Chapter 2
Optimization Concepts and Computational Methods

This chapter is devoted to reviewing optimization concepts and the related computational methods that will be used in the remaining chapters. In particular, we deal with three optimization concepts incorporating fuzziness and ambiguity of human judgements, multiplicity of evaluation criteria, and uncertainty of events characterizing decision making problems. Moreover, we provide the fundamentals of genetic algorithms which are considered to be one of the most practical and proven meta heuristics for difficult classes of optimization problems.

2.1 Fuzzy programming

Zimmermann (1976) introduces the concept of fuzzy set theory into linear programming. Assuming that the membership functions for fuzzy sets are linear, he shows that, by employing the principle of the fuzzy decision by Bellman and Zadeh (1970), a linear programming problem with a fuzzy goal and fuzzy constraints can be solved by using standard linear programming techniques.

A linear programming problem is represented as

$$\begin{aligned}
&\text{minimize} && z(x_1,\ldots,x_n) = c_1x_1 + \cdots + c_nx_n && \text{(2.1a)}\\
&\text{subject to} && a_{11}x_1 + \cdots + a_{1n}x_n \leqq b_1 && \text{(2.1b)}\\
& && \cdots\cdots\cdots && \\
& && a_{m1}x_1 + \cdots + a_{mn}x_n \leqq b_m && \text{(2.1c)}\\
& && x_j \geqq 0, \quad j = 1,\ldots,n, && \text{(2.1d)}
\end{aligned}$$

where x_j is a decision variable, and c_j, a_{ij} and b_i are given coefficients of the objective function and the constraints. Let $\boldsymbol{x} = (x_1,\ldots,x_n)^T$ denote a column vector of the decision variables, and let $\boldsymbol{c} = (c_1,\ldots,c_n)$, $A = [a_{ij}]$, and $\boldsymbol{b} = (b_1,\ldots,b_m)^T$ denote an n-dimensional row vector of the coefficients of the objective function, an $m \times n$ matrix of the coefficients of the left-hand side of the constraints, and an m-dimensional

M. Sakawa and I. Nishizaki (eds.), *Cooperative and Noncooperative Multi-Level Programming*, 11
Operations Research/Computer Science Interfaces Series 48, DOI: 10.1007/978-1-4419-0676-2_2,

column vector of the coefficients of the right-hand side of the constraints, respectively; the superscript T means transposition of a vector or a matrix. Then, problem (2.1) is simply rewritten in the following vector and matrix representation:

$$\text{minimize } z(\boldsymbol{x}) = \boldsymbol{cx} \tag{2.2a}$$
$$\text{subject to } A\boldsymbol{x} \leqq \boldsymbol{b} \tag{2.2b}$$
$$\boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.2c}$$

To a standard linear programming problem (2.2), taking into account the imprecision or fuzziness of a decision maker's judgment, Zimmermann considers the following linear programming problem with a fuzzy goal and fuzzy constraints.

$$\boldsymbol{cx} \lesssim z_0 \tag{2.3a}$$
$$A\boldsymbol{x} \lesssim \boldsymbol{b} \tag{2.3b}$$
$$\boldsymbol{x} \geqq \boldsymbol{0}, \tag{2.3c}$$

where the symbol $\lesssim$ denotes a relaxed or fuzzy version of the ordinary inequality $\leqq$. From the decision maker's preference, the fuzzy goal (2.3a) and the fuzzy constraints (2.3b) mean that the objective function $\boldsymbol{cx}$ should be essentially smaller than or equal to a certain level z_0, and that the values of the constraints $A\boldsymbol{x}$ should be essentially smaller than or equal to $\boldsymbol{b}$, respectively. Assuming that the fuzzy goal and the fuzzy constraints are equally important, he employs the following unified formulation.

$$B\boldsymbol{x} \lesssim \boldsymbol{b}' \tag{2.4a}$$
$$\boldsymbol{x} \geqq \boldsymbol{0}, \tag{2.4b}$$

where

$$B = \begin{bmatrix} \boldsymbol{c} \\ A \end{bmatrix}, \quad \boldsymbol{b}' = \begin{bmatrix} z_0 \\ \boldsymbol{b} \end{bmatrix}. \tag{2.5}$$

To express the imprecision or fuzziness of the decision maker's judgment, the ith fuzzy constraint $(B\boldsymbol{x})_i \lesssim b_i'$ is interpreted as the following linear membership function:

$$\mu_i((B\boldsymbol{x})_i) = \begin{cases} 1, & \text{if } (B\boldsymbol{x})_i \leqq b_i' \\ 1 - \dfrac{(B\boldsymbol{x})_i - b_i'}{d_i}, & \text{if } b_i' < (B\boldsymbol{x})_i \leqq b_i' + d_i \\ 0, & \text{if } (B\boldsymbol{x})_i > b_i' + d_i, \end{cases} \tag{2.6}$$

where d_i is a subjectively specified constant expressing the limit of the admissible violation of the ith constraint, and it is depicted in Figure 2.1.

On the basis of the principle of the fuzzy decision by Bellman and Zadeh (1970), the problem of finding the maximum decision is represented as

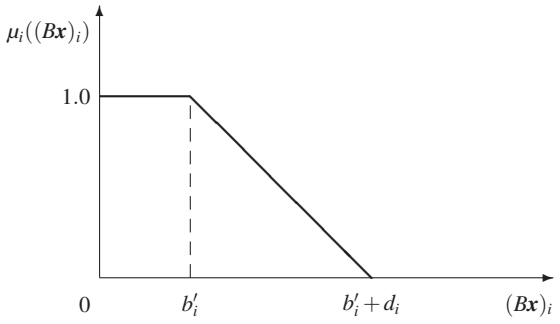


Fig. 2.1 Linear membership function for the ith fuzzy constraint.

$$\text{maximize} \quad \min_{0 \leqq i \leqq m+1} \mu_i((B\boldsymbol{x})_i) \tag{2.7a}$$

$$\text{subject to} \quad \boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.7b}$$

With the variable transformation $b''_i = b'_i/d_i$ and $(B'\boldsymbol{x})_i = (B\boldsymbol{x})_i/d_i$, and an auxiliary variable λ, problem (2.7) can be transformed into the following conventional linear programming problem:

$$\text{maximize} \quad \lambda \tag{2.8a}$$

$$\text{subject to} \quad \lambda \leqq 1 + b''_i - (B'\boldsymbol{x})_i, \ i = 0, \ldots, m+1 \tag{2.8b}$$

$$\boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.8c}$$

Because the fuzzy decision is represented as $\min_{0 \leqq i \leqq m+1} \mu_i((B\boldsymbol{x})_i)$, it is often called the minimum operator.

2.2 Multiobjective programming

2.2.1 Multiobjective programming problem

A problem to optimize multiple conflicting linear objective functions simultaneously under a given linear constraints is called a multiobjective linear programming problem. Let $\boldsymbol{c}_i = (c_{i1}, \ldots, c_{in})$, $i = 1, \ldots, k$ denote a vector of coefficients of the i objective function. Then, the multiobjective linear programming problem is represented as

$$\begin{aligned}
&\text{minimize} && z_1(\boldsymbol{x}) = c_{11}x_1 + \cdots + c_{1n}x_n && \text{(2.9a)}\\
&&& \cdots\cdots\cdots\\
&\text{minimize} && z_k(\boldsymbol{x}) = c_{k1}x_1 + \cdots + c_{kn}x_n && \text{(2.9b)}\\
&\text{subject to} && a_{11}x_1 + \cdots + a_{1n}x_n \leqq b_1 && \text{(2.9c)}\\
&&& \cdots\cdots\cdots\\
&&& a_{m1}x_1 + \cdots + a_{mn}x_n \leqq b_m && \text{(2.9d)}\\
&&& x_j \geqq 0, \quad j = 1, \ldots, n. && \text{(2.9e)}
\end{aligned}$$

Alternatively, it is expressed by

$$\begin{aligned}
&\text{minimize} && \boldsymbol{z}(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{x} && \text{(2.10a)}\\
&\text{subject to} && \boldsymbol{A}\boldsymbol{x} \leqq \boldsymbol{b} && \text{(2.10b)}\\
&&& \boldsymbol{x} \geqq \boldsymbol{0}. && \text{(2.10c)}
\end{aligned}$$

Following the convention in the literature of multiobjective optimization, we introduce the following binary relations:

$$\begin{aligned}
&\boldsymbol{x} = \boldsymbol{y} \Leftrightarrow x_i = y_i,\ i = 1, \ldots, k && \text{(2.11a)}\\
&\boldsymbol{x} \geqq \boldsymbol{y} \Leftrightarrow x_i \geqq y_i,\ i = 1, \ldots, k && \text{(2.11b)}\\
&\boldsymbol{x} \geq \boldsymbol{y} \Leftrightarrow x_i \geqq y_i,\ i = 1, \ldots, k, \text{ and } \boldsymbol{x} \neq \boldsymbol{y} && \text{(2.11c)}\\
&\boldsymbol{x} > \boldsymbol{y} \Leftrightarrow x_i > y_i,\ i = 1, \ldots, k. && \text{(2.11d)}
\end{aligned}$$

First, we give the notion of optimality in a multiobjective linear programming problem. Because there does not always exist a solution minimizing all of the objective functions simultaneously, a solution concept of Pareto optimality plays an important role in multiobjective optimization, and it is defined as follows. Let S denote the nonempty set of all feasible solutions of problem (2.10), i.e., $S \triangleq \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}\}$.

Definition 2.1. A point $\boldsymbol{x}^*$ is said to be a *Pareto optimal solution* if and only if there does not exist another $\boldsymbol{x} \in S$ such that $\boldsymbol{z}(\boldsymbol{x}) \leq \boldsymbol{z}(\boldsymbol{x}^*)$.

By substituting the strict inequality $<$ for the inequality $\leq$ in Definition 2.1, weak Pareto optimality is defined as a slightly weaker solution concept.

2.2.2 Interactive multiobjective programming

As seen from Definition 2.1, in general there exist a infinite number of Pareto optimal solutions if the feasible region S is not empty. In real-world decision making problems, to make a reasonable decision or implement a desirable scheme, the decision maker should select one point among the set of Pareto optimal solutions. For this end, several interactive multiobjective programming methods were developed

from the 1970s to the 1980s, and it is known that the reference point method developed by Wierzbicki (1979) is relatively practical.

For each of the objective functions $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), \ldots, z_k(\boldsymbol{x}))^T$ in problem (2.10), the decision maker specifies the reference point $\bar{\boldsymbol{z}} = (\bar{z}_1, \ldots, \bar{z}_k)^T$ which reflects the desired values of the objective functions, and it is thought that by changing the reference points in the interactive solution procedure, the decision maker can perceive, understand and learn the decision maker's own preference.

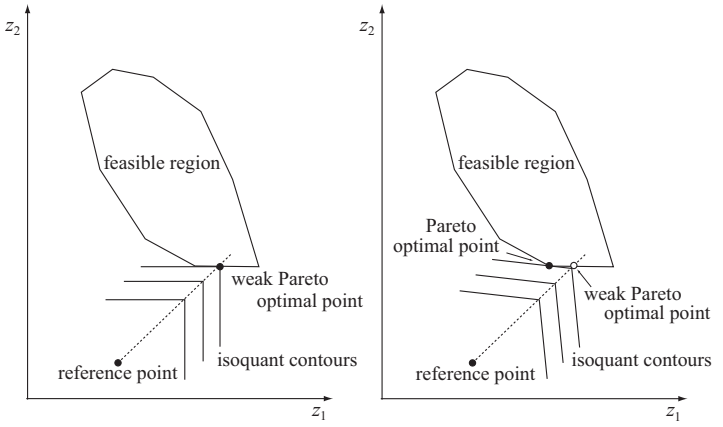


Fig. 2.2 Reference point method.

After the reference point $\bar{\boldsymbol{z}}$ is specified, the following minimax problem is solved:

$$\text{minimize} \quad \max_{i=1,\ldots,k} \{z_i(\boldsymbol{x}) - \bar{z}_i\} \tag{2.12a}$$

$$\text{subject to} \quad A\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.12b}$$

An optimal solution to problem (2.12) is a Pareto optimal solution closest to the reference point in the L_∞ norm; the L_∞ norm is also called the Tchebyshev norm or the Manhattan distance. Introducing an auxiliary variable v, problem (2.12) is equivalently expressed as follows:

$$\text{minimize} \quad v \tag{2.13a}$$

$$\text{subject to} \quad z_i(\boldsymbol{x}) - \bar{z}_i \leqq v,\ i = 1, \ldots, k \tag{2.13b}$$

$$A\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.13c}$$

Because the isoquant contour of the objective function $\max_{i=1,\ldots,k}\{z_i(\boldsymbol{x}) - \bar{z}_i\}$ is orthogonal, there is a possibility that an optimal solution to problem (2.13) is not a Pareto optimal solution but a weak a Pareto optimal solution due to a location of the reference point and the shape of the feasible region as seen in the left-hand graph

of Figure (2.2). Let ρ be a given small positive number. By adding the augmented term $\rho \sum_{i=1}^{k} (z_i(\boldsymbol{x}) - \bar{z}_i)$ to the objective function, the isoquant contour of the revised objective function has an obtuse angle as seen in the right-hand graph of Figure (2.2). From this fact, a Pareto optimal solution to multiobjective linear programming problem (2.10), which is closest to the reference point $\bar{\boldsymbol{z}}$, can be obtained by solving the following revised problem:

$$\text{minimize} \quad v + \rho \sum_{i=1}^{k} (z_i(\boldsymbol{x}) - \bar{z}_i) \tag{2.14a}$$

$$\text{subject to} \quad z_i(\boldsymbol{x}) - \bar{z}_i \leqq v, \ i = 1, \ldots, k \tag{2.14b}$$

$$A\boldsymbol{x} \leqq \boldsymbol{b}, \ \boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.14c}$$

2.2.3 Fuzzy multiobjective programming

To multiobjective linear programming problem (2.10), Zimmermann (1978) extends the fuzzy programming shown in the previous section by introducing fuzzy goals for all the objective functions. Assuming that the decision maker has a fuzzy goal for each of the objective functions, the corresponding linear membership function is defined as

$$\mu_i(z_i(\boldsymbol{x})) = \begin{cases} 1, & \text{if } z_i(\boldsymbol{x}) \leqq z_i^1 \\ \dfrac{z_i(\boldsymbol{x}) - z_i^0}{z_i^1 - z_i^0}, & \text{if } z_i^1 < z_i(\boldsymbol{x}) \leqq z_i^0 \\ 0, & \text{if } z_i(\boldsymbol{x}) > z_i^0, \end{cases} \tag{2.15}$$

where z_i^0 and z_i^1 denote the values of the ith objective function $z_i(\boldsymbol{x})$ such that the degrees of membership function are 0 and 1, respectively, and it is depicted in Figure 2.3.

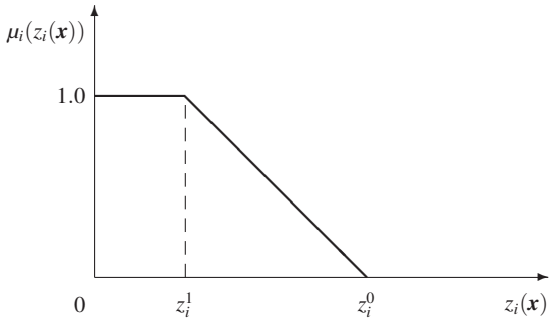


Fig. 2.3 Linear membership function for the ith fuzzy goal.

Following the principle of the fuzzy decision by Bellman and Zadeh (1970), the multiobjective linear programming problem (2.10) can be interpreted as the following maximin problem:

$$\text{maximize} \quad \min_{i=1,\ldots,k} \{\mu_i(z_i(\boldsymbol{x}))\} \tag{2.16a}$$

$$\text{subject to} \quad A\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.16b}$$

Problem (2.16) is equivalently rewritten as a conventional linear programming problem:

$$\text{maximize} \quad \lambda \tag{2.17a}$$

$$\text{subject to} \quad \mu_i(z_i(\boldsymbol{x})) \geqq \lambda,\ i = 1,\ldots,k \tag{2.17b}$$

$$A\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}, \tag{2.17c}$$

where λ is an auxiliary variable.

Sakawa, Yano and Yumine (1987) propose an interactive method for a multiobjective linear programming problem with fuzzy goals. After identifying the membership functions $\mu_i(z_i(\boldsymbol{x}))$, $i = 1,\ldots,k$ for the fuzzy goals of the objective functions $z_i(\boldsymbol{x})$, $i = 1,\ldots,k$, the decision maker is asked to specify the reference membership values which are the aspiration levels of achievement for the values of the membership functions. It follows that the reference membership value is a natural extension of the reference point in the reference point method by Wierzbicki (1979).

Let $\bar{\boldsymbol{\mu}} = (\bar{\mu_1},\ldots,\bar{\mu_k})^T$ denote the reference membership values for the membership functions $\boldsymbol{\mu}(\boldsymbol{z}(\boldsymbol{x})) = (\mu_1(z_1(\boldsymbol{x})),\ldots,\mu_k(z_k(\boldsymbol{x})))^T$. Then, by solving the minimax problem

$$\text{minimize} \quad \max_{i=1,\ldots,k} \{\bar{\mu}_i - \mu_i(z_i(\boldsymbol{x}))\} \tag{2.18a}$$

$$\text{subject to} \quad A\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}, \tag{2.18b}$$

a Pareto optimal solution closest to the vector of the reference membership values in the L_∞ norm can be obtain.

Problem (2.18) is equivalently expressed as follows:

$$\text{minimize} \quad v \tag{2.19a}$$

$$\text{subject to} \quad \bar{\mu}_i - \mu_i(z_i(\boldsymbol{x})) \leqq v,\ i = 1,\ldots,k \tag{2.19b}$$

$$A\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}. \tag{2.19c}$$

2.3 Stochastic programming

When some elements governed by stochastic events are included in the constraints of a linear programming problem, such constraints are difficult to completely hold. To formulate deterministic problems from the problems with random variables,

Charnes and Cooper (1963) propose chance constraint programming which admits random data variations and permits constraint violations up to specified probability limits.

Let $\tilde{A}$ and $\tilde{\boldsymbol{b}}$ denote an $m \times n$ matrix and an m-dimensional column vector of the coefficients of the left-hand side and the right-hand side of the constraints, respectively; and suppose that some or all of the elements of $\tilde{A}$ and $\tilde{\boldsymbol{b}}$ are random variables. Then, a chance constraint formulation for the constraint $\tilde{A}\boldsymbol{x} \leqq \tilde{\boldsymbol{b}}$ of a linear programming problem is represented as

$$P[\tilde{A}\boldsymbol{x} \leqq \tilde{\boldsymbol{b}}] \geqq \boldsymbol{\alpha}, \tag{2.20}$$

where P means a probability measure. The vector $\boldsymbol{\alpha}$ are probabilities of the extents to which constraint violations are admitted. Then, the element α_i is associated with the ith constraint $\sum_{j=1}^{n} a_{ij}x_j \leqq b_i$, and the ith constraint is interpreted as follows:

$$P\left[\sum_{j=1}^{n} \tilde{a}_{ij}x_j \leqq \tilde{b}_i\right] \geqq \alpha_i. \tag{2.21}$$

Inequality (2.21) means that the ith constraint may be violated, but at most $\beta_i = 1 - \alpha_i$ proportion of the time.

First, assume that only $\tilde{b}_i$ in the right-hand side of the chance constraint condition (2.21) is a random variable and $\tilde{a}_{ij}$ is a constant, i.e, $\tilde{a}_{ij} = a_{ij}$. Let $F_i(\tau)$ denote its probability distribution function. From the fact that

$$P\left(\sum_{j=1}^{n} a_{ij}x_j \leqq \tilde{b}_i\right) = 1 - F_i\left(\sum_{j=1}^{n} a_{ij}x_j\right),$$

the chance constraint condition (2.21) can be rewritten as

$$F_i\left(\sum_{j=1}^{n} a_{ij}x_j\right) \leqq 1 - \alpha_i. \tag{2.22}$$

Let $K_{1-\alpha_i}$ denote the maximum of τ such that $\tau = F_i^{-1}(1-\alpha_i)$, and then inequality (2.22) can be simply expressed as

$$\sum_{j=1}^{n} a_{ij}x_j \leqq K_{1-\alpha_i}. \tag{2.23}$$

Second, consider a more general case where not only $\tilde{b}_i$ but also $\tilde{a}_{ij}$ in the left-hand side of (2.21) are random variables, and particularly we assume that $\tilde{b}_i$ and $\tilde{a}_{ij}$ are normal random variables. Let $\mu_{\tilde{b}_i}$ and $\sigma_{\tilde{b}_i}$ be the mean and the variance of $\tilde{b}_i$, and let $\mu_{\tilde{a}_{ij}}$ and V_{ij} be the mean and the variance-covariance matrix of $\tilde{a}_{ij}$. Moreover, assume that $\tilde{b}_i$ and $\tilde{a}_{ij}$ are independent. Then, the chance constraint condition (2.21) can be transformed into

$$\sum_{j=1}^{n} \mu_{\tilde{a}_{ij}} x_j - \bar{K}_{1-\alpha_i} \sqrt{\sigma_{\tilde{b}_i}^2 + \boldsymbol{x}^T V \boldsymbol{x}} \leqq \mu_{\tilde{b}_i}, \tag{2.24}$$

where $\bar{K}_{1-\alpha_i} = F_N^{-1}(1-\alpha_i)$ and $F_N(\cdot)$ is the standardized normal distribution function with parameter $(0,1)$.

Charnes and Cooper (1963) also consider three kinds of decision rules for optimizing objective functions with random variables: (i) the minimum or maximum expected value model, (ii) the minimum variance model, and (iii) the maximum probability model, which are referred to as the E-model, the V-model, and the P-model, respectively. Moreover, Kataoka (1963) and Geoffrion (1967) individually propose the fractile criterion model.

Let $\tilde{\boldsymbol{c}} = (\tilde{c}_1, \ldots, \tilde{c}_n)$ denote an n-dimensional row vector of the coefficients of the objective function, and suppose that some or all of coefficients $\tilde{c}_j$, $j = 1, \ldots, n$ are random variables. Then, the objective function in the E-model is represented as

$$E[\tilde{\boldsymbol{c}}\boldsymbol{x}] = E\left[\sum_{j=1}^{n} \tilde{c}_j x_j\right], \tag{2.25}$$

where E means the function of expectation. Let m_i denote the mean value of $\tilde{c}_j$. Then, the objective function of the E-model can be transformed to

$$E\left[\sum_{j=1}^{n} \tilde{c}_j x_j\right] = \sum_{j=1}^{n} m_j x_j. \tag{2.26}$$

The realization value of the objective function may vary quite widely even if the expected value of the objective function is minimized. In such a case, it may be suspicious if a plan based on the solution of the E-model would work well because uncertainty is large. Some decision makers would prefer to plans with lower uncertainty. The objective function in the V-model is represented as

$$Var[\tilde{\boldsymbol{c}}\boldsymbol{x}] = Var\left[\sum_{j=1}^{n} \tilde{c}_j x_j\right], \tag{2.27}$$

where Var means the function of variance. Let V denote an $n \times n$ variance-covariance matrix for the vector of the random variables $\tilde{\boldsymbol{c}}$, then the objective function of the V-model can be transformed into

$$Var\left[\sum_{j=1}^{n} \tilde{c}_j x_j\right] = \boldsymbol{x}^T V \boldsymbol{x}. \tag{2.28}$$

In the P-model, the probability that the objective function value is smaller than a ceratin target value is maximized, and then the objective function of the P-model is represented as

$$P[\tilde{\boldsymbol{c}}\boldsymbol{x} \leqq f_0], \tag{2.29}$$

where f_0 is a given target value for the objective function.

The fractile criterion model is considered as complementary to the P-model; a target variable to the objective function is minimized after the probability that the objective function value is smaller than the target variable is guaranteed to be larger than a given assured level. Then, the objective function of the fractile criterion model is represented as

$$f \text{ subject to } P[\tilde{\boldsymbol{c}}\boldsymbol{x} \leqq f] \geqq \alpha, \tag{2.30}$$

where f and α are the target variable to the objective function and the given assured level for the probability that the objective function value is smaller than the target variable.

2.4 Genetic algorithms

It is hard to obtain exact optimal solutions of difficult classes of optimization problems such as combinatorial problems and nonconvex nonlinear problems, and thus it is quite natural for decision makers to require approximate optimal solutions instead. To meet this demand, recently several meta-heuristics have been developed and their effectiveness is demonstrated. Among them, genetic algorithms are known to be one of the most practical and proven methods. A computational framework of genetic algorithms initiated by Holland (1975) has been attracted attention of many researchers with applicability in optimization, as well as in search and learning. Furthermore, publications of books by Goldberg (1989) and Michalewicz (1996) bring heightened and increasing interests in applications of genetic algorithms to complex function optimization.

The fundamental procedure of genetic algorithms is shown as a flowchart in Figure 2.4, and it is summarized as follows:

Step 0: Initialization Generate a given number of individuals randomly to form the initial population.

Step 1: Evaluation Calculate the fitness value of each individual in the population.

Step 2: Reproduction According to the fitness values and a reproduction rule specified in advance, select individuals from the current population to form the next population.

Step 3: Crossover Select two individuals randomly from the population, and exchange some part of the string of one individual for the corresponding part of the other individual with a given probability for crossover.

Step 4: Mutation Alter one or more genes in the string of an individual with a given probability of mutation.

Step 5: Termination Stop the procedure if the condition of termination is satisfied, and an individual with the maximum fitness value is determined as an approximate optimal solution. Otherwise, go to Step 1.

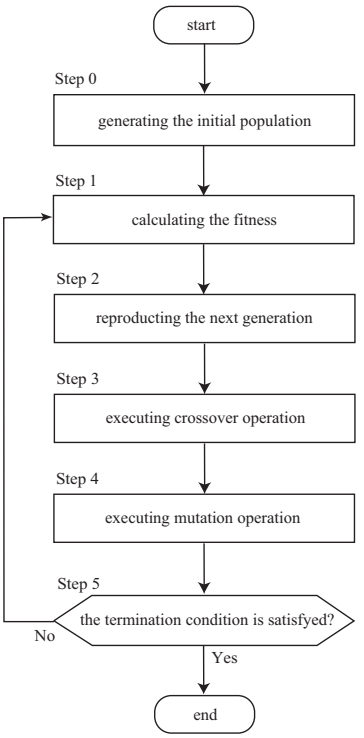


Fig. 2.4 Flowchart of genetic algorithms.

Representation of individuals

When genetic algorithms are applied to optimization problems, a vector of decision variables corresponds to an individual in the population, which is represented by a string as in Figure 2.5.

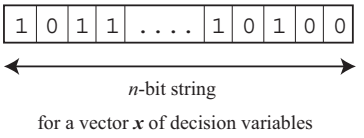


Fig. 2.5 Individual represented by a string.

As seen in Figure 2.5, each element of the string is either 1 or 0 usually, but real numbers, integers, alphabets, or some other symbols can also be used to represent individuals.

Let $\boldsymbol{s}$ and $\boldsymbol{x}$ denote an individual represented by a string and a vector of the decision variables, respectively. The string $\boldsymbol{s}$ which means a chromosome in the context of biology is called the genotype of an individual, and the decision variables $\boldsymbol{x}$ is called the phenotype. The mapping from phenotypes to genotypes is called coding, and the reverse mapping is called decoding.

Fitness function and scaling

In an optimization problem where the objective function is minimized or maximized, a solution with the lowest objective function value or the highest objective function value is searched. When genetic algorithms are applied to solving an optimization problem, a solution to the optimization problem is associated with an individual in the genetic algorithm, and the objective function value of the solution corresponds to the fitness of the individual. Thus, an individual with a higher fitness value has a higher probability of surviving in the next generation.

Let $z(\boldsymbol{x})$ denote an objective function to be minimized in an optimization problem. The corresponding fitness function in genetic algorithms is commonly defined as (Goldberg, 1989)

$$f(\boldsymbol{s}_i) = \begin{cases} C_{\max} - z(\boldsymbol{x}) & \text{if } z(\boldsymbol{x}) < C_{\max} \\ 0 & \text{otherwise,} \end{cases} \tag{2.31}$$

where $\boldsymbol{s}_i$ denotes the ith individual in the population, and $C_{\max}$ is a given constant. For example, the value of $C_{\max}$ is determined as the largest objective function value $z(\boldsymbol{x})$ observed thus far, the largest value $z(\boldsymbol{x})$ in the current population, or the largest value $z(\boldsymbol{x})$ in the last t generations. Similarly, in maximization problems, to prevent the fitness value from being negative, the constant $C_{\min}$ is introduced, and the following fitness function is often used:

$$f(\boldsymbol{s}_i) = \begin{cases} z(\boldsymbol{x}) + C_{\min} & \text{if } z(\boldsymbol{x}) + C_{\min} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.32}$$

For example, the value of $C_{\min}$ is determined as the absolute value of the smallest $z(\boldsymbol{x})$ in the current population or in the last t generations.

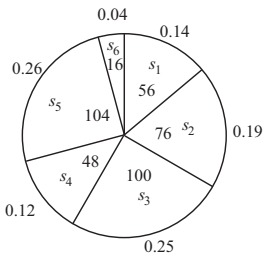
To properly distribute fitness values in the population, fitness scaling is employed. The linear scaling, which is a simple and useful procedure, is represented by

$$f'(\boldsymbol{s}_i) = af(\boldsymbol{s}_i) + b, \tag{2.33}$$

where f and f' are the raw fitness value and the scaled fitness value, respectively; a and b are coefficients. To perform the operation of reproduction appropriately, the coefficients a and b may be chosen in such a way that the average scaled fitness value f'_{ave} is equal to the average raw fitness value f_{ave}, and the maximum scaled fitness value is determined as $f'_{\max} = C_{\text{mult}} f_{\text{ave}}$, where C_{mult} is a given constant.

Genetic operators

The three genetic operators, reproduction, crossover, and mutation, are outlined below. Individuals are copied into the next generation according to their fitness values by a reproduction operator. The roulette wheel selection is one of the most popular reproduction operators, and in this method, each individual in the current population has a roulette wheel slot sized in proportion to its fitness value. Let *pop_size* be the number of individuals in the population. The percentage of the roulette wheel given to an individual $\boldsymbol{s}_i$ is $100f(\boldsymbol{s}_i)/\sum_{l=1}^{pop\_size} f(\boldsymbol{s}_l)\%$. Namely, the individual $\boldsymbol{s}_i$ is reproduced with the probability $p(\boldsymbol{s}_i) = f(\boldsymbol{s}_i)/\sum_{l=1}^{pop\_size} f(\boldsymbol{s}_l)$ each spin of the roulette wheel. An example of the roulette wheel is given in Figure 2.6; the numbers in the wheel are fitness values of individuals, and the decimal numbers outside of the wheel are the corresponding probabilities.



	s_1	s_2	s_3	s_4	s_5	s_6
$p(s_i)$	0.14	0.19	0.25	0.12	0.26	0.04
$f(s_i)$	56	76	100	48	104	16

Fig. 2.6 Biased roulette wheel.

Crossover creates offsprings into the next population by combining the genetic material of two parents. The variation caused by the crossover process may bring offsprings better fitness values, and thus it is thought that crossover plays an important roll in genetic algorithms. Although there are many different types of crossover, we provide a simple example here: a single-point crossover operator is the most simple operator of crossover. In this operation, two parent strings $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are randomly chosen from a mating pool in which newly reproduced individuals are entered temporarily, and then one crossover point in the strings is chosen at random. Two offsprings are made by exchanging the substrings which are parts of the left side of the parent strings $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ from the crossover point. The crossover operation is illustrated in Figure 2.7.

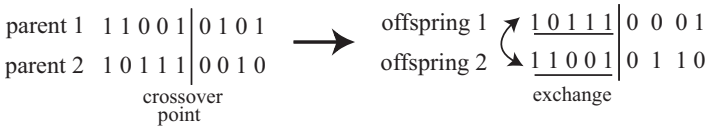


Fig. 2.7 Crossover operation.

With a small probability, an operation of mutation provides the string of an individual with a randomly tiny alteration, and it is recognized that mutation serves as local search. In the representation of the 0-1 bit strings, mutation means changing a 1 to a 0 and vice versa. A simple version of mutation operator is illustrated in Figure 2.8.

Fig. 2.8 Mutation operation.

http://www.springer.com/978-1-4419-0675-5

Cooperative and Noncooperative Multi-Level Programming

Sakawa, M.; Nishizaki, I.

2009, X, 250 p. 19 illus., Hardcover
ISBN: 978-1-4419-0675-5