handling large-scale problems that have many design variables. Therefore, this paper proposes a new gradient-based multiobjective optimization method which can utilize design sensitivity information when updating design variables.

4. Method of aggregative gradient-based multiobjective optimization

In this method, objective functions and constraints are evaluated at multiple points and the design variables at each point are updated using information aggregatively obtained from all other points in the objective function space. A typical nonlinear multiobjective optimization problem can be written as:

$$\text{minimize} \quad \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})]^{\mathrm{T}} \tag{1}$$

subject to:

$$\mathbf{g}(\mathbf{x}) \leq 0 \tag{2}$$

$$\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \tag{3}$$

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}, \tag{4}$$

where objective function vector $\mathbf{f}$ is a function of design variable vector $\mathbf{x}$, and $\mathbf{g}$ is a constraint vector. $\mathbf{x}_L$ and $\mathbf{x}_U$ respectively denote the lower and upper bound of the design variables.

The multiobjective optimization method proposed in this paper is based on the weighting method, and weighting coefficients are adaptively given during optimization process. In the proposed method,

weighing coefficients are determined using a Data Envelopment Analysis (DEA) technique. DEA is a tool originally used in the field of economics to evaluate the relative performance of decision-making units (DMUs) in multi-input and multi-output environments [9]. In this implementation, the performance values of a DMU for multiple criteria are converted, by solving a linear programming problem, to a single value, termed the efficiency value, which is then used to evaluate relative performance among multiple DMUs. An important feature of the DEA technique is that it can provide optimal weighting coefficients when the efficiency is calculated. In the proposed method, DEA is conducted for each point to obtain appropriate weighting coefficients.

In the case that all objective functions are to be minimized, the efficiency of the M-th point, θ^M, is calculated by solving the following linear programming problem.

$$\text{minimize} \quad \theta^M = \sum_{i=1}^{m} w_i^M f_i^M \quad \text{w.r.t.} \quad w_i^M \tag{5}$$

subject to:

$$\sum_{i=1}^{m} w_i^M f_i^k \geq 1 \quad (\text{for} \quad k = 1, 2, ..., K) \tag{6}$$

$$w_i^M \geq 0 \quad (\text{for} \quad i = 1, 2, ..., m), \tag{7}$$

where K is the total number of points, f_i^k is the k-th point's i-th objective function value, and w_i^M represents the weighting coefficients. If the M-th point is a non-dominated point among K points, θ^M becomes 1, and for dominated points, θ^M becomes larger than 1.

The weighting coefficients calculated by the DEA act to minimize θ^M, which is converted to a single objective function by the weighting method. Therefore, if the M-th point has a smaller value of f_1^M and a larger value of f_2^M than the other points in a bi-objective problem, the above linear programming problem returns a larger w_1^M value and a smaller w_2^M value, which increases the importance of the first objective function. The use of such calculated weighting coefficients when the design variables are subsequently updated is the main idea of the proposed method.

4. Procedures

The aggregative gradient-based multiobjective optimization procedure is now described in detail. Figure 1 shows the flowchart of the procedure.

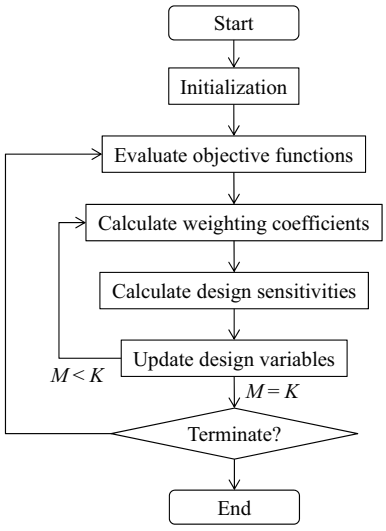

Figure 1: Flowchart

- **Step 1: Initialization**

  Generate initial design variables with random values for K points.

- **Step 2: Calculate objective functions**

  Evaluate objective functions for all K points.

- **Step 3: Calculate weighting coefficients**

  Solve the linear programming problem shown in Eqs. (5) through (7) for all points and obtain adaptive weighting coefficients w_i^M. M is then set to 1.

- **Step 4: Calculate sensitivities**

  Evaluate sensitivities of the objective functions and constraint functions for the M-th point.

- **Step 5: Update design variables**

  Update the design variables of the M-th point, minimizing the weighted sum of the objective functions using weighting coefficients w_i^M. A Sequential Linear Programming (SLP)-based updating scheme is used in Step 5 since it can stably handle a large number of design variables using well-established linear programing solvers. That is, the single objective optimization problem, converted from the multiobjective optimization problem through the use of adaptive weighting coefficients, is linearly approximated. In this step, the following approximated linear programming problem is solved to update the design variables of the point.

$$\min \quad f^M = \sum_{i=1}^{m} w_i^M \sum_{j=1}^{n} \frac{\partial f_i(\mathbf{x^M})}{\partial x_j} x_j \quad \text{w.r.t.} \quad x_j \tag{8}$$

  subject to:

$$g_s(\mathbf{x^M}) + \sum_{j=1}^{n} \frac{\partial g_s(\mathbf{x^M})}{\partial x_j}(x_j - x_j^M) \leq 0$$

$$(\text{for} \quad s = 1, 2, ..., t) \tag{9}$$

$$\tilde{\mathbf{x}}_L \leq \mathbf{x} \leq \tilde{\mathbf{x}}_U, \tag{10}$$

  where f^M is the weighted sum of the objective functions obtained in Step 3, and $\mathbf{x^M}$ is the design variable vector of the M-th point before updating. $\tilde{\mathbf{x}}_L$ and $\tilde{\mathbf{x}}_U$ are respectively the lower and upper bound for this linear programming problem considering the moving limit of the design variables. If $M = K$, the procedure then advances to Step 6 (Check termination condition), otherwise $M = M+1$ and the procedure returns to Step 4.

- **Step 6: Check termination condition**

  If the termination condition is satisfied, the procedure ends, otherwise it returns to Step 2.

Note that in the above procedure, two linear programming problems are solved for each point during a single iteration: 1) to determine weighting coefficient values, and 2) to update the design variables. The weighting coefficients for each point are therefore adaptively updated at every iteration. We note that although our method utilizes the concept of a weighting method, it does not require setting the weighting coefficients to predetermined values.

5. Examples

5.1. Example 1

The proposed method is now applied to two numerical examples to demonstrate its performance. In these examples, values 5% larger or smaller than those of the design variables were used as SLP move limits. First, the proposed method was applied to a test function [10]. The problem is stated as follows.

$$f_1 = x_1^4 + x_2^4 - x_1^2 + x_2^2 - 10x_1x_2 + 0.25x_1 + 20 \tag{11}$$

$$f_2 = (x_1 - 1)^2 + x_2^2 \tag{12}$$

$$-2 \leq x_1, x_2 \leq 2 \tag{13}$$

Figure 2 shows the results for this problem, where K was set to 40. The eight-point stars indicate the initial points and the "+" symbols represent the obtained solutions. The circled "+" marks indicate

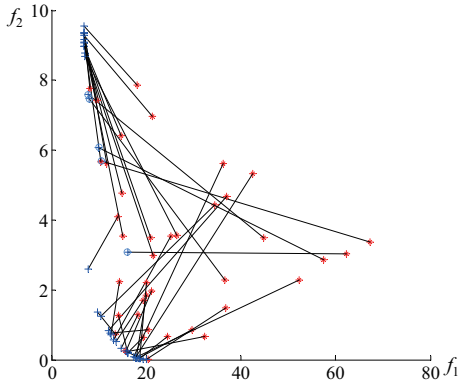

Figure 2: Obtained solutions in Example 1

dominated solutions that represent local optima for this problem. This result illustrates that the proposed method can provide true Pareto-optimal solutions, although several points are stacked at local optima. In other words, the multi-point aggregative approach that our method uses can minimize the selection of local optima in multimodal problems.

5.2. Example 2

Next, the proposed method is used to solve the topology optimization problem illustrated in Fig. 3. A downward traction $\mathbf{f}$ is applied at the lower right-hand corner of the design domain. The objective functions are the mean compliance and total volume, and both are minimized. The density method is used in this problem and the design domain is discretized into 60×40 elements. This problem is therefore a large-scale problem with 2,400 design variables, and the range of each design variable was set to $[0, 1]$. We note that a metaheuristics-based multiobjective optimization method could not be applied to such a large-scale problem.

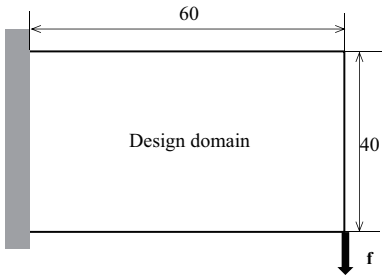

Figure 3: Topology optimization problem

The multiobjective topology optimization is formulated as follows.

$$f_1 = \mathbf{u}^{\mathrm{T}}\mathbf{K}\mathbf{u} \tag{14}$$

$$f_2 = V \tag{15}$$

subject to:

$$\mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \tag{16}$$

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{17}$$

where $\mathbf{u}$ and $\mathbf{f}$ are respectively the displacement and force vector, $\mathbf{K}$ is the stiffness matrix, and V denotes the total volume.

Figure 4 shows the non-dominated solutions obtained by the proposed method when K was set to 60. Four selected points, A–D in this figure, have corresponding configurations shown in Fig. 5. The illustrated configurations demonstrate that the proposed method can effectively obtain non-dominated

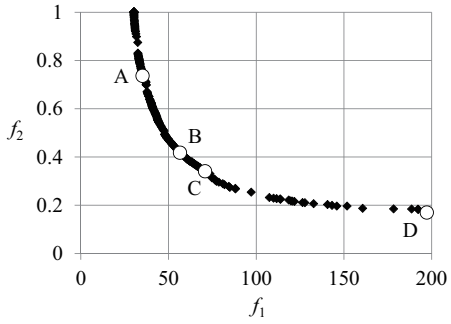

Figure 4: Non-dominated solutions in Example 2

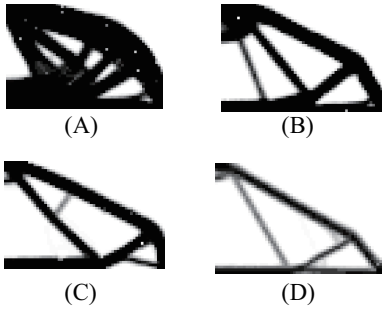

Figure 5: Examples of obtained solutions in Example 2

solutions for topology optimization problems that include a large number of design variables, since design sensitivities are used when updating the design variables. We note that conventional multiobjective optimization methods employing metaheuristic techniques, classified as direct methods since design sensitivities are not used in the optimization process, are almost always unsuitable for such large-scale problems.

6. Conclusions

This paper proposed a new gradient-based multiobjective optimization method to obtain Pareto-optimal solutions. The proposed method uses a weighting method to convert a multiobjective optimization problem to a single objective optimization problem, and a linear programming problem is solved to determine adaptive weighting coefficients for each point while considering the point's position relative to all other points in the objective function space. The converted single objective optimization problem is linearly approximated, and the design variables are updated by a linear programming technique.

7. References

[1] Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Professional (1989)

[2] Tamaki, H., Kita, H., Kobayashi, S.: Multi-objective optimization by genetic algorithms: A review. In Proceedings of 1996 IEEE International Conference on Evolutionary Computation, pp. 517-522 (1996)

[3] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182-197 (2002)

[4] Coello, C.A.C., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 256-279 (2004)

[**5**] Reyes-Sierra, M., Coello, C.C.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research 2(3), 287-308 (2006)

[**6**] Cai, Z., Wang, Y.: A multiobjective optimization-based evolutionary algorithm for constrained optimization. IEEE Transactions on Evolutionary Computation 10(6), 658-675 (2006)

[**7**] Qu, B.Y., Suganthan, P.N.: Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods, Engineering Optimization 43(4), 403-416 (2011)

[**8**] Cagnina, L.C., Esquivel, S.C., Coello, C.A.C.: Solving constrained optimization problems with a hybrid particle swarm optimization algorithm. Engineering Optimization 43(8), 843-866 (2011)

[**9**] Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. European journal of operational research 2(6), 429-444 (1978)

[**10**] Preuss, M., Naujoks, B., Rudolph, G.: Pareto set and emoa behavior for simple multimodal multiobjective functions. Parallel Problem Solving from Nature-PPSN IX pp. 513-522 (2006)