Chapter 2
Interactive Programming Methods for Multiobjective Optimization

In general, there may be many Pareto solutions in multiobjective optimization problems. The final decision is made among them taking the total balance over all objectives into account. This is a problem of value judgment of decision maker (DM). The totally balancing over criteria is usually called *trade-off*. Interactive multiobjective programming searches a solution in an interactive way with DM while eliciting information on his/her value judgment. Then it is important how easily DM can make trade-off analysis to get a final solution. To this aim, several kinds of interactive techniques for multiple criteria decision making have been developed so far. For details, see the literatures [47, 90, 130, 147, 157, 159]. Above all, the *aspiration level approach* (reference point methods in some literatures) is now widely recognized to be effective in many practical fields, because:

1. It does not require any consistency of DM's judgment.
2. Aspiration levels reflect the wish of DM very well.
3. Aspiration levels play the role of probe better than the weight for objective functions.

In this chapter, first we will discuss the difficulty in weighting method which is commonly used in the traditional goal programming, and next explain how the aspiration level approach overcomes this difficulty.

2.1 Goal Programming

Goal programming (GP) was proposed by Charnes and Cooper in 1961 to get rid of cases in which no feasible solution exists in usual linear programming (LP) [17]. In order to explain GP, we should begin with the story of birth of LP.

H. Nakayama et al., *Sequential Approximate Multiobjective Optimization Using Computational Intelligence,* Vector Optimization,
DOI 10.1007/978-3-540-88910-6_2,

Dantzig [28] retrospected the birth of LP as follows: "Initially, there was no objective function: explicit *goals* did not exist because practical planners simply had no way to implement such a concept. ... By mid-1947 I decided that the objective had to be made explicit. ... The use of linear form as the objective function to be extremized was the novel feature. ..."

LP has been applied to various kinds of practical problems since its proposal. Practitioners, in particular engineers, however, often encounter cases in which no feasible solution exists. The motivation that Dantzig introduced the objective function to be maximized seems to owe to an idea of utilitarianism that human beings try to maximize their utilities on the background. On the contrary, Simon insisted that the rationality of human behavior is in "satisficing" rather than in optimization [142].

In order to get rid of cases in which no feasible solution exists in LP, Charnes and Cooper introduced the idea of goal attainment along the idea of satisficing: they insisted that any constraints can be regarded as "goal."

Goals in mathematical programming model are classified into the following three categories:

1. $g_i(\boldsymbol{x}) \geqq \overline{g}_i, \quad i \in I_{GT},$
2. $g_i(\boldsymbol{x}) \leqq \overline{g}_i, \quad i \in I_{LT},$
3. $g_i(\boldsymbol{x}) = \overline{g}_i, \quad i \in I_{EQ},$

where $\overline{g}_i$ is called the *goal level* of g_i. The goal itself may be feasible or infeasible. In cases where the goal is infeasible, the aim of goal programming is to obtain a solution as close to the given goal level as possible.

Let

$$g(\boldsymbol{x}) + y^+ - y^- = \overline{g}, \quad y^+ \geqq 0,\ y^- \geqq 0.$$

Then the amount y^+ and y^- denotes, respectively, the *overattainment* and the *underattainment*, if $y^+y^- = 0$ and less of the level of the criterion g is preferred to more.

Now, GP is formulated in general as follows:

$$\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{y}^+,\boldsymbol{y}^-}{\text{minimize}} \quad & \sum_{i \in I_{GT}} w_i y_i^+ + \sum_{i \in I_{LT}} w_i y_i^- + \sum_{i \in I_{EQ}} w_i (y_i^+ + y_i^-) \qquad \text{(GP)} \\
\text{subject to} \quad & g_i(\boldsymbol{x}) + y_i^+ - y_i^- = \overline{g}_i, \quad i \in I_{GT} \cup I_{LT} \cup I_{EQ}, \\
& y_i^+,\ y_i^- \geqq 0, \quad i \in I_{GT} \cup I_{LT} \cup I_{EQ}, \\
& \boldsymbol{x} \in X \subset \mathbb{R}^n.
\end{aligned}$$

Remark 2.1. The condition $y_i^+ y_i^- = 0$ is crucial for y_i^+ and y_i^- to have the meaning of the overattainment and the underattainment, respectively, if less of the level of the criterion g_i is preferred to more. Fortunately, the optimal solution to (GP) satisfies this condition automatically. The following lemma proves this in a more general form.

Lemma 2.1. *Let $\boldsymbol{y}^+$ and $\boldsymbol{y}^-$ be vectors of $\mathbb{R}^r$. Then consider the following problem:*

$$\begin{array}{lll} \underset{\boldsymbol{x},\boldsymbol{y}^+,\boldsymbol{y}^-}{\textit{minimize}} & G(\boldsymbol{y}^+, \boldsymbol{y}^-) & \\ \textit{subject to} & g_i(\boldsymbol{x}) + y_i^+ - y_i^- = \overline{g}_i, \quad i = 1, \dots, r, & \\ & y_i^+,\ y_i^- \geqq 0, \quad i = 1, \dots, r, & \\ & \boldsymbol{x} \in X \subset \mathbb{R}^n. & \end{array}$$

Suppose that the function G is monotonically increasing with respect to elements of $\boldsymbol{y}^+$ and $\boldsymbol{y}^-$ and strictly monotonically increasing with respect to at least either y_i^+ or y_i^- for each i, $i = 1, \dots, r$. Then, the solution $\hat{\boldsymbol{y}}^+$ and $\hat{\boldsymbol{y}}^-$ to the above problem satisfy

$$\hat{y}_i^+ \hat{y}_i^- = 0, \qquad i = 1, \dots, r.$$

Proof. See, e.g., [130]. □

Consider a case in which less values of the criteria $f_i,\ i = 1, \dots, r$ are more preferable, but it is desirable for f_i to be at least less than $\overline{f}_i,\ i = 1, \dots, r$. For this situation, GP is formulated as follows:

$$\begin{array}{lll} \underset{\boldsymbol{x},\boldsymbol{y}^+,\boldsymbol{y}^-}{\text{minimize}} & \displaystyle\sum_i^r w_i y_i^- & \qquad (\mathrm{GP}') \\ \text{subject to} & f_i(\boldsymbol{x}) + y_i^+ - y_i^- = \overline{f}_i, \quad i = 1, \dots, r, & \\ & y_i^+,\ y_i^- \geqq 0, \quad i = 1, \dots, r, & \\ & \boldsymbol{x} \in X \subset \mathbb{R}^n. & \end{array}$$

According to Lemma 2.1, we have $y_i^+ y_i^- = 0$ for $i = 1, \dots, r$ at the optimal solution. In the above problem (GP′), therefore, y_i^- has the meaning of underattainment of the criterion f_i. Since $w_i \geqq 0,\ y_i^+ \geqq 0,\ y_i^- \geqq 0,$ for $i = 1, \dots, r$, the optimal value of objective function of (GP′) is nonnegative. If some y_i^- is positive at the solution, the goal level $\overline{f}_i$ is not attained, while if $y_i^- = 0$ at the solution, the goal level $\overline{f}_i$ is attained. In cases with $y_i^- = 0$, it should be noted that we have a solution $\hat{\boldsymbol{x}}$ with $f_i(\hat{\boldsymbol{x}}) \leqq \overline{f}_i$ but cannot improve f_i any more than the level $f_i(\hat{\boldsymbol{x}})$, even though there are feasible solutions which yield less values than $f_i(\hat{\boldsymbol{x}})$. In other words, goal programming does not necessarily assure the Pareto optimality of solution. This seems reasonable because the basic idea of goal programming is based on satisficing. However, this fact is not satisfactory in view of criteria for which less values are more preferable. It is not so difficult to find any solution which yields as less value of f_i as possible keeping less than $\overline{f}_i$, if any, for the problems which can be formulated as mathematical programming including goal programming. Therefore, this fact is one of drawbacks of goal programming.

Remark 2.2. If less level of f is more preferable, and if we aim to find a solution which yields as less level of f as possible even though it clears the goal level $\overline{f}$, then we should formulate as follows:

$$\begin{aligned} &\underset{\boldsymbol{x},\, z}{\text{minimize}} && \sum_{i=1}^{r} w_i z_i && \text{(GP}''\text{)} \\ &\text{subject to} && f_i(\boldsymbol{x}) - \overline{f}_i \leqq z_i, \quad i = 1, \ldots, r, \\ & && \boldsymbol{x} \in X \subset \mathbb{R}^n. \end{aligned}$$

Note that the new variable z is not necessarily nonnegative. The stated problem is equivalent to multiobjective programming problem:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} && \sum_{i=1}^{r} w_i f_i(\boldsymbol{x}) \\ &\text{subject to} && \boldsymbol{x} \in X \subset \mathbb{R}^n. \end{aligned}$$

However, by (GP$''$), we can treat f_i not only as an objective function but also as a constraint by setting $z_i = 0$. This technique will be used to interchange the role of objective function and constraint in subsequent sections.

From a viewpoint of decision making, it is important to obtain a solution which the decision maker accepts easily. This means it is important whether the obtained solution is suited to the value judgment of the decision maker. In goal programming, the adjustment of weights w_i is required to obtain a solution suited to the value judgment of the decision maker. However, this task is very difficult in many cases as will be seen in the following.

2.2 Why is the Weighting Method Ineffective?

In multiobjective 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

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

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 cannot provide a solution among sunken parts of Pareto surface (frontier) 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 cannot necessarily provide a solution as DM desires.

In an extended form of goal programming, e.g., *compromise programming* [165], some kind of metric function from the goal $\overline{\boldsymbol{f}}$ is used as the one representing the preference of DM. For example, the following is well known:

$$\left(\sum_{i=1}^{r} w_i |f_i(\boldsymbol{x}) - \overline{f}_i|^p \right)^{1/p} . \tag{2.1}$$

The preference of DM is reflected by the weight w_i, the value of p, and the value of the goal $\overline{f}_i$. If the value of p is chosen appropriately, a Pareto solution among a sunken part of Pareto surface can be obtained by minimizing (2.1). However, it is usually difficult to predetermine appropriate values of them. Moreover, the solution minimizing (2.1) cannot be better than the goal $\overline{\boldsymbol{f}}$, even though the goal is pessimistically underestimated.

In addition, one of the most serious drawbacks in the weighted sum scalarization 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 $\boldsymbol{f}(\hat{\boldsymbol{x}})$ corresponding to the resulting solution $\hat{\boldsymbol{x}}$ as will be seen in the following example.

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

$$\left\{ (y_1, y_2, y_3) \mid (y_1 - 1)^2 + (y_2 - 1)^2 + (y_3 - 1)^2 \leqq 1 \right\}.$$

Suppose that the goal is $(\overline{y}_1,\ \overline{y}_2,\ \overline{y}_3) = (0,\ 0,\ 0)$. The solution minimizing the metric function (2.1) with $p = 1$ and $w_1 = w_2 = w_3 = 1$ is $(y_1,\ y_2,\ y_3) =$ (0.42265, 0.42265, 0.42265). 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 (0.02410, 0.80482, 0.90241). Note that the value of f_2 is worse than before despite that DM wants to improve it by increasing the weight of f_2 up to twice. Someone might think that this is due to the lack of 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 (0.18350, 0.42845, 0.91835). 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. This difficulty seems to be caused due to the fact that there is no positive correlation between the weight and the resulting solution in cases with more than two objective functions. Therefore, it seems much better to take another probe for getting a solution which DM desires. The aspiration level of DM is promising as the probe. Interactive multiobjective programming techniques based on aspiration levels have been developed so that the drawbacks of the traditional weighting method may be overcome [13,157]. In Sect. 2.3, we shall discuss the *satisficing trade-off method* developed by one of the authors (Nakayama [100]) as an example.

2.3 Satisficing Trade-off Method

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

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

Here, the operator P selects the Pareto solution nearest in some sense to the given aspiration level $\overline{\boldsymbol{f}}^k$. The operator T is the trade-off operator which changes the kth aspiration level $\overline{\boldsymbol{f}}^k$ if DM does not compromise with the shown solution $P(\overline{\boldsymbol{f}}^k)$. Of course, since $P(\overline{\boldsymbol{f}}^k)$ is a Pareto solution, there exists no feasible solution which makes all criteria better than $P(\overline{\boldsymbol{f}}^k)$, 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 $T \circ P(\overline{\boldsymbol{f}}^k)$. Similar process is continued until DM obtains an agreeable solution. This idea is implemented in DIDASS [51] and the satisficing trade-off method [100]. While DIDASS leaves the trade-off to the heuristics of DM, the satisficing trade-off method provides a device based on the sensitivity analysis which will be stated later.

2.3.1 On the Operation P

The operation which gives a Pareto solution $P(\overline{\boldsymbol{f}}^k)$ nearest to $\overline{\boldsymbol{f}}^k$ is performed by some *auxiliary scalar optimization*. It has been shown in [130] that the only one scalarization technique, which provides any Pareto solution regardless of the structure of problem, is of the Tchebyshev type. As was stated before, however, the Tchebyshev type scalarization function 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 criteria while others being fixed, they are not necessarily “efficient” as a solution in decision making. In order to exclude weak Pareto solutions, we apply the *augmented Tchebyshev scalarization function*:

$$\max_{1 \leq i \leq r} w_i(f_i(\boldsymbol{x}) - \overline{f}_i) + \alpha \sum_{i=1}^{r} w_i f_i(\boldsymbol{x}), \tag{2.2}$$

where α is usually set a sufficiently small positive number, say 10^{-6}. 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 $f_i^* < \min\{f_i(\boldsymbol{x}) \mid \boldsymbol{x} \in X\}$, and let f_{*i} be a nadir value which is usually given by

$$f_{*i} = \max_{1 \leq j \leq r} f_i(\boldsymbol{x}_j^*),$$

where

$$\boldsymbol{x}_j^* = \arg \min_{\boldsymbol{x} \in X} f_j(\boldsymbol{x}).$$

For this circumstance, we set

$$w_i^k = \frac{1}{\overline{f}_i^k - f_i^*} \tag{2.3}$$

or

$$w_i^{k'} = \frac{1}{f_{*i} - f_i^*}. \tag{2.4}$$

The minimization of (2.2) with (2.3) or (2.4) is usually performed by solving the following equivalent optimization problem, because the original one is not smooth:

$$\begin{aligned}
&\underset{\boldsymbol{x},\, z}{\text{minimize}} && z + \alpha \sum_{i=1}^{r} w_i f_i(\boldsymbol{x}) && \text{(AP)}\\
&\text{subject to} && w_i^k \left(f_i(\boldsymbol{x}) - \overline{f}_i^k \right) \leqq z, \quad i = 1, \ldots, r && (2.5)\\
& && \boldsymbol{x} \in X. &&
\end{aligned}$$

Remark 2.3. Note the weight (2.3) depends on the kth aspiration level, while the one by (2.4) is independent of aspiration levels. The difference between solutions to (AP) for these two kinds of weight can be illustrated in Fig. 2.1. In the auxiliary min–max problem (AP) with the weight by (2.3), $\overline{f}_i^k$ in the constraint (2.5) may be replaced with f_i^* without any change in the solution. For we have

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

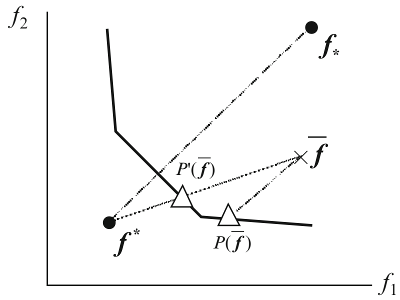

Fig. 2.1 Difference between solutions associated with two kinds of weight

The following theorem is a slight modification of Lemmas 7.3–7.5 of Sawaragi–Nakayama–Tanino [130]. It assures that we can obtain a desirable Pareto solution by adjusting the aspiration level $\overline{\boldsymbol{f}}$ appropriately.

Theorem 2.1. *For arbitrary $\boldsymbol{w} \geq 0$ and $\alpha > 0$, $\hat{\boldsymbol{x}} \in X$ minimizing (2.2) is a properly Pareto optimal solution to (MOP). Conversely, if $\hat{\boldsymbol{x}}$ is a properly Pareto optimal solution to (MOP), then there exist $\alpha > 0$ and $\overline{\boldsymbol{f}}$ such that $\hat{\boldsymbol{x}}$ minimizes (2.2) (or equivalently (AP)) with $\boldsymbol{w}$ decided by (2.3) or (2.4) over X.*

2.3.2 On the Operation T

In cases 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 $\boldsymbol{x}^k$ denote the Pareto solution obtained by operation $P(\overline{\boldsymbol{f}}^k)$, and classify the objective functions into the following three groups:

1. The class of criteria I_I^k which are to be improved more
2. The class of criteria I_R^k which may be relaxed
3. The class of criteria I_A^k which are acceptable as they are

Clearly, $\overline{f}_i^{k+1} < f_i(\boldsymbol{x}^k)$ for all $i \in I_I^k$. Usually, for $i \in I_A^k$, we set $\overline{f}_i^{k+1} = f_i(\boldsymbol{x}^k)$. 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 $i \in I_I^k$.

Remark 2.4. It is not necessarily needed to classify all of the objective functions explicitly. Indeed, in many practical cases, the objective functions are automatically classified after setting new aspiration levels. However, in using *automatic trade-off* or *exact trade-off* which will be stated later, we need classify criteria to be relaxed and accepted to decide their amounts of relaxation.

Example 2.2. Consider the same problem as in Example 2.1: Let $f_1 := y_1$, $f_2 := y_2$ and $f_3 := y_3$, and let the feasible region in the objective space be given by

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

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 (2.4) 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 (AP) 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 (AP) 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, it is also possible to use the sensitivity analysis in mathematical programming (refer to the following automatic trade-off or exact trade-off). We have observed in the previous section that it is difficult to adjust weights for criteria so that we may get a desirable solution in the goal programming. However, the aspiration level can lead DM to his/her desirable solution easily in many practical problems.

Remark 2.5. The idea of classification and trade-off of criteria is originated from STEM [7] and followed also by NIMBUS developed by Miettinen [90].

2.3.3 Automatic Trade-off

It is of course possible for DM to answer new aspiration levels of all objective functions. In practical problems, however, we often encounter cases with a large number of objective functions, for which DM tends to get tired with answering new aspiration levels for all objective functions. Therefore, it is more practical in problems with very many objective functions for DM to answer only his/her improvement rather than both improvement and relaxation. Using the properties described in Sect. 1.4 for the augmented Tchebyshev scalarization function, we have the following relation: For some perturbation $\Delta f_i, \;\; i = 1, \ldots, r$ from a Pareto value,

$$0 = \sum_{i=1}^{r} (\lambda_i + \alpha) w_i \Delta f_i + o(\| \Delta f \|), \tag{2.6}$$

where λ_i is the Lagrange multiplier associated with the constraints in problem (AP). Therefore, under some appropriate condition, $((\lambda_1 + \alpha)w_1, \ldots, (\lambda_r + \alpha)w_r)$ is the normal vector of the tangent hyperplane of the Pareto surface. In particular, in multiobjective linear programming problems, the simplex multipliers corresponding to a nondegenerated solution is the feasible trade-off vector of Pareto surface [98].

Dividing the right-hand side of (2.6) into the total amount of improvement and that of relaxation,

$$0 \cong \sum_{i \in I_I} (\lambda_i + \alpha) w_i \Delta f_i + \sum_{j \in I_R} (\lambda_j + \alpha) w_j \Delta f_j .$$

Using the above relation, we can assign the amount of sacrifice for f_j $(j \in I_R)$ which is automatically set in the equal proportion to $(\lambda_i + \alpha)w_i$, namely, by

$$\Delta f_j = \frac{-1}{N(\lambda_j + \alpha) w_j} \sum_{i \in I_I} (\lambda_i + \alpha) w_i \Delta f_i, \quad j \in I_R, \tag{2.7}$$

where N is the number of elements of the set I_R.

By using the above automatic trade-off method, the burden of DM can be decreased so much in cases that there are a large number of criteria. Of course, if DM does not agree with this quota Δf_j laid down automatically, he/she can modify it in a manual way.

Example 2.3. Consider the same problem as in Example 2.2. Suppose that DM has the solution $(y_1^1, \; y_2^1, \; y_3^1) = (0.423, \; 0.423, \; 0.423)$ associated with his first aspiration level $(\overline{y}_1^1, \; \overline{y}_2^1, \; \overline{y}_3^1) = (0.4, \; 0.4, \; 0.4)$. Now suppose that DM modifies the aspiration level into $\overline{y}_1^2 = 0.35$ and $\overline{y}_2^2 = 0.4$. For the amount of improvement of $|\Delta f_1| = 0.073$ and $|\Delta f_2| = 0.023$, the amount of relaxation of f_3 on the basis of automatic trade-off is $|\Delta f_3| = 0.095$. In other words, the new aspiration level of f_3 should be 0.52. If DM agrees with this trade-off, he/she will have the new Pareto solution $(y_1^2, \; y_2^2, \; y_3^2) = (0.354, \; 0.404, \; 0.524)$ to the problem (AP) corresponding to the new aspiration level $(\overline{y}_1^2, \; \overline{y}_2^2, \; \overline{y}_3^2) = (0.35, \; 0.4, \; 0.52)$. It should be noted that the obtained solution is much closer to DM's wish rather than the one in Example 2.2.

Example 2.4. Consider the following multiple objective linear programming problem:

$$
\begin{aligned}
\underset{x_1,\, x_2}{\text{minimize}} \quad & f_1 = -2x_1 - x_2 + 25 \\
& f_2 = x_1 - 2x_2 + 18 \\
\text{subject to} \quad & -x_1 + 3x_2 \leqq 21, \\
& x_1 + 3x_2 \leqq 27, \\
& 4x_1 + 3x_2 \leqq 45, \\
& 3x_1 + \ x_2 \leqq 30, \\
& x_1,\ x_2 \geqq 0.
\end{aligned}
$$

Suppose that the ideal point $\boldsymbol{f}^* = (4,\ 4)^T$ and the nadir point $\boldsymbol{f}_* = (18,\ 21)^T$ by using the payoff matrix based on minimization of each objective function separately. Letting the first aspiration level be $\overline{\boldsymbol{f}}^1 = (15,\ 9)^T$, we have the first Pareto solution (11.632, 4.910) by solving the auxiliary min–max problem (AP). This Pareto point in the objective function space is the intercept of the line parallel to the line passing through $\boldsymbol{f}^*$ and $\boldsymbol{f}_*$ with the Pareto surface (curve, in this case). Now we shall consider the following three cases:

Case 1. Suppose that DM is not satisfied with the obtained Pareto solution, and he/she wants to improve the value of f_2. Let the new aspiration level of f_2 be 4.5. The relaxation amount for f_1 calculated by the automatic trade-off is 2.87. Therefore, the new aspiration level of f_1 based on the automatic trade-off is 14.5. Since the automatic trade-off is based on the linear approximation of Pareto surface at the present point, the new aspiration level obtained by the automatic trade-off is itself Pareto optimal in this case as shown in Fig. 2.2.

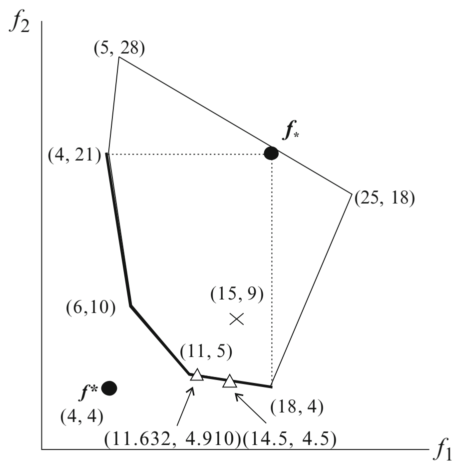

Fig. 2.2 Automatic trade-off (case 1)

Case 2. Suppose that DM wants to improve f_1 rather than f_2 at the moment when the first Pareto solution $(11.632, 4.910)$ is obtained. Let the new aspiration level of f_1 that DM desires be 9.0. Then the new aspiration level by the automatic trade-off is $\overline{\boldsymbol{f}}^2 = (9.0,\ 5.286)^T$, and is not Pareto optimal. Solving the auxiliary min–max problem (AP) with the new aspiration level, in this case, we have the new Pareto solution $(9.774, 6.226)$. Since the improvement which DM desires is not so large after solving the min–max problem (AP) with an aspiration level in many practical cases, the new aspiration level produced by automatic trade-off based on the linear approximation of Pareto surface is close to the Pareto surface. Therefore, the satisficing trade-off method using the automatic trade-off yields a desirable solution usually only in a few iterations (Fig. 2.3).

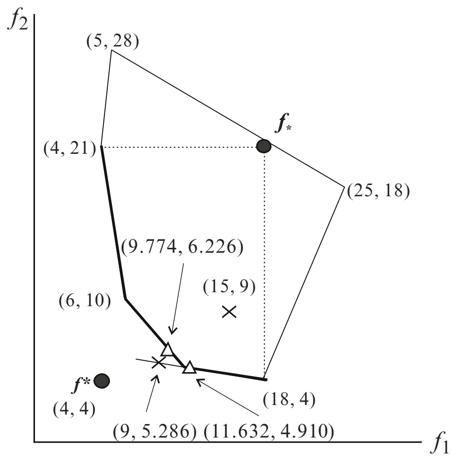

Fig. 2.3 Automatic trade-off (case 2)

Case 3. Suppose that DM wants to make f_1 less than 9.0 by all means at the moment when the first Pareto solution $(11.632, 4.910)$ is obtained. In this case, we should treat f_1 as the constraint

$$f_1(x) \leqq 9.0.$$

As will be shown in Sect. 2.3.5, the interchange between objectives and constraints can be made so easily in the formulation of auxiliary min–max problem (we can treat the criteria as DM wishes between objectives and constraints by adjusting one parameter β in the min–max problem) (Fig. 2.4).

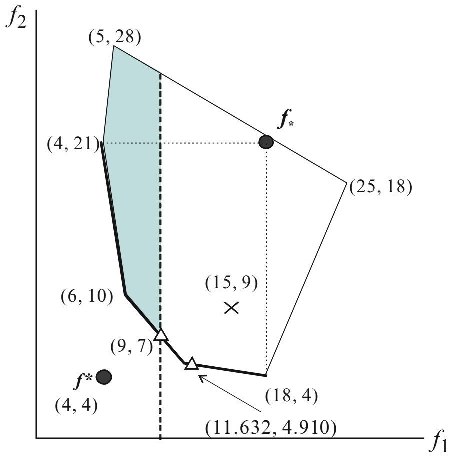

Fig. 2.4 Automatic trade-off (case 3)

2.3.4 Exact Trade-off

In linear or quadratic cases, we can evaluate the exact trade-off in an extended form of the automatic trade-off stated above. This implies that we can calculate exactly the amount of relaxation such that the new aspiration level is on the Pareto surface (Nakayama [98]). The main idea in it is that the parametric optimization technique is used in stead of the simple sensitivity analysis. Using this technique, a new Pareto solution can be obtained without solving the auxiliary scalarized optimization problem again. This implies that we can obtain a new solution very quickly. Therefore, using some graphic presentation as computer outputs, DM can see the trade-off among criteria in a dynamic way, e.g., as an animation. This makes DM's judgment easier (see Fig. 2.5).

2.3.5 Interchange Between Objectives and Constraints

In the formulation of the auxiliary scalarized optimization problem (AP), change the right-hand side of (2.5) into $\beta_i z$, namely

$$w_i(f_i(\boldsymbol{x}) - \overline{f}_i) \leqq \beta_i z. \tag{2.8}$$

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

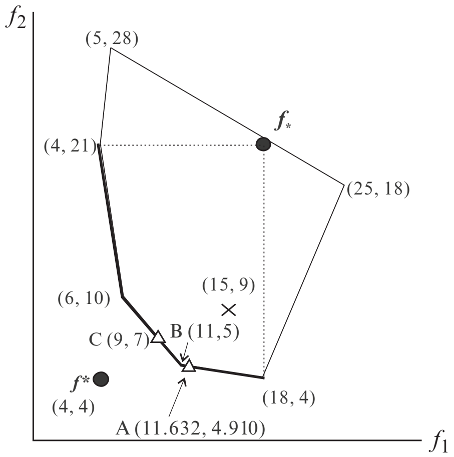

Fig. 2.5 Exact trade-off

is almost no case in which we consider the role of objective and constraint fixed from the beginning to the end, but usually we want to interchange them depending on the situation. Using (2.8), this can be done very easily (Korhonen et al. [75]). 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.

2.3.6 Remarks on Trade-off for Objective Functions with 0-Sensitivity

Since α is sufficiently small like 10^{-6} and $\lambda_1 + \dots \lambda_r = 1$, we can consider in many cases

$$(\lambda_i + \alpha)w_i \simeq \lambda_i w_i.$$

When $\lambda_j = 0$ and f_j is not to be improved, we set $\Delta f_j = 0$ in the automatic trade-off. Therefore, unless we select at least one f_j with $\lambda_j \neq 0$ as an objective function to be relaxed, we cannot attain the improvement that DM wishes.

Since $(\lambda_i + \alpha)w_i$ (or approximately, $\lambda_i w_i$) can be regarded to provide the sensitivity information in the trade-off, $\lambda_i = 0$ means that the objective function f_i does not contribute to the trade-off among the objective functions. In other words, since the trade-off is the negative correlation among objective functions, $\lambda_i = 0$ means that f_i has the nonnegative correlation with some other objective functions. Therefore, if all objective functions to be relaxed, f_j, $(j \in I_R)$ have $\lambda_j = 0$ $(j \in I_R)$, then they cannot compensate for the improvement which DM wishes, because they are affected positively by some of objective functions to be improved.

Example 2.5. Consider the following problem (Fig. 2.6):

$$\begin{aligned} \underset{x_1,\,x_2,\,x_3}{\text{minimize}} \quad & (f_1,\ f_2,\ f_3) = (x_1,\ x_2,\ x_3)^T \\ \text{subject to} \quad & x_1 + x_2 + x_3 \geqq 1, \\ & x_1 \qquad - x_3 \geqq 0, \\ & x_1,\ x_2,\ x_3 \geqq 0. \end{aligned}$$

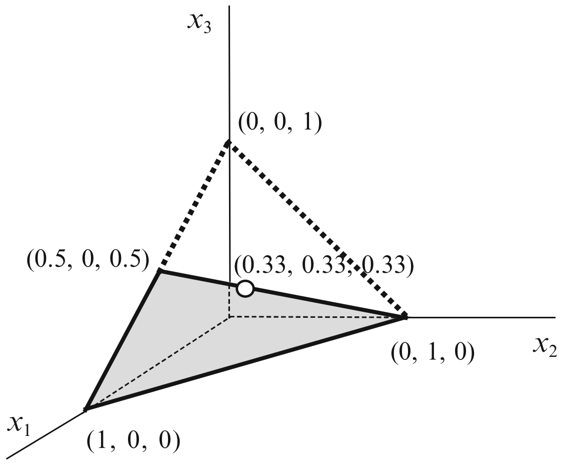

Fig. 2.6 Case of automatic trade-off with 0-sensitivity

For the first aspiration level (0.2, 0.2, 0.4), we have a Pareto value (0.333, 0.333, 0.333) and the corresponding simplex multiplier $(\lambda_1,\ \lambda_2,\ \lambda_3) = (2/3,\ 1/3,\ 0)$. Suppose that DM wants to improve f_1 and f_2, and sets their new aspiration levels 0.2 and 0.3, respectively. Since the relaxation $\Delta f_3 = 0$ by the automatic trade-off, the new aspiration level becomes (0.2, 0.3, 0.3). Associated with the new aspiration level, we have the Pareto value (0.275, 0.45, 0.275), in which neither f_2 is improved nor f_3 is relaxed. This is because that the objective functions f_1 and f_3 have a positive correlation along the edge of Pareto surface at the point $(0.333, 0.333, 0.333)$, while f_1 and f_2 have trade-off relation with each other there. As a result, though f_3 was considered to be relaxed, it was affected strongly by f_1 and hence improved. On the other hand, despite that f_2 was considered to be improved, it was relaxed finally. This is due to the fact that the objective function to be relaxed is only f_3 despite that λ_3 is 0, and due to the fact that we did not take into account that f_3 has positive correlation with f_1. This example suggests that we should make the trade-off analysis deliberately seeing the value of simplex multiplier (Lagrange multiplier, in nonlinear cases). Like this, the satisficing trade-off method makes the DM's trade-off analysis easier by utilizing the information of sensitivity.

2.3.7 Relationship to Fuzzy Mathematical Programming

In the aspiration level approach to multiobjective programming such as the satisficing trade-off method, the wish of DM is attained by adjusting his/her aspiration level. In other words, this means that the aspiration level approach can deals with the fuzziness of right-hand side value in traditional mathematical programming as well as the total balance among the criteria. There is another method, namely *fuzzy mathematical programming*, which treats the fuzziness of right-hand side value of constraint in traditional mathematical programming. In the following, we shall discuss the relationship between the satisficing trade-off method and the fuzzy mathematical programming.

For simplicity, consider the following problem:

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{maximize}} & f_0(\boldsymbol{x}) \\ \text{subject to} & f_1(\boldsymbol{x}) = \overline{f}_1. \end{array} \tag{F}$$

Suppose that the right-hand side value $\overline{f}_1$ is not needed to meet so strictly, but that it is fuzzy. The *membership function* for the criterion f_1 is usually given as in Fig. 2.7. Since our aim is to maximize this membership function, we can adopt the following function without change in the solution:

$$m_1(\boldsymbol{x}) = \min \left\{ \frac{\overline{f}_1 - f_1(\boldsymbol{x})}{\varepsilon} + 1, \ -\frac{\overline{f}_1 - f_1(\boldsymbol{x})}{\varepsilon} + 1 \right\},$$

where ε is a parameter representing the admissible error for the target $\overline{f}_1$.

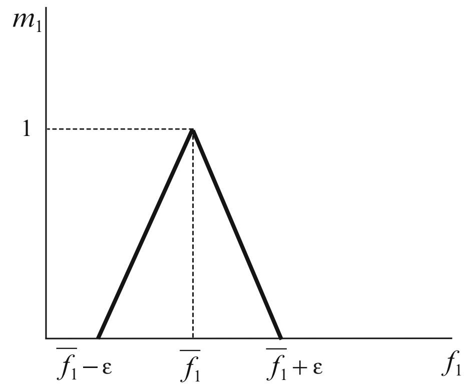

Fig. 2.7 Membership function for f_1 in (F)

Now, the problem (F) is reduced to a kind of multiobjective optimization in which f_0 and m_1 should be maximized. Then a membership function for maximization of f_0 is usually defined with its aspiration level $\overline{f}_0$. For example,

$$m_0'(\boldsymbol{x}) = \min\left\{ -\frac{\overline{f}_0 - f_0(\boldsymbol{x})}{\overline{f}_0 - f_{0*}} + 1,\ 1 \right\}.$$

However, if we maximize the above m_0' as it is, the solution will be merely the one to the satisficing problem for which f_0 is to be just greater than $\overline{f}_0$. As was stated in the previous section, we shall use the following function in stead of m_0' in order to assure the Pareto optimality of the solution:

$$m_0(\boldsymbol{x}) = -\frac{\overline{f}_0 - f_0(\boldsymbol{x})}{\overline{f}_0 - f_{0*}} + 1.$$

Finally, our problem is to maximize both m_0 and m_1, which is usually reduced to the following problem:

$$\begin{aligned}
&\underset{\boldsymbol{x},\, z}{\text{minimize}} && z \\
&\text{subject to} && \frac{\overline{f}_0 - f_0(\boldsymbol{x})}{\overline{f}_0 - f_{0*}} - 1 \leqq z, \\
& && \frac{\overline{f}_1 - f_1(\boldsymbol{x})}{\varepsilon} - 1 \leqq z, \\
& && -\frac{\overline{f}_1 - f_1(\boldsymbol{x})}{\varepsilon} - 1 \leqq z.
\end{aligned}$$

Now, one may see some similarity between the above formulation and the one in the satisficing trade-off method. In the satisficing trade-off method, the objective function with target such as $f_1 \to \overline{f}_1$ is usually treated as two objective functions, $f_1 \to$ max and $f_1 \to$ min. Under this circumstance, suppose that for $f_1 \to$ max we set the ideal value $f_1^* = \overline{f}_1$, the nadir value $f_{1*} = \overline{f}_1 - \varepsilon$ and the aspiration level $\overline{f}_1 - \varepsilon$, and for $f_1 \to$ min we set the ideal value $f_1^* = \overline{f}_1$, the nadir value $f_{1*} = \overline{f}_1 + \varepsilon$ and the aspiration level $\overline{f}_1 + \varepsilon$. Then the treatment of f_1 is the same between the above formulation and the satisficing trade-off method.

However, usually in the satisficing trade-off method we do not contain ε in the denominator of constraints in the min–max problem, because we make the trade-off analysis by adjusting ε rather than the target $\overline{f}_1$; for example, using (2.4) the constraint for f_1 in the min–max problem is given by

$$\begin{aligned}
\frac{\overline{f}_1 - \varepsilon - f_1(x)}{f_1^* - f_{1*}} &\leqq z, \\
-\frac{\overline{f}_1 + \varepsilon - f_1(x)}{f_1^* - f_{1*}} &\leqq z.
\end{aligned}$$

With this formulation, even if DM wants $\varepsilon = 0$ and if there is no solution to $f_1(\boldsymbol{x}) = \overline{f}_1$, we can get a solution approximate to $f_1(\boldsymbol{x}) = \overline{f}_1$ as much as possible. In the fuzzy mathematical programming, however, if $\varepsilon = 0$, then we have a crisp constraint $f_1(\boldsymbol{x}) = \overline{f}_1$, and we sometimes have no feasible solution to it.

Finally as a result, we can see that the satisficing trade-off method deals with the fuzziness of right-hand side value of constraint automatically and can effectively treat problems for which fuzzy mathematical programming provides no solution. Due to this reason, we can conclude that it is better to formulate the given problem as a multiobjective optimization from the beginning and to solve it by the aspiration level approach such as the satisficing trade-off method.

2.4 Applications

Interactive multiobjective programming methods have been applied to a wide range of practical problems. Good examples in engineering applications can be seen in Eschenauer et al. [39]. One of the authors himself also has applied to several real problems: feed formulation for live stock [91,104], plastic materials blending [103], cement production [97], bond portfolio [96], erection management of cable-stayed bridges [46,105,109], scheduling of string selection in steel manufacturing [152].

In the following, some of examples are introduced briefly.

2.4.1 Feed Formulation for Live Stock

Stock farms in Japan are modernized recently. Above all, the feeding system in some farms is fully controlled by computer: Each cow has its own place to eat which has a locked gate. And each cow has a key on her neck, which can open the corresponding gate only. Everyday, on the basis of ingredient analysis of milk and/or of the growth situation of cow, the appropriate blending ratio of materials from several viewpoints should be made.

There are about 20–30 kinds of raw materials for feed in cow farms such as corn, cereals, fish meal, etc. About ten criteria are usually taken into account for feed formulation of cow:

- Cost
- Nutrition
  - Protein
  - TDN
  - Cellulose

  - Calcium
  - Magnesium
  - etc.
- Stock amount of materials
- etc.

This feeding problem is well known as the diet problem from the beginning of the history of mathematical programming, which can be formulated as the traditional linear programming. In the traditional mathematical programming, the solution often occurs on the boundary of constraints. In many cases, however, the right-hand side value of constraint such as nutrition needs not to be satisfied rigidly. Rather, it seems to be natural to consider that a criterion such as nutrition is an objective function whose target has some allowable range. As was seen in the previous section, the satisficing trade-off method deals well with the fuzziness of target of such an objective function. The author and others have developed a software for feed formulation using the satisficing trade-off method, called F-STOM (feed formulation by satisficing trade-off method [91, 98]). This software is being distributed to live stock farmers and feed companies in Japan through an association of live stock systems.

2.4.2 Erection Management of Cable-Stayed Bridge

In erection of cable-stayed bridge, the following criteria are considered for accuracy control [46, 105, 109]:

1. Residual error in each cable tension
2. Residual error in camber at each node
3. Amount of shim adjustment for each cable
4. 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, $\Delta T_i \ \ (i = 1, \ldots, n)$ as the difference between the designed tension values and the measured ones, and x_{ik} as the tension change of ith cable caused from the change of the kth cable length by a unit. The residual error in cable tension caused by the shim adjustment of $\Delta l_1, \ldots, \Delta l_n$ is given by

$$p_i = |\Delta T_i - \sum_{k=1}^{n} x_{ik} \Delta l_k|, \ \ i = 1, \ldots, n.$$

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

$$q_j = |\Delta Z_j - \sum_{k=1}^{n} y_{jk} \Delta l_k|, \ \ j = 1, \ldots, m.$$

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

$$r_i = |\Delta l_i|, \ \ i = 1, \ldots, n.$$

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

$$\Delta l_{Li} \leq \Delta l_i \leq \Delta l_{Ui}, \ \ i = 1, \ldots, n.$$

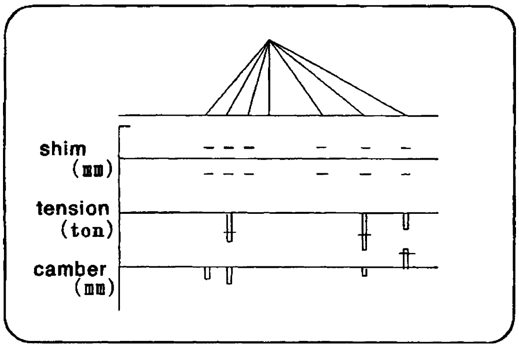

Fig. 2.8 Phase of an erection management system

Figure 2.8 shows one phase of erection management system of cable-stayed bridge using the satisficing trade-off method. 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, Tokiwa Great Bridge (Ube City) and Karasuo Harp Bridge (Kita-Kyusyu City) in 1992.

2.4.3 An Interactive Support System for Bond Trading

In portfolio selection problems, many companies are now widely trying to use mathematical analysis for bond trading. In this section, some bond trading problem is formulated as a kind of multiobjective fractional problem. It will be seen in the following that the satisficing trade-off method can be effectively applied to such a portfolio problem.

Bond traders are facing almost everyday a problem which bonds and what amount they should sell and/or buy in order to attain their customers' desires. The economic environment is changing day by day, and sometimes gives us a drastic change. Bond traders have to take into account many factors, and make their decisions very rapidly and flexibly according to these changes. The number of bonds to be considered is usually more than 500, and that of criteria is about ten as will be shown later. The amount of trading is usually more than 100 million yen, and hence even a slight difference of portfolio combination might cause a big difference to profit or loss. This situation requires some effective method which helps bond traders following faithfully their value judgment on a real-time basis not only mathematically but also in such a way that their intuition fostered by their experiences can be taken in.

Bond portfolio problems are a kind of blending problems. Therefore, mathematical programming approach can be used very effectively. However, the traditional mathematical programming approach with a single objective function cannot take in the value judgment and intuition of bond traders so easily in a flexible manner for the changes of environment. We shall show that the satisficing trade-off method fits to this purpose.

Mathematical Formulation for Bond Portfolio Problems

We shall follow the mathematical model given by Konno and Inori [73]. Assume that an investor holds u_j units of bonds $B_j, \;\; j = 1, \dots, N$. Associated with B_j, we have the following indices:

- c_j: coupon to be paid at a fixed rate (yen/bond/year)
- f_j: principal value to be refunded at maturity (yen/bond)
- p_j: present price in the market (yen/bond)
- t_j: maturity (number of years until its principal value is refunded)

Returns from bonds are the income from coupon and the capital gain due to price increase. Bond portfolio problems are to determine which bonds and what amount the investor should sell and/or buy taking into account many factors, say, expected returns and risk, the time needs money for another investment, and so on. Therefore, we have the following criteria for each bond B_j:

I. Returns

1. Direct yield (short-term index of return)

$$\gamma_j = \frac{c_j}{p_j}$$

2. Effective yield

$$\nu_j = \frac{\left[c_j\{(1+\alpha)^{t_j} - 1\}/\alpha + f_j\right]^{1/t_j}}{p_j} - 1,$$

where α is the interest rate.

II. Risk

3. Price variation

$$\pi_j = \frac{t_j}{1+\nu_j t_j}, \quad j = 1, \ldots, N$$

Let $x_j,\ j = 1, \ldots, n_1$ and $X_k,\ k = 1, \ldots, n_2$ denote the number of bonds to be sold and to be purchased, respectively. Then S_0 and S_1 represent, respectively, the total quantity of bonds and the total value of bonds after the transaction. Namely,

$$S_0 = \sum_{j=1}^{N} u_j - \sum_{j=1}^{n_1} x_j + \sum_{k=1}^{n_2} X_k,$$

$$S_1 = \sum_{j=1}^{N} p_j u_j - \sum_{j=1}^{n_1} p_j x_j + \sum_{k=1}^{n_2} P_k X_k.$$

In addition, we set

$$S_2 = \sum_{j=1}^{N} p_j t_j u_j - \sum_{j=1}^{n_1} p_j t_j x_j + \sum_{k=1}^{n_2} P_k T_k X_k.$$

Then the average for each index is taken as an objective function:

1′. Average direct yield

$$F_1 = \frac{\sum_{j=1}^{N} \gamma_j p_j u_j - \sum_{j=1}^{n_1} \gamma_j p_j x_j + \sum_{k=1}^{n_2} \gamma_k P_k X_k}{S_1}$$

2′. Average effective yield

$$F_2 = \frac{\sum_{j=1}^{N} \nu_j p_j t_j u_j - \sum_{j=1}^{n_1} \nu_j p_j t_j x_j + \sum_{k=1}^{n_2} \nu_k P_k T_k X_k}{S_2}$$

3′. Average price variation

$$F_3 = \frac{\sum_{j=1}^{N} \pi_j u_j - \sum_{j=1}^{n_1} \pi_j x_j + \sum_{k=1}^{n_2} \pi_k X_k}{S_0}$$

Our constraints are divided into soft constraints and hard constraints:

4. Average unit price

$$F_4 = \frac{\sum_{j=1}^{N} p_j u_j - \sum_{j=1}^{n_1} p_j x_j + \sum_{k=1}^{n_2} P_k X_k}{S_0}$$

5. Average maturity

$$F_5 = \frac{\sum_{j=1}^{N} t_j u_j - \sum_{j=1}^{n_1} t_j x_j + \sum_{k=1}^{n_2} T_k X_k}{S_2}$$

6. Specification of time of coupon to be paid

$$F_6 = \frac{\sum_{i \in I_1} x_i}{S_0}, \quad F_7 = \frac{\sum_{i \in I_2} x_i}{S_0},$$

where I_m is the set of indices of bonds whose coupon are paid at the time t_m.

III. Hard constraints

7. Budget constraints

$$\sum_{j=1}^{n_1} -p_j x_j + \sum_{k=1}^{n_2} P_k X_k \leqq C$$

$$\sum_{k=1}^{n_2} P_k X_k \geqq C$$

8. Specification of bond

$$
\begin{aligned}
l_j \leqq x_j \leqq u_j,\ j = 1, \ldots, n_1 \\
L_k \leqq X_k \leqq U_k,\ k = 1, \ldots, n_2
\end{aligned}
$$

For this kind of problem, the satisficing trade-off method can be effectively applied. Then we have to solve a linear fractional min–max problem. In the following section, we shall give a brief review of the method for solving it.

An Algorithm for Linear Fractional Min–Max Problems

Let each objective function in our bond trading problem be of the form $F_i(\boldsymbol{x}) = p_i(\boldsymbol{x})/q_i(\boldsymbol{x}),\ \ i = 1, \ldots, r$ where p_i and q_i are linear in $\boldsymbol{x}$. Then since

$$
F_i^* - F_i(\boldsymbol{x}) = \frac{F_i^* q_i(\boldsymbol{x}) - p_i(\boldsymbol{x})}{q_i(\boldsymbol{x})} := \frac{f_i(\boldsymbol{x})}{g_i(\boldsymbol{x})},
$$

the auxiliary min–max problem (2.2) becomes a kind of linear fractional min–max problem. For this kind of problem, several methods have been developed: Here we introduce a Dinkelbach-type algorithm (Borde–Crouzeix [11] and Ferland–Potvin [42]) as is stated in the following:

Step 1. Let $\boldsymbol{x}^0 \in X$. Set $\theta^0 = \max\limits_{1 \leq i \leq r} f_i(\boldsymbol{x}^0)/g_i(\boldsymbol{x}^0)$ and $k = 0$.

Step 2. Solve the problem

$$
T_k(\theta^k) = \min_{\boldsymbol{x} \in X} \max_{1 \leq i \leq r} \frac{f_i(\boldsymbol{x}) - \theta^k g_i(\boldsymbol{x})}{g_i(\boldsymbol{x}^k)}. \qquad (P_k)
$$

Let $\boldsymbol{x}^{k+1}$ be a solution to (P_k).

Step 3. If $T_k(\theta^k) = 0$ then stop: θ^k is the optimal value of the given min–max Problem, and $\boldsymbol{x}^{k+1}$ is the optimal solution.

If not, take $\theta^{k+1} = \max\limits_{1 \leq i \leq r} f_i(\boldsymbol{x}^{k+1})/g_i(\boldsymbol{x}^{k+1})$. Replace k by $k+1$ and go to Step 2.

Note that the problem (P_k) is the usual linear min–max problem. Therefore, we can obtain its solution by solving the following equivalent problem in a usual manner:

$$
\begin{aligned}
&\underset{\boldsymbol{x},\, z}{\text{minimize}} && z && (Q_k) \\
&\text{subject to} && \frac{f_i(\boldsymbol{x}) - \theta^k g_i(\boldsymbol{x})}{g_i(\boldsymbol{x}^k)} \leqq z, \ \ i = 1, \ldots, r.
\end{aligned}
$$

An Experimental Result

A result of our experiments is shown below: The problem is to decide a bond portfolio among 37 bonds selected from the market. The holding bonds are $x(1) = 5{,}000$, $x(9) = 1{,}000$, $x(13) = 2{,}500$, $x(17) = 4{,}500$, $x(19) = 5{,}500$, $x(21) = 6{,}000$, $x(23) = 5{,}200$, $x(25) = 4{,}200$, $x(27) = 3{,}200$ and $x(37) = 3{,}800$. The experiment was performed by a worker of a security company in Japan who has a career of acting as a bond trader.

	Pareto sol.	Asp. level (target range)	Lowest	Highest	Sensitivity
F1 (max)	5.8335	5.8000	5.4788	5.9655	0.0133
F2 (max)	6.8482	6.8000	6.7165	6.8698	0.0000
F3 (min)	0.1292	0.1300	0.1262	0.1325	1.0000
$*$ F4	102.7157	$\mathrm{F4} \leq 103.00$			
$*$ F5	4.0000	$4.00 \leq \mathrm{F5} \leq 5.00$			
$*$ F6	0.2000	$0.20 \leq \mathrm{F6}$			
$*$ F7	0.2000	$0.20 \leq \mathrm{F7}$			

x(1)=	1,042.0643	x(2)=	400.0000
x(3)=	400.0000	x(4)=	200.0000
x(5)=	0.0000	x(6)=	400.0000
x(7)=	0.0000	x(8)=	0.0000
x(9)=	547.3548	x(10)=	0.0000
x(11)=	0.0000	x(12)=	0.0000
x(13)=	2,500.0000	x(14)=	200.0000
x(15)=	0.0000	x(16)=	200.0000
x(17)=	4,500.0000	x(18)=	0.0000
x(19)=	5,321.9573	x(20)=	0.0000
x(21)=	6,000.0000	x(22)=	0.0000
x(23)=	5,200.0000	x(24)=	0.0000
x(25)=	4,200.0000	x(26)=	0.0000
x(27)=	3,200.0000	x(28)=	274.6025
x(29)=	400.0000	x(30)=	200.0000
x(31)=	0.0000	x(32)=	200.0000
x(33)=	400.0000	x(34)=	400.0000
x(35)=	0.0000	x(36)=	0.0000
x(37)=	3,800.0000		

The asterisk of F4–F7 implies soft constraints. In this system, we can change objective function into soft constraints and vice versa. Here, the bond trader changed F2 (effective yield) into a soft constraint, and F4 (unit price) into an objective function. Then under the modified aspiration level by trade-off, the obtained result is as follows:

	Pareto sol.	Asp. level (target range)	Lowest	Highest	Sensitivity
F1 (max)	5.8629	5.9000	5.8193	5.9608	0.0000
* F2 (max)	6.8482	$6.8482 \leq F2$			
F3 (min)	0.1302	0.1292	0.1291	0.1322	1.0000
F4 (min)	102.3555	102.1000	102.0676	102.7228	0.0043
* F5	4.0000	$4.00 \leq F5 \leq$ 5.00			
* F6	0.2000	$0.20 \leq F6$			
* F7	0.2000	$0.20 \leq F7$			

x(1)= 0.0000	x(2)= 400.0000
x(3)= 400.0000	x(4)= 200.0000
x(5)= 0.0000	x(6)= 400.0000
x(7)= 0.0000	x(8)= 0.0000
x(9)= 139.4913	x(10)= 0.0000
x(11)= 0.0000	x(12)= 0.0000
x(13)= 2,500.0000	x(14)= 381.5260
x(15)= 0.0000	x(16)= 200.0000
x(17)= 4,500.0000	x(18)= 0.0000
x(19)= 5,277.1577	x(20)= 0.0000
x(21)= 6,000.0000	x(22)= 0.0000
x(23)= 5,200.0000	x(24)= 0.0000
x(25)= 4,200.0000	x(26)= 0.0000
x(27)= 3,200.0000	x(28)= 14.8920
x(29)= 400.0000	x(30)= 200.0000
x(31)= 0.0000	x(32)= 200.0000
x(33)= 400.0000	x(34)= 400.0000
x(35)= 400.0000	x(36)= 222.7743
x(37)= 3,800.0000	

2.5 Some Remarks on Applications

Decision making is a problem of value judgment. One of most important tasks in multiobjective optimization is how to treat the value judgment of decision maker (DM). In order to get a solution reflecting faithfully the value judgment of DM in a flexible manner for the multiplicity of value judgment and complex changes of environment of decision making, cooperative systems of man and computers are very attractive: above all, interactive multiobjective programming methods seem promising.

Among several interactive multiobjective programming techniques, the aspiration level approach has been applied to several kinds of real problems, because:

1. It does not require any consistency of judgment of DM.
2. It reflects the value of DM very well.
3. It is easy to implement.

In particular, the point (1) is very important, because DM tends to change his attitude even during the decision making process. This implies that the aspiration level approach such as the satisficing trade-off method can work well not only for the multiplicity of value judgment of DMs but also for the dynamics of value judgment of them.

Even in cases with a large number of objective functions, the aspiration level approach works well, because aspiration levels are intuitive for DM and reflect the value of DM very well. Using graphic user interface, DM can input the aspiration level very easily, and decrease the burden by using automatic (or, exact) trade-off method as was stated in previous section.

Trade-off analysis is relatively easily made if we know the whole configuration of Pareto frontier. At least in a neighborhood of the obtained Pareto solution, the information on the configuration of Pareto frontier is needed for trade-off analysis. For many smooth cases under some appropriate conditions, Lagrange multipliers for auxiliary scalarized problems provide those information. However, we cannot utilize Lagrange multiplier as trade-off information in nonsmooth cases such as discrete problems. Evolutionary multiobjective optimization (EMO) methods, which have been studied actively in recent years, work well to generate Pareto frontiers not only in smooth cases but also nonsmooth (even discrete) cases. However, it is in general time consuming to generate the whole of Pareto frontier. In many practical problems, DM is interested in some part of Pareto frontier, but not in the whole. Therefore, it is important to combine the aspiration level approach and EMO methods effectively. This will be discussed in subsequent chapters.

http://www.springer.com/978-3-540-88909-0

Sequential Approximate Multiobjective Optimization
Using Computational Intelligence
Nakayama, H.; Yun, Y.; Yoon, M.
2009, XVI, 200 p. 111 illus., Hardcover
ISBN: 978-3-540-88909-0