Honglei Xu · Song Wang
Soon-Yi Wu *Editors*

Optimization Methods, Theory and Applications

Springer

Chapter 3
An Extension of the MOON^2/MOON^{2R} Approach to Many-Objective Optimization Problems

Yoshiaki Shimizu

Abstract A multi-objective optimization (MUOP) method that supports agile and flexible decision making to be able to handle complex and diverse decision environments has been in high demand. This study proposes a general idea for solving many-objective optimization (MAOP) problems by using the MOON^2 or MOON^{2R} method. These MUOP methods rely on prior articulation in trade-off analysis among conflicting objectives. Despite requiring only simple and relative responses, the decision maker's trade-off analysis becomes rather difficult in the case of MAOP problems, in which the number of objective functions to be considered is larger than in MUOP. To overcome this difficulty, we present a stepwise procedure that is extensively used in the analytic hierarchy process. After that, the effectiveness of the proposed method is verified by applying it to an actual problem. Finally, a general discussion is presented to outline the direction of future work in this area.

Keywords Many-objective optimization • MOON^2 • MOON^{2R} • Pairwise comparison • AHP • Neural network

3.1 Introduction

A multi-objective optimization (MUOP) method that supports flexible and adaptive decision making for application in complex, diverse, and competitive environments has been in high demand. Notably, MUOP applies to problems involving incommensurable objectives that conflict or compete with each other. Although Pareto optimal solutions represent a rational norm in MUOP, there can be an infinite

---

This paper was presented at ICOTA 2013 held in Taipei., Taiwan.

Y. Shimizu (✉)
Department of Mechanical Engineering, Toyohashi University of Technology,
Toyohashi, Aichi 441-8580, Japan
e-mail: shimizu@me.tut.ac.jp

H. Xu et al. (eds.), *Optimization Methods, Theory and Applications*,
DOI 10.1007/978-3-662-47044-2_3

number of members of this class. The set of optimal solutions is known as the Pareto front. Generally speaking, however, decision making as an engineering task aims at obtaining a limited number of candidates for the final decision.

From this viewpoint, this study proposes a general idea for solving the many-objective optimization (MAOP) problem in which more than several objective functions are considered simultaneously. Effort is devoted to obtain a unique solution known as the preferentially optimal solution or the best-compromise solution. This approach is notably different from that of multi-objective evolutionary algorithms (MOEA), which attempt to derive only the Pareto front (Coello 2001; Czyzak and Jaszkiewicz 1998; Deb et al. 2000; Jaeggi et al. 2005; Robic and Filipic 2005). However, recent studies have revealed that even in MOEA, conventional methods are not necessarily effective for dealing with MAOP problems (Hughes 2005; Sato et al. 2010).

In this context, we extend our previously proposed methods, named MOON^2 and MOON^{2R} (Shimizu and Kawada 2002; Shimizu et al. 2004), to be able to handle MAOP problems. Although MOON^2 and MOON^{2R} require only simple and relative responses, handling the decision makers' (DMs') responses in trade-off analysis becomes rather difficult in MAOP. To overcome this difficulty, this study proposes an approach that is easily applicable to MAOP. Consequently, the proposed idea can extend the applicability and practicality of existing methods to the complex decision making environments mentioned above.

The rest of this chapter is organized as follows. In Sect. 3.2, the general procedures of MOON^2 and MOON^{2R} are explained. Section 3.3 extends this procedure to MAOP. In Sect. 3.4, the validity and effectiveness of the proposed method is verified by applying it to an actual problem. A general discussion is also presented in that section to give a definite and comprehensive outline of the direction of future work in this area. A conclusion is given in Sect. 3.5.

3.2 MOON^2 and MOON^{2R} for MUOP and MAOP

General MUOP problems are described as follows.

$$\text{(p. 1)} \quad Min\, \boldsymbol{f}(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_N(\boldsymbol{x})\}\ subject\ \ to\ \boldsymbol{x} \in \mathrm{X},$$

where $\boldsymbol{x}$ denotes a decision variable vector; X, is a feasible region; and $\boldsymbol{f}$ is an objective function vector, some elements of which are incommensurable and conflict with one another. When $n > 3$, this problem is commonly referred to as a MAOP problem. The abbreviation MOP is used below in cases where the distinction between MUOP and MAOP is irrelevant.

As a particular characteristic of MOP, in addition to the mathematical procedures, we need some information on the DM's preference to obtain the best-compromise solution as a final goal. The solution methods of MOP problems are generally classified as prior articulation methods or interactive methods (Shimizu 2010). Naturally, each of these conventional methods has both advantages and disadvantages. For example, since in the former method a value function is derived separately from the search process, the DM does not need to perform repeated interactions during the search process, whereas such interactions are required in the latter method. On the other hand, although the latter method allows for elaborate articulation of attainability among the conflicting objectives, such articulation is difficult to obtain with the former method. Consequently, the derived solution may sometimes differ substantially from the best-compromise solution provided by the DM.

MOEA methods, which differ substantially from the two methods mentioned above, have been developed recently. However, these methods require further steps before attaining the final solution because the DM has to find the best solution among a potentially large number of candidates scattered along the Pareto front. In contrast, MOON^2 and MOON^{2R} can readily derive the best-compromise solution while being free from the requirement of repeated responses during the search, without giving up elaborate trade-off analysis. Therefore, MOON^2 and MOON^{2R} are expected to serve as powerful tools for enabling flexible decision making in agile engineering under diverse customer requirements.

Because MOON^2 and MOON^{2R} belong to the prior articulation methods in MOP, they have to identify the value function of the DM in advance. Such modeling can be performed with a suitable artificial neural network to deal with the non-linearity commonly seen in the value function. A back-propagation network (BPN) is used in MOON^2, while MOON^{2R} employs a radial-basis function network (RBFN).

To train the neural network, training data representing the preferences of the DM should be gathered by an appropriate means. These methods use pairwise comparison among the appropriate trial solutions, which are spread over the search area in the objective-function space. It is natural to constrain this modeling space to within the convex hull enclosed by the utopia and nadir solutions, which are defined as $\boldsymbol{f}^* = (f_1(\boldsymbol{x}^{utop}), f_2(\boldsymbol{x}^{utop}), \ldots, f_N(\boldsymbol{x}^{utop}))^{\mathrm{T}}$ and $\boldsymbol{f}_* = (f_1(\boldsymbol{x}^{nad}), f_2(\boldsymbol{x}^{nad}), \ldots, f_N(\boldsymbol{x}^{nad}))^{\mathrm{T}}$, respectively, where $\boldsymbol{x}^{utop}$ and $\boldsymbol{x}^{nad}$ are the respective utopia and nadir solutions in the decision variable space.

Then, the DM is asked to indicate the preferred solution and the spacing between each pair of trial solutions, for example, $\boldsymbol{f}^i = \boldsymbol{f}(\boldsymbol{x}^i)$ and $\boldsymbol{f}^j = \boldsymbol{f}(\boldsymbol{x}^j)$, $\boldsymbol{x}^i, \boldsymbol{x}^j \in \mathrm{X}$. These responses are provided in the form of linguistic statements, which are later transformed into scores denoted as a_{ij} (Table 3.1), similarly to the analytic hierarchy process (AHP) (Saaty 1980). For example, when the answer is such that $\boldsymbol{f}^i$ is strongly preferable to $\boldsymbol{f}^j$, a_{ij} takes a value of 5 (Table 3.1).

Table 3.1 Conversion table for linguistic statements

Linguistic statement	a_{ij}
Equally	1
Moderately	3
Strongly	5
Demonstrably	7
Extremely	9
Intermediate values between adjacent statements	2, 4, 6, 8

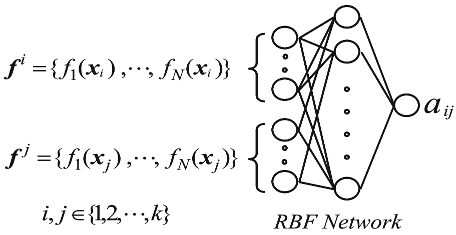

Fig. 3.1 Learning process using RBFN

Table 3.2 Pairwise comparison

	$\boldsymbol{f}^1$	$\boldsymbol{f}^2$	$\boldsymbol{f}^3$	$\cdots$	$\boldsymbol{f}^k$
$\boldsymbol{f}^1$	1	a_{12}	a_{13}	$\cdots$	a_{1k}
$\boldsymbol{f}^2$		1	a_{23}	$\cdots$	a_{2k}
$\boldsymbol{f}^3$			1	$\cdots$	$\vdots$
$\vdots$	$a_{ij} = 1/a_{ij}$			$\ddots$	$\vdots$
$\boldsymbol{f}^k$					1

By performing such pairwise comparisons over k trial solutions, we can obtain a pairwise comparison matrix (PCM) (Table 3.2). Element a_{ij} represents the degree of preference of $\boldsymbol{f}^j$ compared to $\boldsymbol{f}^i$. Note that although a_{ij} is defined as the ratio of relative degrees of preference, it does not necessarily mean that $\boldsymbol{f}^i$ is a_{ij} times more preferable to $\boldsymbol{f}^j$. According to the same conditions as AHP, such that $a_{ii} = 1$ and $a_{ji} = 1/a_{ij}$, the DM is required to provide $k(k\text{-}1)/2$ responses for the pairs highlighted in Table 3.2. Under these conditions, it is also easy to examine the consistency of such pairwise comparisons from the consistency index CI used in AHP.

Since information on the preferences of the DM is embedded in the PCM, we can derive a value function based on it. However, in general, it is almost impossible to give a mathematically definite form of the value function, as it is likely to be highly nonlinear. Unstructured modeling techniques that use neural networks are suitable for modeling in such situations. All objective values of each pair $\boldsymbol{f}^i$ and $\boldsymbol{f}^j$ ($\forall i, j \in \{1, 2, \ldots, k\}$) are used as $2N$ inputs of the neural network, and a_{ij} is the single output. Hence, PCM provides a total of k^2 training data sets for the neural network. Eventually, the trained neural network can be viewed as an implicit function mapping the $2N$ dimensional space to the scalar space (i.e., $V_{NN} : (\boldsymbol{f}^i, \boldsymbol{f}^j) \in \mathrm{R}^{2N} \rightarrow a_{ij} \in \mathrm{R}$).

Next, looking at the relations in Eq. (3.1), we can easily compare the preferences for any pair of solutions. Therefore, by fixing one of the input vectors of the neural network at an appropriate reference vector $\boldsymbol{f}^{\mathrm{R}}$, we can evaluate any solution from the output of the neural network (Eq. (3.2)). In other words, V_{NN} can serve as a value function. We can nominate some candidates for the reference point $\boldsymbol{f}^{\mathrm{R}}$, such as utopia, nadir, a center of gravity between them, or the point where the total sum of distances from all trial points is a minimum.

$$V_{NN}\left(\boldsymbol{f}^i,\boldsymbol{f}^k\right) = a_{ik} > V_{NN}\left(\boldsymbol{f}^j,\boldsymbol{f}^k\right) = a_{jk} \iff \boldsymbol{f}^i \succ \boldsymbol{f}^j \tag{3.1}$$

$$V_{NN}\left(\boldsymbol{f}\left(\boldsymbol{x}\right),\boldsymbol{f}^{\mathrm{R}}\right) = a_{x\mathrm{R}} > V_{NN}\left(\boldsymbol{f}\left(\boldsymbol{y}\right),\boldsymbol{f}^{\mathrm{R}}\right) = a_{y\mathrm{R}} \iff f(\boldsymbol{x}) \succ f(\boldsymbol{y}),\ \forall \boldsymbol{x},\boldsymbol{y} \in X \tag{3.2}$$

Once the value function is identified, the original MOP problem is transformed into an ordinal single-objective problem.

$$(p.2)\quad Max\quad V_{NN}\left(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{f}^R\right) \text{ subject to } \boldsymbol{x} \in X$$

Because the value function is built separately from the search process, a DM can carry out trade-off analyses whatever pace is desired without having to provide immediate responses or wait for queries, as is often required in interactive methods. In addition, because the required responses are simple and relative, the load on the DM in such interaction is rather small. These are some of the notable advantages of this approach. Moreover, the following proposition supports the validity of the above formulation.

[Proposition] The optimal solution of Problem (p. 2) is a Pareto optimal solution of Problem (p. 1) if the value function is chosen so as to satisfy the relation given by Eq. (3.1).

(Proof) Let $\widehat{f}_i^*$, $(i=1, \ \ldots, N)$ be the values of the objective functions for the optimal solution $\widehat{\boldsymbol{x}}^*$ of Problem (p. 2), so that $\widehat{f}_i^* = f_i\left(\widehat{\boldsymbol{x}}^*\right)$. Here, let us assume for contradiction that $\widehat{\boldsymbol{f}}^*$ is not a Pareto optimal solution. Then there exists a certain $\boldsymbol{f}^0$ such that for $\exists j$, $f_j^0 < \widehat{f}_j^* - \Delta f_j$, $\left(\Delta f_j > 0\right)$) and $f_j^0 \leq \widehat{f}_i^*$, $(i=1, \ \ldots, N$, $i \neq j)$. Because the DM apparently prefers $\boldsymbol{f}^0$ to $\widehat{\boldsymbol{f}}^*$, it holds that $V_{NN}\left(\boldsymbol{f}^0,\boldsymbol{f}^R\right) > V_{NN}\left(\widehat{\boldsymbol{f}}^*,\boldsymbol{f}^R\right)$. This contradicts that $\widehat{\boldsymbol{f}}^*$ is the optimal solution of Problem (p. 2). Hence, $\widehat{\boldsymbol{f}}^*$ must be a Pareto optimal solution.

Once $\boldsymbol{x}$ is given, we can readily evaluate any candidate solution through V_{NN}. Hence, it is possible to choose the most appropriate method from among a variety of conventional single-objective optimization methods. In addition to direct methods, meta-heuristic methods such as genetic algorithms, simulated annealing and tabu search are also applicable. At the same time, it is almost impossible to apply any of the interactive methods of MOP due to the large number of interactions during the search, which are likely to make the DM rather careless in providing responses.

When this approach is applied with an algorithm that requires the gradients of the objective function, such as nonlinear programming, we need to obtain these gradients by numeric differentiation. The derivative of the value function with respect to a decision variable is calculated by using the following chain rule.

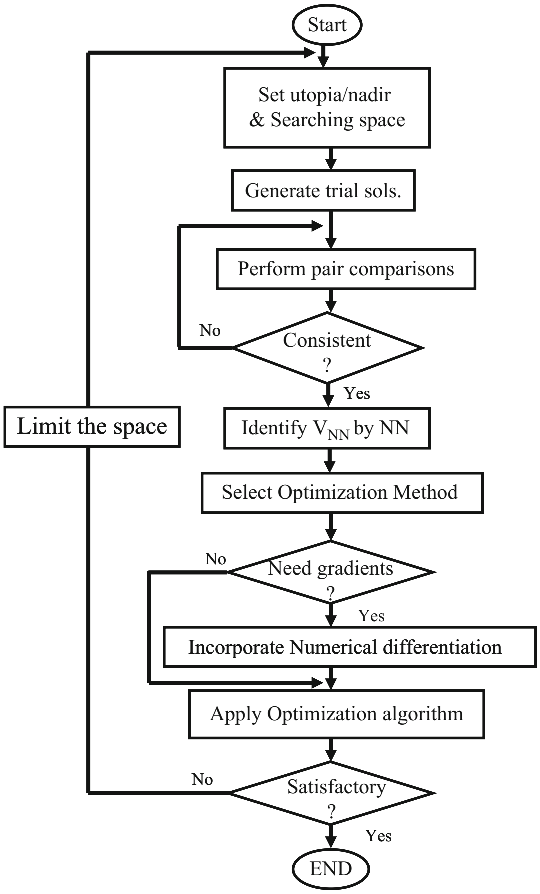

Fig. 3.2 Flowchart of the proposed method

$$\frac{\partial V_{NN}\left(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{f}^R\right)}{\partial \boldsymbol{x}} = \left(\frac{\partial V_{NN}\left(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{f}^R\right)}{\partial \boldsymbol{f}(\boldsymbol{x})}\right)\left(\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) \tag{3.3}$$

The derivative can be calculated from the analytic form of the second part in the right-hand side of Eq. (3.3) and the following numeric differentiation. Since most nonlinear programming software supports numeric differentiation, the algorithm can be realized without any special concerns.

$$\frac{\partial V_{NN}}{\partial f_i} \cong \frac{V_{NN}\left(\cdots f_i(\mathbf{x})+\Delta f_i,\cdots;\boldsymbol{f}^R\right)-V_{NN}\left(\cdots f_i(\mathbf{x}),\cdots;\boldsymbol{f}^R\right)}{\Delta f_i} \tag{3.4}$$

The proposed procedure can be summarized as follows (Fig. 3.2).

Step 1: Generate several trial solutions in the objective-function space.

Step 2: Extract the preferences of the DM through pairwise comparison between every pair of trial solutions.
Step 3: Train the neural network with the preference information obtained from the above responses. This network serves as a value function V_{NN} by selecting a certain reference solution $\boldsymbol{f}^{\mathrm{R}}$.
Step 4: Apply an appropriate optimization method to solve the resulting Problem (p. 2).
Step 5: If the DM is not satisfied with the result obtained in the above process, limit the search space around that result and repeat the same procedure until he or she accepts the result.

3.3 Procedure for MAOP

Because the aforementioned methods are natural and easy to work with for value assessments by humans, we have applied them to various problems and confirmed their effectiveness (Shimizu et al. 2005, 2006, 2010, 2012a; Shimizu and Tanaka 2003; Shimizu and Nomachi 2008). However, the case of MAOP is different if we consider the limit to the abilities of humans to perform assessment. As the number of objective functions increases, the difficulty of such value assessment through pairwise comparison increases rapidly. For example, suppose that a customer intends to buy a ticket for transportation in a certain situation. It seems rather easy to choose between a pair of candidates if they are evaluated on only two objectives, such as travel time and expense. According to the procedure outlined above, in this case, the customer has to make a pairwise comparison between the pair of solutions (i, j) in terms of the objectives (time i, cost i) and (time j, cost j), respectively. However, what will happen if there are more objectives to be compared? Suppose that the customer has to compare a pair of candidates in terms of four objectives: time, cost, service and comfort. Undoubtedly, the difficulty of assessment will grow substantially, and the customer may often give up on the comparison altogether, except in special cases.

For MAOP, therefore, it is impractical to deploy the proposed idea while maintaining the portability of the previous method. The basic idea of the proposed procedure involves replacing the pairwise comparison on many objectives with a comparison on a scalar objective. Assuming independence of the objective functions of (p.1), this procedure can be realized by the following steps.

Step 1: Determine the relative importance among the objective functions as weights $w_k, \quad (k = 1, \ldots, \ N)$, such that $\sum_k w_k = 1$, through pairwise comparison and eigenvalue calculation, as in AHP. Repeat if the pairwise comparison fails the consistency test.
Step 2: Narrowing the focus to the kth objective function only, ask the DM to give a preference for every pair of trial solutions $\boldsymbol{f}^i = \boldsymbol{f}\left(\boldsymbol{x}^i\right)$ and $\boldsymbol{f}^j = \boldsymbol{f}\left(\boldsymbol{x}^j\right)$, $\{f_k(\boldsymbol{x}^i)$,

$f_k(\boldsymbol{x}^j)\}$ $(\forall i,\ j,\ i > j)$, and obtain the preference intensity $s_k^i, \forall i$ by calculating the eigenvalues of this PCM. Repeat this process for every objective function.

Step 3: Calculate the total preference of the ith trial as $S_i = \sum_{k=1}^{N} w_k s_k^i, \forall i.$

Step 4: Finally, calculate a_{ij}, which is the PCM element corresponding to the preference between $\boldsymbol{f}^i$ and $\boldsymbol{f}^j$, as $a_{ij} = S_i/S_j$.

Step 5: Similarly to the previous step, identify the value function of the DM from $\boldsymbol{f}^i$ and $\boldsymbol{f}^j$ as the inputs and a_{ij} as the output of the neural network.

The above procedure can be easily implemented by a DM who is familiar with AHP, and does not introduce additional complexity to the original procedures of MOON^2 and MOON^{2R}.

3.4 Case Study

3.4.1 Evaluation Method

To verify the feasibility of our approach, we applied it to a problem assuming a virtual DM whose value function is given by Eq. (3.5) as a reference. We compared the result obtained by the proposed method with that from the optimization problem by using the following comprehensive objective function of (p.1):

$$U(\boldsymbol{f}(\boldsymbol{x})) = \left\{ \sum_{k}^{N} w_k \left(\frac{f_k(\boldsymbol{x}) - f_{k*}}{f_k^* - f_{k*}} \right)^t \right\}^{1/t}, \tag{3.5}$$

where f_k^* and f_{k*} denote the utopia and nadir values of the kth objective function, respectively. Moreover, w_k and t are a weight representing relative importance and a norm parameter, respectively. Hence, $U(\boldsymbol{f}(\boldsymbol{x}))$ represents the attainability ratio for utopia and takes a value of 1.0 for the utopia and 0.0 for the nadir.

We carried out the experiment along with the following procedures that correspond to those in Sect. 3.3.

Step 1: Determine a set of weights w_k $(\sum w_k = 1)$, each of which stands for the relative importance of the corresponding objective function regarding the preference.

Step 2: Instead of interactive pairwise comparison, for the virtual DM, obtain the preference index s_k^i of the ith trial for the kth objective as $s_k^i = \left(\frac{f_k^i - f_{k*}}{f_k^* - f_{k*}}\right)^t$, $(\forall k,\ \forall i)$.

Step 3: Calculate the total preference score S^i as $S^i = \left(\sum_{k}^{N} w_k s_k^i \right)^{1/t}$, $(\forall i)$.

shimizu@me.tut.ac.jp

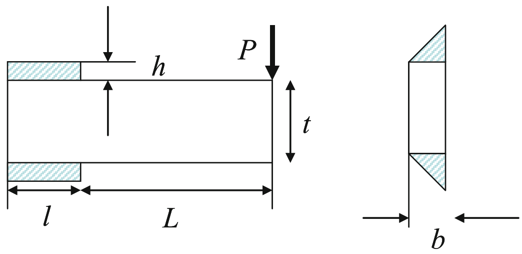

Fig. 3.3 Welded beam design

Step 4: Obtain the *ij*th element of the PCM as $a_{ij} = S^i/S^j$

Step 5: By using the data obtained above, train the neural network so that the relation $V_{NN}\left(\boldsymbol{f}^i,\ \boldsymbol{f}^j\right) = a_{ij}, (\forall i, j)$ is satisfied. Then, select an appropriate reference solution $\boldsymbol{f}^R$.

Step 6: From the above steps, make Problem (p. 2) definite and solve it by an appropriate ordinal optimization method.

Step 7: Compare the above result with that of another optimization problem, such as Max(Eq. (3.5)) subject to $\boldsymbol{x} \in$ X.

3.4.2 *Welded Beam Design Problem*

We considered a welded beam design problem (Fig. 3.3) and described it as a four-objective optimization Problem (p. 3). This is originally studied in (Erfani and Utyuzhnikov 2012) as a bi-objective optimization problem.

(p. 3) Min $\{f_1, f_2, f_3, f_4\}$ subject to Eqs. (3.6), (3.7), (3.8), (3.9), (3.10), (3.11), (3.12), (3.13), (3.14), (3.15), (3.16), and (3.17–3.20)

3.4.2.1 Objective Functions

$$f_1 := 1.105h^2 l + 0.048tb\,(L + l) \;\rightarrow\; min(Cost) \tag{3.6}$$

$$f_2 := \delta = \frac{4PL^3}{Et^3 b} \rightarrow \text{min (Deflection)} \tag{3.7}$$

$$f_3 := \tau = \sqrt{\tau'^2 + \tau'\tau''\frac{l}{R} + \tau''^2} \rightarrow \text{min (Shear stress)} \tag{3.8}$$

$$f_4 := \sigma = \frac{6PL}{t^2 b} \rightarrow \text{min (Bending stress)} \tag{3.9}$$

3.4.2.2 Constraints

$$h \le b \tag{3.10}$$

$$Pc \ge P \tag{3.11}$$

$$Pc = 64746.02\,(1 - 0.3t)\,tb^3 \tag{3.12}$$

$$\tau' = \frac{P}{\sqrt{2}hl} \tag{3.13}$$

$$\tau'' = P\,(L + 0.5l)\,R/J \tag{3.14}$$

$$R = \sqrt{0.25\Big(l^2 + (h+t)^2} \tag{3.15}$$

$$J = \sqrt{2}hl\left(\frac{l^2}{12} + \frac{(h+t)^2}{4}\right) \tag{3.16}$$

$$0.125 \le b \le 5, \quad 0.1 \le t \le 10, \quad 0.1 \le l \le 10, \quad 0.125 \le h \le 5, \tag{3.17–3.20}$$

3.4.2.3 Decision Variables

h [m]: welding thickness; l [m]: welding length; t [m]: beam width; b [m]: beam thickness

3.4.2.4 Parameters

$$P = 6000.0 \;\; [\text{lb}]\,, L = 14.0 \;\; [\text{in}]\,, E = 3.0\,\text{E8} \;\; [\text{psi}]$$

3.4.3 Numerical Results

First, we described the objective tree as shown in Fig. 3.4. Then, we generated six random trials within the hyper-rectangular space enclosed by the utopia and nadir, which are shown in Table 3.3 together with the test trials. Next, we set the weights

shimizu@me.tut.ac.jp

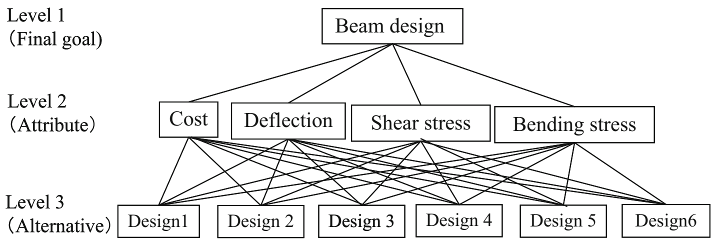

Fig. 3.4 Hierarchy of evaluation factors

Table 3.3 Specification of each trial with utopia and nadir

	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6	Utopia	Nadir
Cost	20.00	9.27	16.49	11.52	13.75	11.13	5.00	20.00
Deflection	4.11 E-03	5.56 E-03	6.66 E-03	6.82 E-03	2.32 E-03	3.80 E-03	1.00 E-03	8.00 E-03
Shear stress	7281.69	8071.22	7333.68	5421.39	12214.13	9246.70	3200.00	13600.00
Bending stress	29256.52	19066.19	18203.62	22550.54	29709.86	27521.33	15000.00	30000.00

Table 3.4 PCM ($t = 1$)

	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
Design 1	1	0.51	0.84	0.61	0.67	0.57
Design 2	1.95	1	1.63	1.18	1.30	1.10
Design 3	1.20	0.61	1	0.72	0.80	0.68
Design 4	1.65	0.85	1.38	1	1.10	0.94
Design 5	1.50	0.77	1.25	0.91	1	0.85
Design 6	1.76	0.91	1.48	1.07	1.18	1

representing the relative importance as $w = (0.4, 0.3, 0.2, 0.1)$, which are the same as those given for the reference value function in Eq. (3.5). Then, the preference intensity of every trial with respect to each objective function was derived from the formula given in Step 2. Finally, the total preference was calculated as $S = (0.293, 0.570, 0.350, 0.484, 0.439, 0.517)$ for $t = 1$. In Step 4, S^i/S^j was calculated to derive the elements of the PCM shown in Table 3.4. Based on that procedure, we built the value function $V_{NN}(\boldsymbol{f}(\boldsymbol{x});\boldsymbol{f}^R)$ of the neural network.

Letting $\boldsymbol{f}^R = \boldsymbol{f}_*$, we solved (p. 3) under this value function by the modified nonlinear simplex method (Nelder and Mead 1965) so that it can accommodate the constraints. In Table 3.5, the result is compared with that obtained by optimizing Problem (p. 3) under the objective function in Eq. (3.5). This problem is solved by using the commercial software package LINGO (Ver. 13.0).

Table 3.5 Results of MUOP (Independent: $t = 1$)

	Decision variable				Objective function value			
	l	t	b	h	Cost	Deflection	Shear stress	Bending stress
This work	2.540	3.328	2.650	1.135	10.614	2.249E-03	11262.08	17179.24
LINGO	2.982	3.329	2.789	1.154	11.955	2.134E-03	9657.292	16307.03
Gap [%]	14.82	0.03	4.98	1.65	11.22	5.39	16.62	5.35

(Input layer: 8 neurons; hidden layer: 10 neurons; learning rate: 0.5; momentum: 0.1; RSME: 3.33×10^{-4})
Gap = | This work − LINGO | /LINGO × 100

Table 3.6 PCM (Independent: $t = 2$)

	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
Design 1	1	0.69	1.00	0.77	0.79	0.76
Design 2	1.45	1	1.45	1.11	1.14	1.11
Design 3	1.00	0.69	1	0.77	0.79	0.77
Design 4	1.31	0.90	1.30	1	1.03	1.00
Design 5	1.27	0.88	1.27	0.98	1	0.97
Design 6	1.31	0.90	1.31	1.00	1.03	1

Table 3.7 Result of MUOP (Independent: $t = 2$)

	Decision variable				Objective function value			
	l	t	b	h	Cost	Deflection	Shear stress	Bending stress
This work	1.857	3.329	2.736	1.149	9.637	2.176E-03	14463.65	16628.19
LINGO	2.118	3.330	3.000	1.102	10.573	1.982E-03	13600.00	15151.22
Gap [%]	12.32	0.03	8.80	4.27	8.85	9.79	6.35	9.75

(Input layer: 8 neurons; hidden layer: 10 neurons; learning rate: 0.5; momentum: 0.1; RSME: 3.60×10^{-4})

In a similar manner, we had $S = (0.408, 0.592, 0.409, 0.534, 0.520, 0.535)$ for $t = 2$ and obtained the results in Tables 3.6 and 3.7. Close correspondence can be observed between the results, with a few exceptions.

3.4.4 Discussion

A definite basis for evaluating subjective decisions in a well-defined manner that is acceptable to everyone does not exist. This fact causes considerable difficulty when attempting to perform a general evaluation to obtain the best-compromise solution found by the mathematical process of MOP. As it often happens, what one person considers the best compromise may not be acceptable to others since each DM has a different value system. We are confident that the procedure outlined here is applicable in such situations since the final preference is evaluated on the basis of

Table 3.8 Comparison of value function values

t	This work	LINGO	Gap [%]
1	0.627	0.633	0.92
2	0.692	0.691	0.18

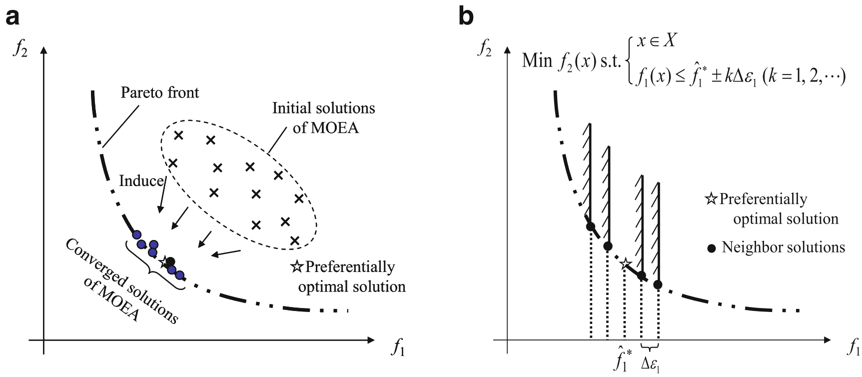

Fig. 3.5 Post-optimal analysis in terms of MUOP. (**a**) Result of elite-induced MOEA. (**b**) Result obtained with ε-constraint method

an implicitly embedded value function, such as Eq. (3.5). This is also a basic norm of utility theory (Fishburn 1970).

Although some results in Tables 3.5 and 3.7 seem to be somewhat far from the reference solution, we can account for this weakness if we compare the results in terms of the above aspects. Both results in Table 3.8 are so similar that the DM cannot distinguish between them. Moreover, we confirmed that the best-compromise solution could not be outperformed by any of 200 solutions obtained with NSGA-II (Deb et al. 2000) after convergence. This numerically validates the proposition in Sect. 3.2, which asserts that the proposed method can derive a Pareto optimal solution.

In addition, we can use the result obtained for the post-optimal analysis combined with a classical multi-objective analysis method, such as the ε constraint method, or recent approaches such as elite-induced evolutionary multi-objective analysis (Shimizu et al. 2012b). As illustrated in Fig. 3.5, by producing several solutions around the optimal result, we can move on to the next stage by choosing among those candidates to make a final decision for actual execution. Based on the above discussion, we again emphasize the validity of the proposed approach.

3.5 Conclusion

A MAOP method that supports flexible and adaptive decision making for complex, diverse and competitive decision environments has been in high demand. From this viewpoint, this study proposed a general idea for solving MAOP problems by extending our previously proposed MUOP methods (MOON^2 and MOON^{2R}).

Although MOON^2 and MOON^{2R} require only simple and relative responses, handling the DM's responses in trade-off analysis becomes rather difficult in MAOP, where more than a few objective functions are to be considered simultaneously. To overcome this difficulty, this study proposed an approach that is easily applicable in such cases. After presenting the general procedure, the effectiveness of the proposed method was verified by applying it to an actual problem. The experimental results showed that the proposed method is moderately more complex than previous methods but maintains flexibility and adaptability. Finally, the general discussion provided a definite and comprehensive outline of the direction of future work in this area.

References

Coello CAC (2001) A short tutorial on evolutionary multiobjective optimization. In: Zitzler E et al (eds) Lecture notes in computer science. Springer, Berlin, pp 21–40

Czyzak P, Jaszkiewicz AJ (1998) Pareto simulated annealing-a meta-heuristic technique for multiple-objective combinatorial optimization. J Multi-Criteria Decis Anal 7:34–47

Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings Parallel Problem Solving from Nature VI (PPSN-VI). Paris, France, pp 849–858

Erfani T, Utyuzhnikov SV (2012) Control of robust design in multiobjective optimization under uncertainties. Struct Multidiscip Optim 45:247–256

Fishburn PC (1970) Utility theory for decision making. Wiley, New York

Hughes EJ (2005) Evolutionary many-objective optimization. Many once or one many? In: Proceedings IEEE Congress on Evolutionary Computation (CEC2005). Edinburgh, UK, pp 222–227

Jaeggi D, Parks G, Kipouros T, Clarkson J (2005) A multiobjective tabu search algorithm for constrained optimization problems. In: Evolutionary multi-criterion optimization, third international conference, EMO 2005, LNCS 3410. Guanajuato, Mexico, pp 490–504

Nelder JA, Mead R (1965) A simplex method for functional minimization. Comput J 7:308–313

Robic T, Filipic B (2005) DEMO: differential evolution for multi-objective optimization. In: Evolutionary multi-criterion optimization, third international conference (EMO 2005), Guanajuato. LNCS 3410, pp 520–533

Saaty TL (1980) The analytic hierarchy process. McGraw-Hill, New York

Sato H, Aguirre H, Tanaka K (2010) Many-objective evolutionary optimization by self-controlling dominance area of solutions. Trans Jpn Soc Evol Comput 1(1):32–41

Shimizu Y (2010) An enhancement of learning optimization engineering – workbench for smart decision making. CORONA Publishing Co., LTD. (In Japanese)

Shimizu Y, Kawada A (2002) Multi-objective optimization in terms of soft computing. Trans Soc Instrum Control Eng 38(11):974–980

Shimizu Y, Nomachi T (2008) Integrated product design through multi-objective optimization incorporated with meta-modeling technique. J Chem Eng Jpn 41(11):1068–1074

Shimizu Y, Tanaka Y (2003) A practical method for multi-objective scheduling through soft computing approach. JSME Int J Ser C 46(1):54–59

Shimizu Y, Tanaka Y, Kawada A (2004) Multi-objective optimization system on the internet. Comput Chem Eng 28(5):821–828

Shimizu Y, Miura K, Yoo J-K, Tanaka Y (2005) A progressive approach for multi-objective design through inter-related modeling of value system and meta-model. J JSME Ser C 71(712):296–303. (In Japanese)

Shimizu Y, Yoo J-K, Tanaka Y (2006) A design support through multi-objective optimization aware of subjectivity of value system. J JSME Ser C 72(717):1613–1620. (In Japanese)
Shimizu Y, Kato Y, Kariyahara T (2010) Prototype development for supporting multiobjective decision making in an ill-posed environment. J Chem Eng Jpn 43(8):691–697
Shimizu Y, Waki T, Sakaguchi T (2012a) Multi-objective sequencing optimization for mixed-model assembly line considering due-date satisfaction. J Adv Mech Des Syst Manuf 6(7):1057–1070
Shimizu Y, Takayama M, Ohishi H (2012b) Multi-objective analysis through elite-induced evolutionary algorithm – in the case of PSA. Trans Jpn Soc Evol Comput 3(2):22–30. (In Japanese)

Yoshiaki Shimizu is a Professor in Department of Mechanical Engineering, Toyohashi University of Technology, Japan. He was responsible for the head of department of Production Systems Engineering during 2006–2009.

He was graduated from Kyoto University in Japan, and earned Doctor of Engineering in 1982. His teaching and research interests include production systems and supply chain management, multi-objective optimization and applied operations research. He is the author of more than 200 academic and technical papers and books. See more detail on his home page (URL http://ise.me.tut.ac.jp/). His email address is shimizu@me.tut.ac.jp