MULTI-OBJECTIVE OPTIMIZATION AND ITS ENGINEERING APPLICATIONS

Hirotaka Nakayama[1]*

Dept. of Information Science and Systems Engineering, Faculty of Science and Engineering, Konan University

Abstract

Many practical optimization problems usually have several conflicting objectives. In those multi-objective optimization, no solution optimizing all objective functions simultaneously exists in general. Instead, Pareto optimal solutions, which are "efficient" in terms of all objective functions, are introduced. In general we have many Pareto optimal solutions. Therefore, we need to decide a final solution among Pareto optimal solutions taking into account the balance among objective functions, which is called "trade-off analysis". It is no exaggeration to say that the most important task in multi-objective optimization is trade-off analysis. Consequently, the methodology should be discussed in view of how it is easy and understandable for trade-off analysis.

In cases with two or three objective functions, the set of Pareto optimal solutions in the objective function space (i.e., Pareto frontier) can be depicted relatively easily. Seeing Pareto frontiers, we can grasp the trade-off relation among objectives totally. Therefore, it would be the best way to depict Pareto frontiers in cases with two or three objectives. (It might be difficult to read the trade-off relation among objectives with three dimension, though). In cases with more than three objectives, however, it is impossible to depict Pareto forntier. Under this circumstance, interactive methods can help us to make local trade-off analysis showing a "certain" Pareto optimal solution. A number of methods differing in which Pareto optimal solution is to be shown, have been developed. This paper discusses critical issues among those methods for multi-objective optimization, in particular applied to engineering design problems.

Keywords: Multi-Objective Optimization, Interactive Multi-Objective Optimization, Evolutionary Algorithms, Pareto Frontier

1. Introduction

Multi-objective programming problems are formulated as follows:

$$\text{(MOP)} \qquad \text{Minimize} \quad f(x) \equiv (f_1(x),\ f_2(x), \dots,\ f_r(x))$$
$$\text{over} \quad x \in X.$$

The constraint set X may be given by

$$g_j(x) \leq 0, \qquad j = 1, \dots, m,$$

and/or a subset of R^n itself. For the problem (MOP), we define Pareto solutions as follows:

Definition 1.1 A solution $\hat{x}$ is said *Pareto optimal*, if there is no better solution $x \in X$ other than $\hat{x}$, namely, if

$$f(x) \not\leq f(\hat{x}) \qquad \text{for any} \quad x \neq \hat{x} \in X.$$

In general, there may be many Pareto solutions. The final decision is made among them taking the total balance

[1] 8-9-1 Okamoto, Higashinada, Kobe, Hyogo 658-8501, Japan, nakayama@konan-u.ac.jp

Dagstuhl Seminar Proceedings 04461
Practical Approaches to Multi-Objective Optimization
http://drops.dagstuhl.de/opus/volltexte/2005/234

over all criteria into account. This is a problem of value judgment of decision maker (in abbreviation, DM). The totally balancing over criteria is usually called *trade-off*. It should be noted that there are very many criteria, say, over one hundred in some practical problems such as erection management of cable stayed bridge, and camera lens design. Therefore, it is very important to develop effective methods for helping DM to trade-off easily even in problems with very many criteria.

Interactive multi-objective programming search a solution in an interactive way with DM while eliciting information on his/her value judgment. Along this line, several methods were developed remarkably in 1980's: Among them, the aspiration level approach is now recognized very effective in practice, because

(i) it does not require any consistency of DM's judgment,

(ii) aspiration levels reflect the wish of DM very well,

and

(iii) aspiration levels play the role of probe better than the weight for objective functions.

In the following, we will discuss the difficulty in weighting method which is commonly used in the traditional goal programming.

2. Why is the Weighting Method Ineffective?

In multi-objective programming problems, the final decision is made on the basis of the value judgment of DM. Hence it is important how we elicit the value judgment of DM. In many practical cases, the vector objective function is scalarized in such a manner that the value judgment of DM can be incorporated.

The most well known scalarization technique is the linearly weighted sum:

$$\sum_{i=1}^{r} w_i f_i(x). \tag{2.1}$$

The value judgment of DM is reflected by the weight. Although this type of scalarization is widely used in many practical problems, there is a serious drawback in it. Namely, it can not provide a solution among sunken parts of Pareto surface due to "*duality gap*" for nonconvex cases. Even for convex cases, for example, in linear cases, even if we want to get a point in the middle of line segment between two vertices, we merely get a vertex of Pareto surface, as long as the well known simplex method is used. This implies that depending on the structure of problem, the linearly weighted sum can not necessarily provide a solution as DM desires.

In the traditional goal programming (Charnes-Cooper, 1961), some kind of metric function from the goal f^* is used as the one representing the preference of DM. For example, the following is well known:

$$\left(\sum_{i=1}^{r} w_i |f_i(x) - f_i^*|^p\right)^{1/p} \tag{2.2}$$

The preference of DM is reflected by the weight w_i, the value of p, and the value of the goal f_i^*. If the value of p is chosen appropriately, a Pareto solution among a sunken part of Pareto surface can be obtained by minimizing the function (2.2). However, it is usually difficult to pre-determine appropriate values of them. Moreover, the solution minimizing (2.2) can not be better than the goal $\boldsymbol{f}^*$, even though the goal is underestimated.

In addition, one of the most serious drawbacks in the goal programming is that people tend to misunderstand that a desirable solution can be obtained by adjusting the weight. It should be noted that there is no positive correlation between the weight w_i and the value $f_i(\hat{x})$ corresponding to the resulting solution $\hat{x}$ as will be seen in the following example.

Example 2.1 Let $y_1 = f_1(x),\ y_2 = f_2(x)$ and $y_3 = f_3(x)$, and let the feasible region in the objective space be given by

$$\{(y_1, y_2, y_3)|\ \ (y_1 - 1))^2 + (y_2 - 1)^2 + (y_3 - 1)^2 \leq 1\}.$$

Suppose that the goal is $(y_1^*,\ y_2^*,\ y_3^*) = (0,\ 0,\ 0)$.The solution minimizing the metric function (2.2) with $p = 1$ and $w_1 = w_2 = w_3 = 1$ is $(y_1,\ y_2,\ y_3) = (1-1/\sqrt{3},\ 1-1/\sqrt{3},\ 1-1/\sqrt{3})$. Now suppose that DM wants to decrease the value of f_1 a lot more and that of f_2 a little more, and hence modify the weight into $w_1' = 10,\ w_2' = 2,\ w_3' = 1$. The solution associated with the new weight is $(1-10/\sqrt{105},\ 1-2/\sqrt{105},\ 1-1/\sqrt{105})$. Note that the value of f_2 is worse than before despite that DM wants to improve it and hence increased the weight of f_2 up to twice. Someone might think that this is due to no normalization of weight. Therefore, we normalize the weight by $w_1 + w_2 + w_3 = 1$. The original weight normalized in this way is $w_1 = w_2 = w_3 = 1/3$ and the renewed weight by the same normalization is $w_1' = 10/13,\ w_2' = 2/13,\ w_3' = 1/13$. We can observe that w_2' is less than w_2. Now increase the normalized weight w_2 to be greater than 1/3. To this end, set the unnormalized weight $w_1 = 10,\ w_2 = 7$ and $w_3 = 1$. With this new weight, we have a solution $(1-10/\sqrt{150},\ 1-7/\sqrt{150},\ 1-1/\sqrt{150})$. Despite that the normalized weight $w_2'' = 7/18$ is greater than the original one $(= 1/3)$, the obtained solution is still worse than the previous one.

As is readily seen in the above example, it is usually very difficult to adjust the weight in order to obtain a solution as DM wants. Therefore, it seems much better to take the aspiration level of DM rather than the weight as the probe. Interactive multi-objective programming techniques based on aspiration levels have been developed so that the drawbacks of the traditional goal programming may be overcome. In the following section, we shall discuss the satisficing trade-off method developed by the author (Nakayama 1984) as one of them.

3. Satisficing Trade-off Method

In the aspiration level approach, the aspiration level at the k-th iteration $\overline{f}^k$ is modified as follows:

$$\overline{f}^{k+1} = T \circ P(\overline{f}^k) \tag{3.1}$$

Here, the operator P selects the Pareto solution nearest in some sense to the given aspiration level $\overline{f}^k$. The operator T is the trade-off operator which changes the k-th aspiration level $\overline{f}^k$ if DM does not compromise with the shown solution $P(\overline{f}^k)$. Of course, since $P(\overline{f}^k)$ is a Pareto solution, there exists no feasible solution which makes all criteria better than $P(\overline{f}^k)$, and thus DM has to trade-off among criteria if he wants to improve some of criteria. Based on this trade-off, a new aspiration level is decided as $T \circ P(\overline{f}^k)$. Similar process is continued until DM obtains an agreeable solution. This idea is implemented in DIDASS (Grauer *et al*. 1984) and the satisficing trade-off method (Nakayama 1984). While DIDASS leaves the trade-off to the heuristics of DM, the satisficing trade-off method provides a device based on the sensitivity analysis.

3.1. On The Opearation P

The operation which gives a Pareto solution $P(\overline{f}^k)$ nearest to $\overline{f}^k$ is performed by some auxiliary scalar optimization. It has been shown in Sawaragi-Nakayama-Tanino (1985) that the only one scalarization technique, which provides any Pareto solution regardless of the structure of problem, is of the Tchebyshev norm type. However, the scalarization function of Tchebyshev norm type yields not only a Pareto solution but also a weak Pareto solution. Since weak Pareto solutions have a possibility that there may be another solution which improves a critera while others being fixed, they are not necessarily "*efficient*" as a solution in decision making. In order to exclude weak Pareto solutions, the following scalarization function of the augmented Tchebyshev type can be used:

$$\max_{1 \le i \le r} w_i(f_i(x) - \overline{f}_i) + \alpha \sum_{i=1}^{r} w_i f_i(x). \tag{3.2}$$

where α is usually set a sufficiently small positive number, say 10^{-6}.

Theorem 3.1 (Nakayama-Tanino 1994) For arbitrary $w \ge 0$ and $\alpha > 0$, $\hat{x} \in X$ minimizing (3.2) is a properly Pareto optimal solution to (MOP). Conversely, if $\hat{x}$ is a properly Pareto optimal solution to (MOP), then there exist $w > 0$, $\alpha > 0$ and $\overline{f}$ such that $\hat{x}$ minimizes (3.2) over X.

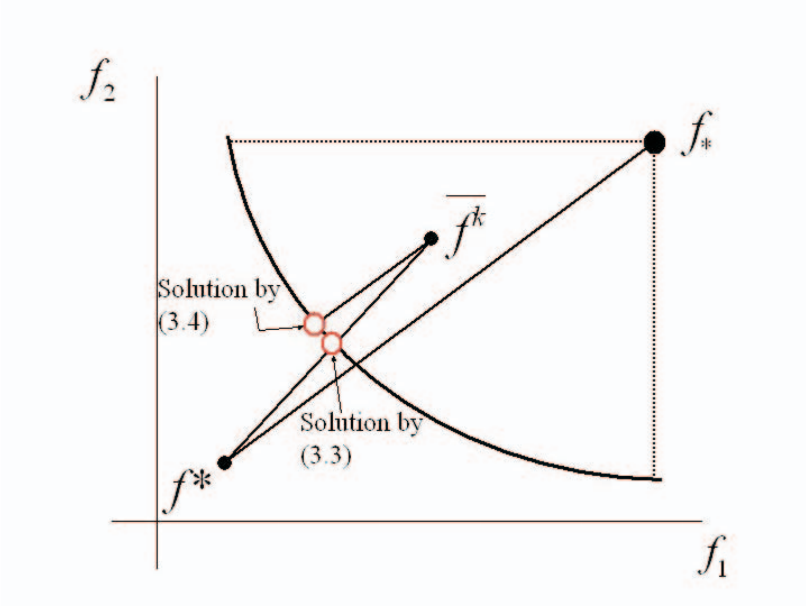

Fig. 1 Difference between Pareto Solutions by (3.3) and(3.4)

The weight w_i is usually given as follows: Let f_i^* be an ideal value which is usually given in such a way that $f_i^* < \mathrm{Min}\{f_i(x)|\ x \in X\}$. For this circumstance, we set

$$w_i^k = \frac{1}{\overline{f}_i^k - f_i^*} \tag{3.3}$$

If the weight is preferable to be unchanged for the change of aspiration level, it can be given by

$$w_i^{k'} = \frac{1}{f_{*i} - f_i^*}. \tag{3.4}$$

Here f_{*i} be a nadir value which is usually defined as

$$f_{*i} = \max_{1 \leq j \leq r} f_i(x_j^*) \tag{3.5}$$

where

$$x_j^* = arg \min_{x \in X} f_j(x). \tag{3.6}$$

The minimization of (3.2) with (3.3) or (3.4) is usually performed by solving the following equivalent optimization problem, because the original one is not smooth:

(Q) Minimize $z + \alpha \sum_{i=1}^{r} w_i f_i(x)$

subject to

$$w_i^k(f_i(x) - \overline{f}_i^k) \ \leq \ z \tag{3.7}$$

$$x \in X.$$

Remark 3.1 The difference between solutions to (Q) for two kinds of weights (3.3) and (3.4) is illustrated in Fig. 1. In the auxiliary min-max problem (Q) with the weight by (3.3), $\overline{f}_i^k$ in the constraint (3.7) may be replaced with f_i^* without any change in the solution. For we have

$$\frac{f_i(x) - f_i^*}{\overline{f}_i^k - f_i^*} = \frac{f_i(x) - \overline{f}_i^k}{\overline{f}_i^k - f_i^*} + 1.$$

3.2. On The Operation T

In cases that DM is not satisfied with the solution for $P(\overline{\boldsymbol{f}}^k)$, he/she is requested to answer his/her new aspiration level $\overline{\boldsymbol{f}}^{k+1}$. Let x^k denote the Pareto solution obtained by projection $P(\overline{\boldsymbol{f}}^k)$, and classify the objective functions into the following three groups:

(i) the class of criteria which are to be improved more,
(ii) the class of criteria which may be relaxed,
(iii) the class of criteria which are acceptable as they are.

Let the index set of each class be denoted by I_I^k, I_R^k, I_A^k, respectively. Clearly, $\overline{f}_i^{k+1} < f_i(x^k)$ for all $i \in I_I^k$. Usually, for $i \in I_A^k$, we set $\overline{f}_i^{k+1} = f_i(x^k)$. For $i \in I_R^k$, DM has to agree to increase the value of $\overline{f}_i^{k+1}$. It should be noted that an appropriate sacrifice of f_j for $j \in I_R^k$ is needed for attaining the improvement of f_i for $i \in I_I^k$.

Example 3.1 Consider the same problem as in Example 2.1: Let $y_1 = f_1(x),\ y_2 = f_2(x)$ and $y_3 = f_3(x)$, and let the feasible region in the objective space be given by

$$\{(y_1, y_2, y_3)|\ \ (y_1 - 1))^2 + (y_2 - 1)^2 + (y_3 - 1)^2 \leq 1\}.$$

Suppose that the ideal point is $(y_1^*,\ y_2^*,\ y_3^*) = (0, 0, 0)$, and the nadir point is $(y_{*1},\ y_{*2},\ y_{*3}) = (1, 1, 1)$. Therefore, using (3.6) we have $w_1 = w_2 = w_3 = 1.0$. Let the first aspiration level be $(\overline{y}_1^1,\ \overline{y}_2^1,\ \overline{y}_3^1) = (0.4,\ 0.4,\ 0.4)$. Then the solution to (Q) is $(y_1^1,\ y_2^1,\ y_3^1) = (0.423,\ 0.423,\ 0.423)$. Now suppose that DM wants to decrease the value of f_1 a lot more and that of f_2 a little more, and hence modify the aspiration level into $\overline{y}_1^2 = 0.35$ and $\overline{y}_2^2 = 0.4$. Since the present solution $(y_1^1,\ y_2^1,\ y_3^1) = (0.423,\ 0.423,\ 0.423)$ is already Pareto optimal, it is impossible to improve all of criteria. Therefore, suppose that DM agrees to relax f_3, and with its new aspiration level of $\overline{y}_3^2 = 0.5$. With this new aspiration level, the solution to (Q) is $(y_1^2,\ y_2^2,\ y_3^2) = (0.366,\ 0.416,\ 0.516)$. Although the obtained solution does not attain the aspiration level of f_1 and f_2 a little bit, it should be noted that the solution is improved more than the previous one. The reason why the improvement of f_1 and f_2 does not attain the wish of DM is that the amount of relaxation of f_3 is not much enough to compensate for the improvement of f_1 and f_2. In the satisficing trade-off method, DM can find a satisfactory solution easily by making the trade-off analysis deliberately. To this end, methods for automatic trade-off or exact trade-off are devised using the sensitivity analysis in mathematical programming. See [23], [19], [24] in more detail.

3.3. Interchange between Objectives and Constraints

In the formulation of the auxiliary scalarized optimization prolem (Q), change the right hand side of the equation (3.7) into $\beta_i z$, namely

$$w_i(f_i(x) - \overline{f}_i)\ \leq\ \beta_i z. \tag{3.9}$$

As is readily seen, if $\beta_i = 1$, then the function f_i is considered to be an objective function, for which the aspiration level $\overline{f}_i$ is not necessarily attained, but the level of f_i should be better as much as possible. On the other hand, if $\beta_i = 0$, then f_i is considered to be a constraint function, for which the aspiration level $\overline{f}_i$ should be guaranteed. In many practical problems, there is almost no cases in which we consider the role of objective and constraint fixed from the beginning, but usually we want to interchange them depending on the situation. Using the formula (3.9), this can be done very easily (Korhonen 1987). In addition, if the value of β_i is set in the middle of 0 and 1, f_i can play a role in the middle of objective and constraint which is neither a complete objective nor a complete constraint (Kamenoi *et al*. 1992). This is also very effective in many practical problems.

4. Applications

Interactive multi-objective programming methods have been applied to a wide range of practical problems. Good examples in engineering applications can be seen in Eschenauer *et al*. (1990). The satisficing trade-off method also has been applied to several real problems:

1) blending

   a) feed formulation for live stock (Mitani *et al*. 1997)

   b) plastic materials (Nakayama *et al*. 1986)

   c) cement production (Nakayama 1991)

   d) portfolio (Nakayama 1989)

2) design

   a) camera lens

   b) erection management of cable-stayed bridge (Furukawa *et al*. 1986; Nakayama *et al.* 1995)

3) planning

   a) scheduling of string selection in steel manufacturing (Ueno *et al*. 1990)

   b) long term planning of atomic power plants

In the following, an application of satisficing trade-off method to erection management of cable-stayed bridges will be explained briefly. In erection of cable stayed bridge, the following criteria are considered for accuracy control (Furukawa *et al*. 1986):

i. residual error in each cable tension,
ii. residual error in camber at each node,
iii. amount of shim adjustment for each cable,
iv. number of cables to be adjusted.

Since the change of cable rigidity is small enough to be neglected with respect to shim adjustment, both the residual error in each cable tension and that in each camber are linear functions of amount of shim adjustment. Let us define n as the number of cable in use, $\Delta T_i \quad (i = 1, \dots, n)$ as the difference between the designed tension values and the measured ones, and x_{ik} as the tension change of i-th cable caused from the change of the k-th cable length by a unit. The residual error in cable tension caused by the shim adjustment is given by

$$p_i = |\Delta T_i - \sum_{k=1}^{n} x_{ik} \Delta l_k| \qquad (i = 1, \dots, n)$$

Let m be the number of nodes, $\Delta z_j \quad (j = 1, \dots, m)$ the difference between the designed camber values and the measured ones, and y_{jk} the camber change at j-th node caused from the change of the k-th cable length by a unit. Then the residual error in the camber caused by the shim adjustments of $\Delta l_1, \dots, \Delta l_n$ is given by

$$q_j = |\Delta Z_j - \sum_{k=1}^{n} y_{jk} \Delta l_k| \qquad (j = 1, \dots, m)$$

In addition, the amount of shim adjustment can be treated as objective functions of

$$r_i = |\Delta l_i| \qquad (i = 1, \dots, n)$$

And the upper and lower bounds of shim adjustment inherent in the structure of the cable anchorage are as follows;

$$\Delta l_{Li} \leq \Delta l_i \leq \Delta l_{Ui} \qquad (i = 1, \dots, n).$$

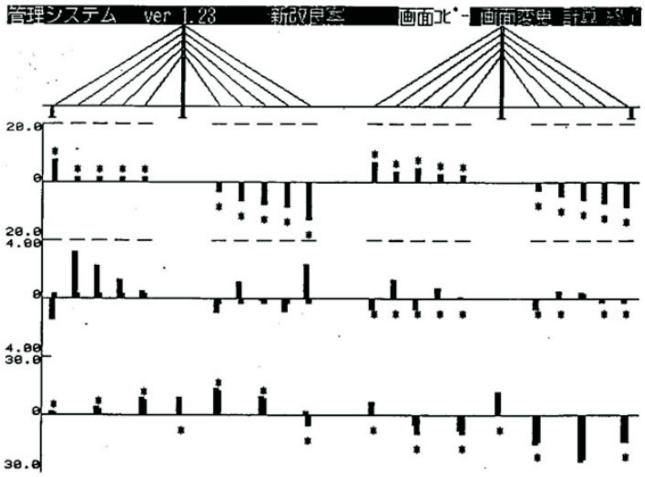

Fig. 2 Erection Management System of Cable-stayed Bridge

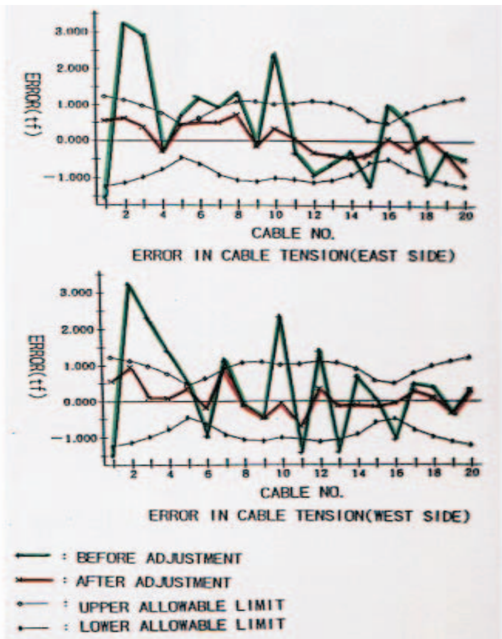

Fig. 3 Result by Erection Management System

Fig. 2 show a monitor output of erection management system of cable stayed bridge using the satisficing trade-off method and Fig.3 does its result. The residual error of each criterion and the amount of shim adjustment are represented by graphs. The aspiration level is inputted by a mouse on the graph. After solving the auxiliary min-max problem, the Pareto solution according to the aspiration level is represented by a graph in a similar fashion. This procedure is continued until the designer can obtain a desirable shim-adjustment. This operation is very easy for the designer, and the visual information on trade-off among criteria is user-friendly. The software was used for real bridge construction, say, Swan Bridge (Ube City) and Karasuo Harp Bridge (Kita-Kyusyu City) in 1992, and so on.

5. Generating Pareto Frontiers

In cases with two or three objective functions, if it does not take so much time to evaluate each objective function, it is most effective to depict Pareto frontiers which lead DM to grasp the trade-off relation among objectives totally. Since we can not so easily read trade-off relation for 3 dimensional Pareto frontiers without rotation, it would be most effective to depict Pareto frontiers in cases with two objectives. The constraint transformation method (ϵ constrained method in some references) can be applied to this end. The method is seen at the dawn of development of multi-objective optimization (Edgeworth [6]). Since a rough but acceptable approximation of Pareto frontiers can be obtained usually at 10-20 sample values of right hand side of objective function transformed into a constraint, the method fits our purpose well, if each optimization is not time consuming. However, if those auxiliary optimization problems are difficult to solve by usual optimization tools (e.g., if the problems are highly nonlinear with multi-modal, combinatorial, nonsmooth and so on), it becomes difficult to depict Pareto frontier by the constraint transformation method.

5.1. Evolutionary Methods and Other Approaches

In recent years, the research applying evolutionary algorithms to give Pareto frontiers has been extensively developed. It has been observed that the performance of evolutionary algorithms is outstanding in particular for optimization of multi-modal, discrete and nonsmooth objective functions. Although it is possible to apply evolutionary algorithms for optimization by the constraint transformation method, main researches using evolutionary algorithms aim to give Pareto frontiers directly. The idea is to move individuals towards Pareto frontier through evolution. In this apporach, the important things are how fast individuals converge to Pareto frontier and how well spread they are on the whole Pareto frontier. To this end, many researchers have reported many devices for evolutionary operators and fitness function (see for example [26], [8], [10], etc). Tutorial books are [5], [4] and so on.

One of most widely applied evolutionary methods is the ranking method proposed by [8]. Although there have been developed several methods for evaluating the diversity of individuals on Pareto frontier, its essential idea is in the evaluating way how far each individual is from Pareto frontier by the number of dominating individuals. However, the rank does not relect the “distance” itself between each individual and Pareto frontier. Arakawa *et al.* proposed to apply DEA (Data Envelopment Analysis) to generate Pareto frontier [1]. DEA was originally developed to measure the efficiency of decision making units by Charnes *et al.* [3]. Its idea is to measure the “distance” between each decision unit and Pareto frontier by solving some linear programming problem. The real Pareto frontier is approximated by a part of the convex hull of decision units. It has been observed through several applications that DEA provides Pareto frontier with relatively less number of individuals. This means that Pareto frontier can be obtained with less number of experiments (analyses) in engineering design problems. However, since DEA is based on the convex hull of decision units, it can not provide nonconvex Pareto frontier as it is. Yun *et al.* extended DEA to GDEA (Generalized Data Envelopment Analysis) so that it may be valid to non-convex cases [30], and applied GDEA to generate Pareto frontier [31]. It has been observed through our experiences that GDEA provides well distributed Pareto frontier with less number of experiments (analyses). Furthermore, Yun *et al.* try to apply some computational intelligence techniques such as SVM (Support Vector Machine) [33] which was originally developed for pattern classification in machine learning [29]. The essential idea of machine learning such as SVM is to approximate the discriminant boundary for classification. In particular, ν-SVM [27] developed recently can be effectively applied to problems with one category. Utilizing this property for one category problems, not only Pareto frontier but also the feasible region itself can be approximated. It has been observed that SVM provides effectively Pareto frontier in some cases, but not in general. Further research should be made along this line.

5.2. Comparison Through Simulations

In this subsection, we shall compare the stated methods through several computational simulations.

i) *Cantilever Beam Problem*

Consider a cantilever design problem with two design variables, that is, diameter (d) and length (l) as shown in Fig. 4 which are cited from [5]. The beam has to carry an end load P. The cantilever design problem has two conflicting objectives of design: minimization of weight f_1 and minimization of end deflection f_2, and two constraints:

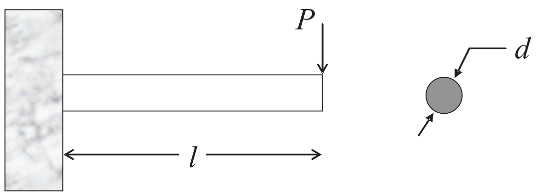

Fig. 4 A Schematic of Cantilever Beam Design

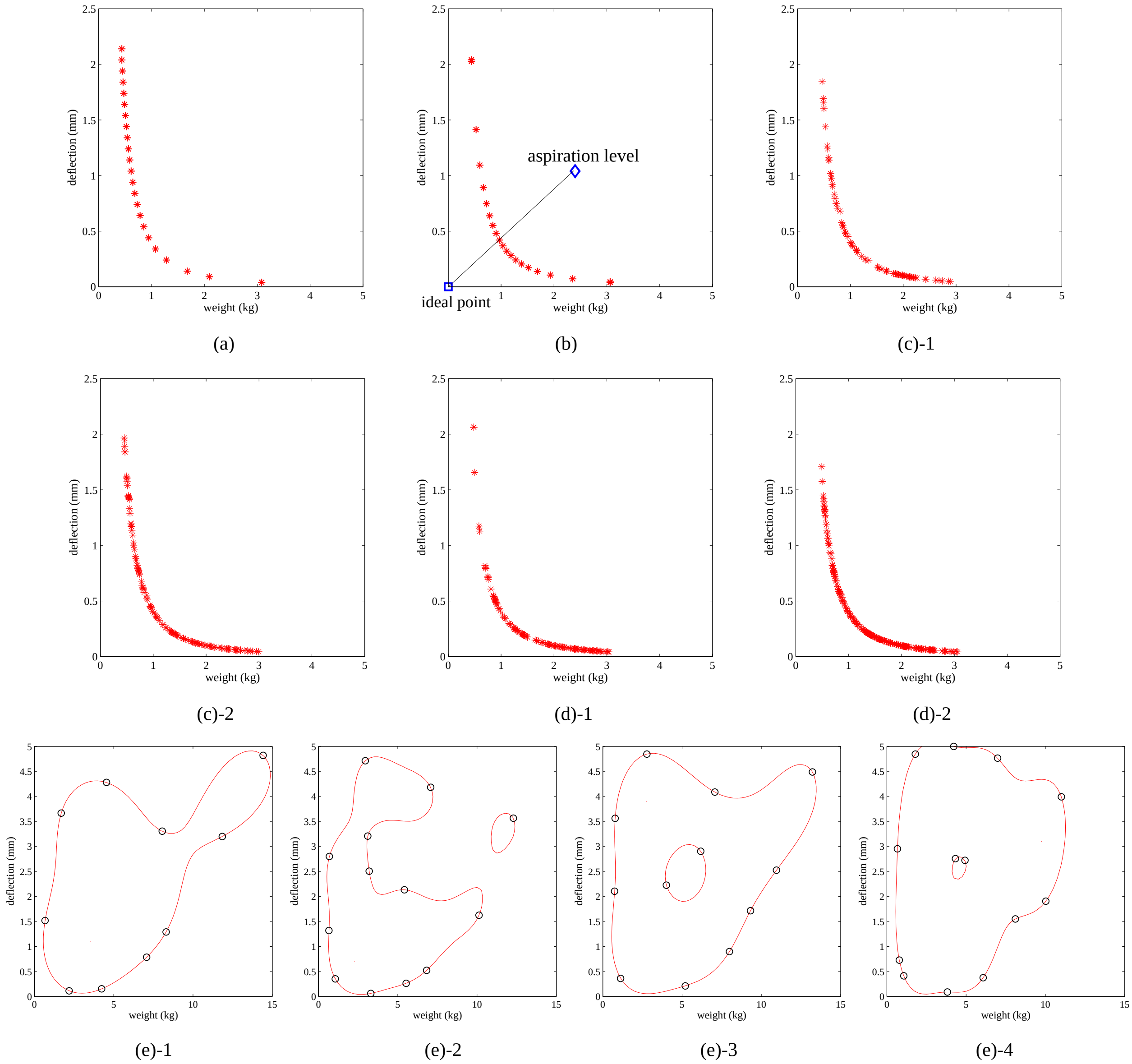

Fig. 5 Comparison of the Results by :
(a) $\varepsilon-$constraint method (b) satisficing trade-off method (c) MOGA (d) GDEA (e) SVM

the developed maximum stress $\sigma_{\max}$ is less than the allowable strength S_y and the end deflection δ is smaller than a

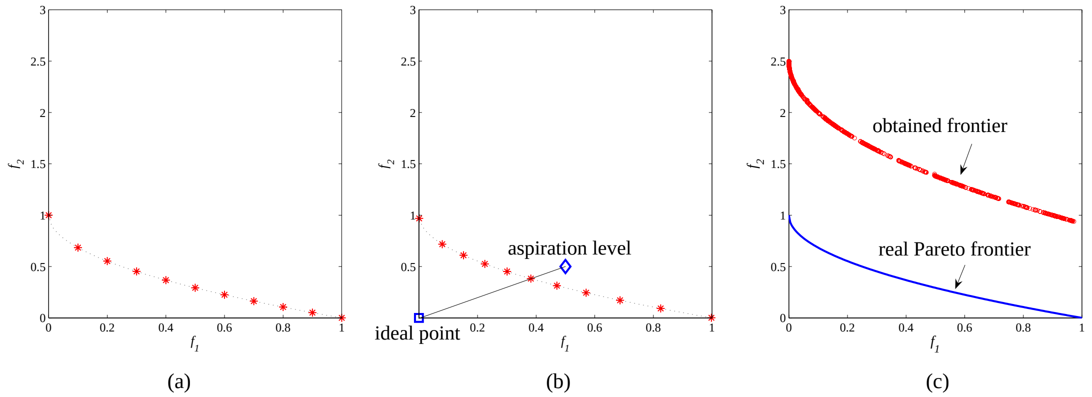

Fig. 6 Comparison of the Results to the problem ZDT4 by :
(a) $\varepsilon-$constraint method (b) satisficing trade-off method (c) GDEA

specified limit $\delta_{\max}$. Now, the optimization problem is formulated as follows:

$$\begin{aligned} &\text{minimize} && f_1(d,l) := \rho\frac{\pi d^2}{4}l \\ &\text{minimize} && f_2(d,l) := \delta = \frac{64Pl^3}{3E\pi d^4} \\ &\text{subject to} && \sigma_{\max} \leqq S_y, \\ &&& \delta \leqq \delta_{\max}, \\ &&& 10 \leqq d \leqq 50, \quad 200 \leqq l \leqq 1000, \end{aligned}$$

where the maximum stress is calculated as follows:

$$\sigma_{\max} = \frac{32Pl}{\pi d^3}.$$

The parameter values are used as follows:

$$\begin{aligned} &\rho = 7800 \text{ kg/m}^3, \ P = 1 \text{ kN}, \quad E = 207 \text{ GPa}, \\ &S_y = 300 \text{ MPa}, \quad \delta_{\max} = 5 \text{ mm}. \end{aligned}$$

ii) *ZDT4 Problem*

As an example which is not so easy to solve by MOGA, ZDT4 is suggested by Zitzler, Deb and Thiele [34]:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} && f_1(\boldsymbol{x}) = x_1 \\ &&& f_2(\boldsymbol{x}) = g(\boldsymbol{x}) \times \left(1 - \sqrt{\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}}\right) \\ &\text{subject to} && g(\boldsymbol{x}) = 1 + 10(N-1) + \sum_{i=2}^{N}\left(x_i^2 - 10\cos(4\pi x_i)\right), \\ &&& x_1 \in [0,1], \ x_i \in [-5,5], \ i = 1,2,\ldots,N \ (N=10). \end{aligned} \tag{ZDT4}$$

There are 10 design variables and two objective functions. Pareto optimal values to the problem (ZDT 4) is composed of $g(\boldsymbol{x}) = 1$.

Fig. 5, Fig. 6 and Table 1 show results for our test problems. The calculation was performed by MATLAB ver 6.5.

In the cantilever beam design problem, almost similar results were obtained regardless methods. ZDT4 is well known as the one difficult to solve by evolutionary algorithms. In applying GDEA, we used a simple GA in the internal process. Therefore, both GDEA and MOGA using simple GA could not provide the exact Pareto frontier. In this case,

Table 1 Comparison of the number of function call

1) Beam Deasign Problem

$\varepsilon-$constraint Method	satisficing trade-off method	MOGA & GDEA	SVM
951 **Fig. 5** (a) *cf.* # ε : 23 cases	52 per one aspiration level **Fig. 5** (b) *cf.* # on the average	1000 (=100 data×10 generation) **Fig. 5** (c)-1, (d)-1	250 (=50 data×5 generation) **Fig. 5** (e)-1
			500 (=50 data×10 generation) **Fig. 5** (e)-2
		1500 (=100 data×15 generation) **Fig. 5** (c)-2, (d)-2	1000 (=100 data×10 generation) **Fig. 5** (e)-3
			1500 (=100 data×15 generation) **Fig. 5** (e)-4

2) ZDT4 Problem

$\varepsilon-$constraint Method	satisficing trade-off method	GDEA	SVM
613 **Fig. 6** (a) *cf.* # ε : 11 cases	264 per one aspiration level **Fig. 6** (b) *cf.* # on the average	25000 (=100 data×250 generation) **Fig. 6** (c)	—

therefore, more sophisticated evolutionary algorithms should be applied. For this problem, SVM could not yield a reasonable result within the number of function calls at the same order.

It should be noted that classical methods such as constraint transformation method (ϵ constraint method) can provide Pareto frontier with less number of function calls in cases that traditional optimization techniques can be applied. It is not sure how the nonlinear optimization tool of MATLAB treats the problem. Using another software on the basis of SQP with numerical differentiation developed by one of author's collaborators, it takes about 400 function calls per one aspiration level. Therefore, if the derivatives of functions are available, the number of function calls can reduce up to 1/10 (because the number of variables are 10 in ZDT4).

6. Concluding Remarks

One of the most impotant aims of multi-objecitve optimization is to assist decision makers to make a satisfactory decision taking into account the balance among several conflicting objectives. Althouhg it would be the best to depict the whole Pareto frontiers so that we can grasp the trade-off relation among objectives totally, it is not so easy in cases with three objectives and impossible in cases with more than three objectives. Therefore, this approach is most effective for cases with two objectives. In recent years, evolutionary methods have been remarkably developed for this purpose. However, those methods needs usually very many individuals, in other words, very many function calls. In engineering design problems in general, it is time consuming and expensive to evaluate functions, because they are calculated through several kinds of analyses such as structural analysis, fluid mechanical analysis, thermo-dynamical analysis and so on, and sometimes even by making real samples. In order to overcome this difficulty, parallelization in computation of evolution is a device. Another device is to evaluate the fitness for only some part of individuals and to approximate that of others by using some computational intelligence [13].

However, it is important to recall those approaches are effective for cases with only two or three objectives. Under this circumstance, we have seen in this paper that the classical constraint transformation method (ϵ constraint method) can be applied with less number of function call. I believe that we must not to miss the essense of multi-objective optimization, and it is sometimes important to go back to classics. There have been a trial combining aspiration level aproach and generating Pareto frontier in cases with more than three objectives [32].

One of outstanding differences between multi-objective optimization and natural sciences is the fact that multi-objective optimization problems include the value judgement of human beings. In many cases, the value judgment of decision makers is not consistent during the decision process. It is very important to get a solution reflecting faithfully the value-judgment of decisin makers even though it may be inconsistent. To this end, it would be almost impossible to obtain a solution by computer only, but cooperative systems of man and computers are inevitable in many engineering design problems. Under this event, it is important to make use of strong points of man and computer respectively. Taking this into account, satisficing from the human side and optimization from the computer side might be practical and effective in particular in multi-objective optimization problems in real life.

Acknowledgment

The author would like to his sincere gratitude to Dr. Yun of Kagawa University and Dr. Miyashita of Ibaraki University for their help in computer simulations, and to Prof. Sugmimoto of Hokkai Gakuen University, Prof. Furukawa of Yamaguchi University and Prof. Arakawa of Kagawa University for their help in applications to structural optimization and valuable discussions. Some results in this paper owe to discussion in the research project "Multi-objective Optimization and Satisficing Design" of the Japan Society for Computational Engineering and Science.

References

1. Arakawa, M., Nakayama, H., Hagiwara, I. and Yamakawa, H. (1998), Multiobjective Optimization Using Adaptive Range Genetic Algorithms with Data Envelopment Analysis, *A Collection of Technical Papers on 7th Symposium on Multidisciplinary Analysis and Optimization (TP98-4970), AIAA*, 3, 2074–2082.
2. Charnes, A. and Cooper, W. (1961), *Management Models and Industrial Applications of Linear Programming, Vol. 1*, John Wiley, New York.
3. Charnes, A., Cooper, W.W. and Rhodes, E. (1978), Measuring the Efficiency of Decision Making Units, *European Journal of Operational Research*, 2, 429-444.
4. Coello Coello, C.A., Van Veldhuizen, D.A. and Lamont, G.B. (2002), *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers.
5. Deb, K. (2001), *Multi-Objective Optimization using Evolutionary Algorithms*, John & Wiley Sons, Ltd..
6. Edgeworth, F.Y., (1881), *Mthematical Psychics*, Kegan Paul & Co.
7. Eschenauer, H.A., Koski, J. and Osyczka, A. (1990), *Multicriteria Design Optimization*, Springer
8. Fonseca, C.M. and Fleming, P. J. (1995), An Overview of Evolutionary Algorithms in Multi-objective Optimization, *Evolutionary Computation Journal*, 3, 1-16.
9. Furukawa, K., Inoue, K. Nakayama, H. and Ishido, K. (1986), Management of Erection for Cable Stayed Bridge using Satisficing Trade-off Method, *Journal of the Japan Soc. of Civil Engineers*, 374/I-6, 495-502. (in Japanese)
10. Goldberg, D.E. (1989), *Genetic Algorithms for Search, optimization, and Machine Learning*, Addison-Wesley
11. Grauer, M., Lewandowski, A. and Wierzbicki A.P. (1984), DIDASS Theory, Implementation and Experiences, in M. Grauer and A.P. Wierzbicki (eds.) *Interactive Decision Analysis, Proceedings of an International Workshop on Interactive Decision Analysis and Interpretative Computer Intelligence*, Springer: 22-30.
12. Kamenoi, T., Sugimoto, H. and Nakayama, H. (1992), Studies on Revised Satisficing Trade-off Method for Structure Design, *Journal of the Japan Soc. of Civil Engineers*, 441/I-18, 117-126 (in Japanese)
13. Karakasis, M. K. and Giannakoglou, K. C. (2004), On the Use of Surrogate Evaluation Models in Multi-objective Evolutionary Algorithms, *Proc. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)* in CD-ROM.
14. Korhonen, P. (1987): VIG-A Visual Interactive Support System for Multiple Criteria Decision Making, *Belgian J. Oper. Res. Stat. Comp. Sci.* 27, 3-15
15. Mitani, K. and Nakayama, H. (1997): A Multi-objective Diet Planning Support System using the Satisficing Trade-off Method, *J. Multi-criteria Decision Analysis*, 6, 131-139.
16. Nakayama, H. (1984), Proposal of Satisficing Trade-off Method for Multi- objective Programming, *Transact. SICE*, 20: 29-35 (in Japanese)
17. Nakayama, H. (1989), An Interactive Support System for Bond Trading, in A. G. Lockett and G. Islei (eds.) *Improving Decision Making in Organizations*, Springer, 325-333
18. Nakayama, H. (1991), Satisficing Trade-off Method for Problems with Multiple Linear Fractional Objectives and its Applications, in A. Lewandowski and V. Volkovich (eds.) *Multiobjective Problems of Mahtematical Programming*, Springer, 42-50.
19. Nakayama, H. (1992), Trade-off Analysis using Parametric Optimization Techniques, *European J. of Operational Research*, 60, 87-98
20. Nakayama, H. (1995), Aspiration Level Approach to Interactive Multi-objective Programming and its Applications, P.M. Pardalos, Y. Siskos and C. Zopounidis (eds.), *Advances in Multicriteria Analysis*, Kluwer Academic Publishers, 147-174.
21. Nakayama, H., Nomura, J., Sawada, K. and Nakajima, R. (1986). An Application of Satisficing Trade-off Method to a Blending Problem of Industrial Materials, in G. Fandel et al. (Eds.), *Large Scale Modelling and Interactive Decision Analysis*. Springer, 303-313.

22. Nakayama, H. and Sawaragi, Y. (1984). Satisficing Trade-off Method for Interactive Multiobjective Programming Methods, in M. Grauer and A.P. Wierzbicki (eds.) *Interactive Decision Analysis, Proceedings of an International Workshop on Interactive Decision Analysis and Interpretative Computer Intelligence*, Springer: 113-122.
23. Nakayama, H. and Tanino, T. (1994): *Multi-objective Programming-Theory and Applications*, Society of Instrument and Control Engineering, (in Japanese)
24. Nakayama, H., Kaneshige, K., Takemoto, S. and Watada, Y. (1995), An Application of a Multi-objective Programming Technique to Construction Accuracy Control of Cable-stayed Bridges *European J. of Operational Research* 87, 731-738.
25. Sawaragi, Y., Nakayama, H. and Tanino, T. (1985). *Theory of Multiobjective Optimization*, Academic Press
26. Schaffer, J.D. (1985). Multiple Obejctive Optimization with Vector Evaluated Genetic Algorithms, *Proc. First International conference on Genetic Algorithms*, 93-100
27. Schölkopf, B. and Smola, A.J. (2002), *Learning with Kernels*, MIT Press, 2002
28. Ueno, N., Nakagawa, Y., Tokuyama, H., Nakayama, H. and Tamura, H., (1990), A Multi-objective Planning for String Selection in Steel Manufactureing, *Communications of the Operations Research Society of Japan*, **35**, 656-661 (in Japanese)
29. Vapnik, V.N. (1995), The Nature of Statistical Learning Theory, *Springer Verlag*, NY.
30. Yun, Y.B., Nakayama, H. and Tanino, T. (1999), A Generalization of DEA model, *Journal of the Society of Instrument and Control Engineers (SICE)*, 35(8), 1813-1818. (in Japanese)
31. Yun, Y.B., Nakayama, H., Tanino, T. and Arakawa, M. (2001), Generation of Efficient Frontiers in Multi-Objective Optimization Problems by Generalized Data Envelopment Analysis *European Journal of Operational Research*, **129** (3), 586–595.
32. Yun, Y.B., Nakayama, H. and Arakawa, M. (2004), Multiple Criteria Decision Making with Generalized DEA and an Aspiration Level Method, *European Journal of Operational Research*, **158**, 697-706
33. Yun, Y.B., Nakayama, H., Arakawa, M. and Shiraki, W. (2004), Multi-objective Optimization Technique using Computational Intelligence, to be presented at CJK-OSM3
34. Zitzler, E., Deb, H. and Thiele, L. (2000), Comparison of Multi-objective Evolutionary Algorithms: Empirical Results, *Eevolutionray Computation Journal*, **8**, 125-148