

# Chapter 2

## Basics for Optimization Problem



### 2.1 Overview of Optimization Problems

#### 2.1.1 General Forms

In different engineering scenarios, the maximizing or minimizing of some functions relative to some sets are common problems. The corresponding set often represents a range of choices available in a certain situation, and the “solution” infers the “best” or “optimal” choices in this scenario. Some common applications include “minimal cost, maximal profit, minimal error, optimal design, and optimal management”. This type of problem has a general form as follows.

$$\begin{aligned} & \min_x / \max_x f(x) \\ & \text{s.t. } h(x) \leq 0 \\ & g(x) = 0, \forall x \in S \end{aligned} \tag{2.1}$$

In (2.1),  $f(x)$  is the objective function, and represents the management tasks; “min” and “max” represent the minimizing and maximizing of  $f(x)$ , respectively;  $x$  is the decision variables, and represents the choices of administrator;  $h(x) \leq 0$  and  $g(x) = 0$  are the inequality and equality constraints to limit the decision variables, which represents the limitations on the choices of administrator by different operating scenarios;  $S$  is the original set for the decision variables, such as continuous variables, binary variables, integer variables, and so on. In this problem, the model expects to find the “best” or “optimal” solution “ $x^*$ ” which meets the minimization or maximization of  $f(x)$ , and in reality, this may represent the minimization of costs or the maximization of profits. Here we give a simple case, Example 2.1, for the optimization problems.

**Example 2.1: Knapsack Problem**

Assuming we have  $n$  types of goods and indexed by  $i \in 1, 2, 3 \dots, n$ . Each good values  $W_i$  and the size is  $S_i$ , and we have a knapsack with the capacity of  $C$ . The problem is how we can pack the goods with the highest value? The model of this problem is shown as follows.

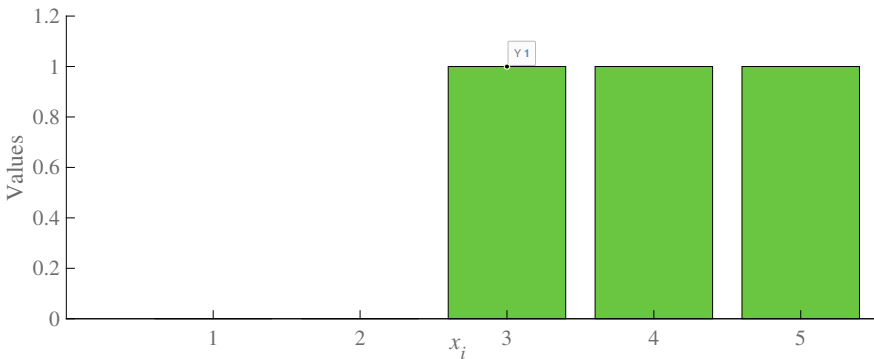
$$\begin{aligned} & \max \sum_{i=1}^n W_i \cdot x_i \\ & s.t. \sum_{i=1}^n S_i \cdot x_i \leq C \\ & \quad x_i \in \{0, 1\} \end{aligned} \tag{2.2}$$

In (2.2),  $x_i$  is the “decision variables”, and represents the choice of the  $i$ th good or not. If choosing the  $i$ th good into the knapsack,  $x_i = 1$  and if not,  $x_i = 0$ .  $x_i \in \{0, 1\}$  is the original set of “decision variables”.  $\sum_{i=1}^n W_i \cdot x_i$  is the “objective function”, and represents the total values of the selected goods, and  $\sum_{i=1}^n S_i \cdot x_i \leq C$  is the “constraint”, represents the total sizes of goods that should be smaller or equal to the capacity of the knapsack. The “best” or “optimal” solution “ $\{x_i^*, i \in 1, 2, 3 \dots, n\}$ ” can achieve the maximization of  $\sum_{i=1}^n W_i \cdot x_i$ .

**Case Study for Example 2.1**

Here we test a simple case, and the parameters are shown as follows:  $n = 5$ ,  $\{W_i | i = 1, 2, \dots, 5\} = [2, 3, 1, 4, 7]$ , and  $\{S_i | i = 1, 2, \dots, 5\} = [2, 2, 1, 2, 3]$ , and  $C = 6$ . The simulation results are shown in Fig. 2.1.

From the Fig. 2.1, the best solution for Example 2.1 is to select the 3rd, 4th and 5th goods, and the maximal total value is 12, and the total size of goods is 6.



**Fig. 2.1** Simulation results of Example 2.1

### 2.1.2 Classifications of Optimization Problems

#### (1) Classifications by decision variables

The decision variable  $x$  may consist of different number types, such as continuous variables, binary variables, and integer variables, and the combination of different number types produces different optimization problems. For example, if  $x$  only consists of continuous variables, then problem (2.1) is “continuous optimization”, and if  $x$  only consists of binary variables or integer variables, then problem (2.1) is “binary optimization” or “integer optimization”. If  $x$  simultaneously has continuous variables and integer variables, then problem (2.1) is “mixed-integer optimization”.

#### (2) Classifications by the objective function

Generally, objective function  $f(x)$  can be a scalar or vector and based on it, problem (2.1) is “single-objective optimization” when  $f(x)$  is a scalar and “multi-objective optimization” or “vector optimization” when  $f(x)$  is a vector.

#### (3) Linear optimization and non-linear optimization

In practical cases,  $f(x)$ ,  $h(x)$  and  $g(x)$  may have different mathematical characteristics. If  $f(x)$ ,  $h(x)$  and  $g(x)$  are all linear functions, problem (2.1) is “linear optimization (LP)”, and is “non-linear optimization (NLP)” if anyone in  $f(x)$ ,  $h(x)$  and  $g(x)$  is non-linear. Specifically, if  $f(x)$  is non-linear, and  $h(x)$  and  $g(x)$  are both linear, then problem (2.1) is “linear constrained and non-linear objective optimization (LCNLP)”, and if  $f(x)$  is linear, and  $h(x)$  and  $g(x)$  are both non-linear, then problem (2.1) is “non-linear constrained and linear objective optimization”. Here we give some typical cases, if  $f(x)$ ,  $h(x)$  and  $g(x)$  are all polynomials and the largest power is two, then problem (2.1) is “quadratic optimization (QP)”. Similarly, we can define the “quadratic-objective quadratic-constrained optimization (QCQP)”, “quadratic-objective linear-constrained optimization (LCQP)”, and so on.

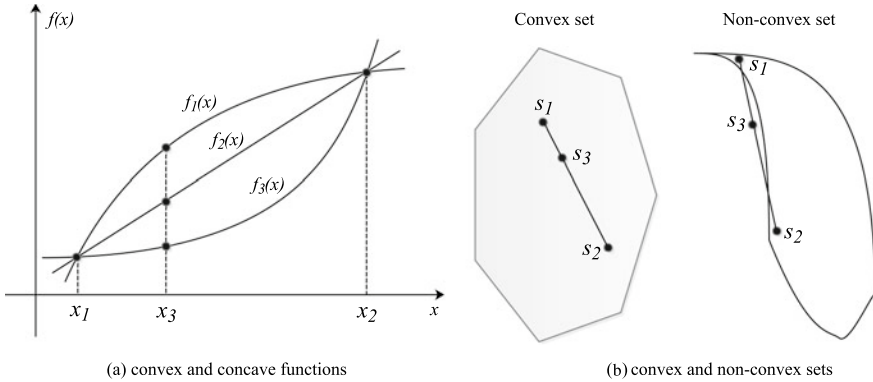
#### (4) Convex optimization and non-convex optimization

Before introducing the convex optimization and non-convex optimization, the convex function and convex set should be described in the first place. Firstly, convex functions should meet (2.3) for any  $x_1$  and  $x_2$  in the domain of  $f(x)$  [1].

$$f(\alpha \cdot x_1 + (1 - \alpha) \cdot x_2) \leq \alpha \cdot f(x_1) + (1 - \alpha) \cdot f(x_2), \forall \alpha \in [0, 1] \quad (2.3)$$

Then the convex set  $S$  should meet: for any two points in  $S$ , denoted as  $s_1$  and  $s_2$ , their linear combination  $\alpha \cdot s_1 + (1 - \alpha) \cdot s_2$  is still within  $S$  [1]. Illustrations for convex function and convex set are shown in Fig. 2.2a and b, respectively.

From Fig. 2.2a,  $x_3 = \alpha \cdot x_1 + (1 - \alpha) \cdot x_2$  and  $f_3(x_3) < \alpha \cdot f_3(x_1) + (1 - \alpha) \cdot f_3(x_2)$ , thus  $f_3(x)$  is a convex function. Similarly,  $f_1(x)$  is a concave function, and  $f_2(x)$  is a convex function and also a concave function. From Fig. 2.2b, any linear combination



**Fig. 2.2** Illustrations of convex functions and convex sets

of  $s_1$  and  $s_2$  belongs to the same set, which represents the convexity. For the non-convex set, at least one combination of  $s_1$  and  $s_2$  is outside the same set, shown in Fig. 2.2b.

With the above definitions, (2.1) is a convex optimization problem when the following two conditions satisfied: (1)  $f(x)$  is convex in case of minimization and concave in case of maximization; (2)  $S = \{x|h(x) \leq 0, g(x) = 0, \forall x \in S\}$  is a convex set. The main characteristic of the convex optimization compared with non-convex optimization is, a local optimal solution of the convex optimization is also the global optimal solution of this convex optimization [1]. This characteristic greatly benefits the applications of convex optimization, and in reality, if we can model or reformulate the problems as convex optimization, then the global optimal solution can be obtained after resolving any local optimal ones. This is one of the main reasons for “the main watershed in optimization problem is not between the linear ones and non-linear ones, but the convex ones and non-convex ones” [1].

In summary, the classification methods can be combined to characterize different optimization problems, such as “mixed-integer linear optimization (MILP)”, “mixed-integer non-linear optimization (MINLP)”, “mixed-integer quadratic optimization (MIQCP)”, and so on.

## 2.2 Optimization Problems with Uncertainties

Uncertainties are inevitable in reality since the measurement and control both have errors. To ensure safety and reliability, considering uncertainties in optimization problems is necessary, and stochastic optimization, robust optimization, and interval optimization are three main types.

### 2.2.1 Stochastic Optimization

A general form of stochastic optimization is shown as (2.4) [2].

$$\min_{x \in X} g(x) + E \left( \min_{y \in Y(x, \xi)} f(y) \right) \quad (2.4)$$

In stochastic optimization (Eq. 2.4),  $x$  is the first stage decision variables which are not determined by uncertainties;  $X$  is the feasible region of  $x$ ;  $g(x)$  is the objective function of the first stage;  $\xi$  is the uncertain variables, and  $Y(x, \xi)$  is the feasible region of  $y$  determined by  $x$  and  $\xi$ ;  $f(y)$  is the objective function of the second stage;  $E(\cdot)$  is the expectation. In this model, the uncertain variable  $\xi$  is depicted by the probability distribution, such as the probability distribution of equipment failure, or the probability distribution of renewable energy output, and so on. Then stochastic optimization seeks the optimal solution within the feasible region defined by the probability distributions. To clearly show the stochastic optimization, Example 2.2 is reformulated as follows.

#### Example 2.2: Stochastic Knapsack Problem

Based on all the assumptions of Example 2.1, we further assume that for  $\forall i \in \{1, 2, \dots, n_f\}$ ,  $W_i$  is a constant, and for  $\forall i \in \{n_f, n_f + 1, \dots, n\}$ ,  $W_i = W_c + \Delta W_i$ , where  $W_c$  is a constant and  $\Delta W_i$  follows a pre-given distribution  $\psi$ . Then the original knapsack problem becomes (2.5).

$$\begin{aligned} \max & \left( \sum_{i=1}^{n_f} W_i \cdot x_i + \sum_{i=n_f}^n W_c \cdot x_i \right) + E \left( \max \left( \sum_{i=n_f}^n \Delta W_i \cdot x_i \right) \right) \\ \text{s.t.} & \sum_{i=1}^n S_i \cdot x_i \leq C \\ & x_i \in \{0, 1\}, \quad \Delta W_i \in \psi \end{aligned} \quad (2.5)$$

where  $\left( \sum_{i=1}^{n_f} W_i \cdot x_i + \sum_{i=n_f}^n W_c \cdot x_i \right)$  is “ $-g(x)$ ”, and the “ $-$ ” is to transform the maximization of (2.5) to the minimization of (2.4), and this term is not influenced by the uncertainties;  $\sum_{i=n_f}^n \Delta W_i \cdot x_i$  is  $f(y)$  which is influenced by the uncertainties; and  $x = \{x_i | i = 1, 2, \dots, n_f\}$ , and  $y = \{x_i | i = n_f, n_f + 1, \dots, n\}$ .

**Case Study for Example 2.2** Here we test a simple case, and the parameters are shown as follows:  $n = 5$ ,  $\{W_i | i = 1, 2, 3\} = [2, 3, 1]$ , and  $\{W_c | i = 4, 5\} = [4, 7]$ , and  $\{\Delta W_i | i = 4, 5\}$  is normally distributed as  $\mathcal{N}(0, 1)$ , and  $\{S_i | i = 1, 2, \dots, 5\} = [2, 2, 1, 2, 3]$ , and  $C = 6$ . The simulation results are shown in Fig. 2.3a and b.

From the Fig. 2.3a, the best solution for Example 2.2 is also to select the 3rd, 4th and 5th goods, and the expected maximal total value is 12.23, and the total size of goods is 6. The main difference between the stochastic optimization (2.5) and the conventional deterministic problem (2.2) is the uncertainties of  $\Delta W_i$  will cause the uncertainties of objective function, which is shown as Fig. 2.3 (b).

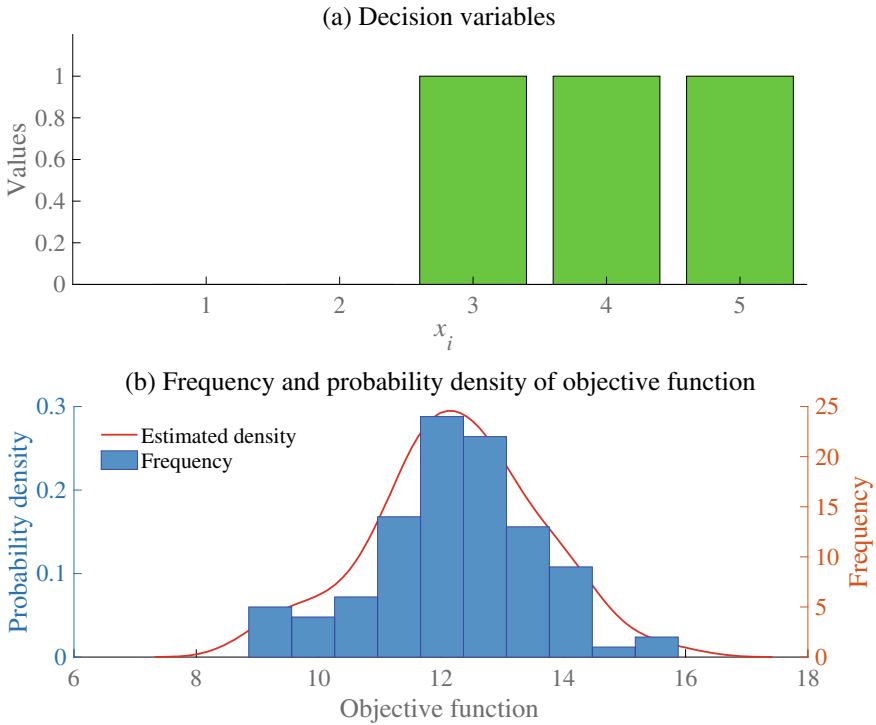


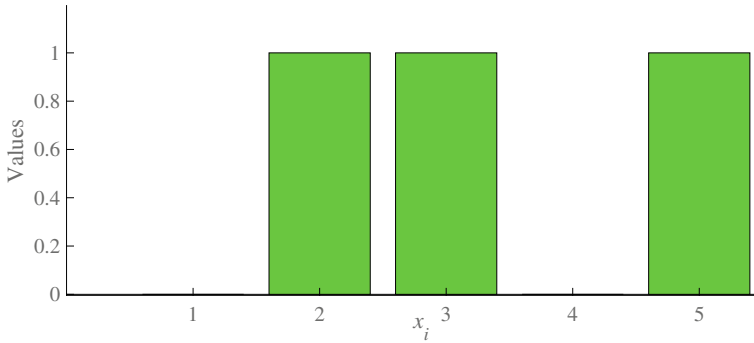
Fig. 2.3 Simulation results of Example 2.2

### 2.2.2 Robust Optimization

A general form of robust optimization is shown as (2.6) [3].

$$\min_{x \in X} g(x) + \max_{\xi \in U} \left( \min_{y \in Y(x, \xi)} f(y) \right) \tag{2.6}$$

In robust optimization (Eq. 2.6), the main difference is the uncertain variable  $\xi$  is described by the uncertainty set  $U$ , including the upper/lower limits and the uncertainty budget. Then robust optimization seeks the optimal solution in the worst case in the defined uncertainty set and therefore brings conservatism. With above, the primary problem of the uncertainty modeling is how to determine the feasible regions, such as the probability distributions in stochastic optimization and the uncertainty set in robust optimization. Similarly, we can give a robust knapsack problem as Example 2.3.



**Fig. 2.4** Simulation results of Example 2.3

### Example 2.3: Robust Knapsack Problem

Based on all the assumptions of Examples 2.1 and 2.2, we further assume  $\Delta W_i$  is within a range denoted as  $[W_L, W_U]$ , and the robust knapsack problem can be shown as (2.7). The meaning of each part is similar to the stochastic model of (2.5).

$$\begin{aligned} \max \left( \sum_{i=1}^{n_f} W_i \cdot x_i + \sum_{i=n_f+1}^n W_c \cdot x_i \right) + \min_{\Delta W_i} \left( \max \left( \sum_{i=n_f+1}^n \Delta W_i \cdot x_i \right) \right) \\ \text{s.t. } \sum_{i=1}^n S_i \cdot x_i \leq C \\ x_i \in \{0, 1\}, \Delta W_i \in [W_L, W_U] \end{aligned} \quad (2.7)$$

### Case Study for Example 2.3

Here we test a simple case, and the parameters are shown as follows:  $n = 5$ ,  $\{W_i | i = 1, 2, 3\} = [2, 3, 1]$ , and  $\{W_c | i = 4, 5\} = [4, 7]$ , and  $\{\Delta W_i | i = 4, 5\} \in [-2, 1]$ , and  $\{S_i | i = 1, 2, \dots, 5\} = [2, 2, 1, 2, 3]$ , and  $C = 6$ . The simulation results are shown in Fig. 2.4.

From the Fig. 2.4, the best solution for Example 2.3 in robust optimization becomes the 2nd, 3rd and 5th goods, and the value of the objective function is 9. This change is due to the risk of the 4th good, since in the worst case, its value becomes 2, and it is not worthy to select. From the above results, we can find the results of robust optimization is conserve.

### 2.2.3 Interval Optimization

Interval optimization can be viewed as an enhancement of robust optimization and consisted of a lower sub-problem and an upper sub-problem, shown as (2.8) [4], and the upper sub-problem is similar with the robust optimization of (2.6). It should

be noted that, for the maximization problem, the lower sub-problem is a robust optimization problem. The main advantage of interval optimization is the interval obtained can be used to analyze the influences of uncertainties on the system. A case is given as Example 2.4.

$$\left[ \underbrace{\min_{x \in X} g(x) + \min_{\xi \in U} \left( \min_{y \in Y(x, \xi)} f(y) \right)}_{\text{Lower sub-problem}}, \underbrace{\min_{x \in X} g(x) + \max_{\xi \in U} \left( \min_{y \in Y(x, \xi)} f(y) \right)}_{\text{Upper sub-problem}} \right] \quad (2.8)$$

#### Example 2.4: Interval Knapsack Problem

$$\text{Lower: } \max \left( \sum_{i=1}^{n_f} W_i \cdot x_i + \sum_{i=n_f}^n W_c \cdot x_i \right) + \min_{\Delta W_i} \left( \max \left( \sum_{i=n_f}^n \Delta W_i x_i \right) \right) \quad (2.9)$$

$$\text{Upper: } \max \left( \sum_{i=1}^{n_f} W_i \cdot x_i + \sum_{i=n_f}^n W_c \cdot x_i \right) + \max_{\Delta W_i} \left( \max \left( \sum_{i=n_f}^n \Delta W_i x_i \right) \right) \quad (2.10)$$

#### Case Study for Example 2.4

The parameters of Example 2.4 is the same as Example 2.3, and the decision variables keep the same as Example 2.3, shown as Fig. 2.5, and the range of objective function is [9, 12]. From this, we can see the interval optimization can give both pessimistic and optimistic scenarios.

In summary, how to get the range of uncertain variables, i.e., the probability distribution function or the uncertainty set of  $\xi$ , is the basic problem of the optimization model. Nowadays, with the development of measurement and communication technology, more operating data can be transmitted and stored in the control center in

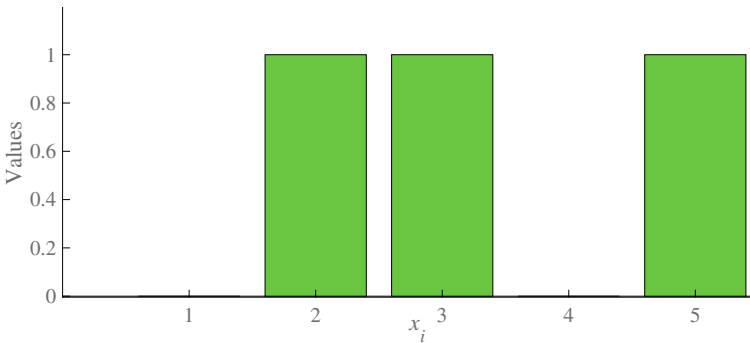


Fig. 2.5 Simulation results of Example 2.4



real-time. How to use this type of massive data to model the feasible region of uncertainty has become a hot topic, and various methods have been proposed. This topic will discuss in Chap. 4.

## 2.3 Convex Optimization

The importance of convex optimization has been emphasized in the former context, and in practical cases, we always want to model or reformulate a complex problem as convex ones, and the semi-definite programming (SDP) and the second-order cone programming (SOCP) are two classic types and have been well studied, which has gained the concerns from both academic and industry.

### 2.3.1 Semi-definite Programming

The general form of SDP is given as (2.11) [5].

$$\begin{aligned} & \min A_0 \cdot X \\ \text{s.t. } & A_p \cdot X = b_p, (p = 1, 2, \dots, m) \\ & 0 \preceq X \in S^n \end{aligned} \quad (2.11)$$

where  $A_0, A_p$  are all coefficient matrixes;  $X$  is the decision matrix which should be semi-definite;  $b_p$  is a coefficient vector;  $S^n$  is the real space with  $n$  dimensions. Conventional linear optimization (LP) and quadratic optimization (QP) can be both formulated as SDP by defining  $X = x \cdot x^T$  [6], then many commercial solvers can be used to solve the reformulated SDP for the global optimal solution, like Sedumi.

### 2.3.2 Second-Order Cone Programming

The general form of SOCP is given as (2.12) [7].

$$\begin{aligned} & \min f^T \cdot x \\ \text{s.t. } & \|A_i \cdot x + b_i\|_2 \leq c_i^T \cdot x + d_i, i = 1, 2, \dots, n \\ & F \cdot x = g \end{aligned} \quad (2.12)$$

where  $f^T, A_i, b_i, c_i^T, d_i, F, g$  are all coefficient vectors or matrixes;  $x$  is the decision variables. It should be noted that the objective function is no need to be linear, and

quadratic objective function can also be solved like conventional SOCP. Similarly, many types of optimization problems can be reformulated as SOCP, and several cases are given below to show the usages of SOCP.

(1) Quadratic terms

For quadratic terms like  $x^2$ , it can be relaxed by the following (2.13) [8].

$$x^2 \leq W, \|2W, \gamma - x\|_2 \leq \gamma + q \quad (2.13)$$

(2) Bilinear terms

For bilinear terms like  $x \cdot y$ , it can be relaxed by the following (2.14) [8].

$$x \cdot y = z \quad (2.14)$$

$$\begin{aligned} \frac{1}{2}(x+y)^2 - \frac{1}{2}(x^2+y^2) &\leq z \\ \frac{1}{2}(x^2+y^2) - \frac{1}{2}(x+y)^2 &\leq z \end{aligned} \quad (2.15)$$

In (2.15),  $-\frac{1}{2}(x^2+y^2)$  and  $-\frac{1}{2}(x+y)^2$  are concave, and the following convex-concave procedure can be used to convexify them [9].

$$\begin{aligned} \frac{1}{2}(x+y)^2 - \frac{1}{2}(\bar{x}^2 + \bar{y}^2) - \bar{x} \cdot (x - \bar{x}) - \bar{y} \cdot (y - \bar{y}) &\leq z \\ \frac{1}{2}(x^2+y^2) - \frac{1}{2}(\bar{x} + \bar{y})^2 - (\bar{x} + \bar{y})(x - \bar{x} + y - \bar{y}) &\leq z \end{aligned} \quad (2.16)$$

where  $(\bar{x}, \bar{y})$  is a constant reference point.

(3) Exponential terms

For bilinear terms like  $e^x$ , it can be relaxed by the following (2.17).

$$y = e^x, \log(y) \geq x, \log(y) \leq x \quad (2.17)$$

Then at a reference point  $\bar{y}$ , (2.17) can be reformulated as (2.18) similarly by the convex-concave procedure [9].

$$\log(\bar{y}) + \frac{1}{\bar{y}} \cdot (y - \bar{y}) \leq x \quad (2.18)$$

## 2.4 Optimization Frameworks

### 2.4.1 Two-Stage Optimization

In reality, there are many cases that the decision variables cannot be determined in the same time, and this is the main motivation of two-stage optimization. The stochastic and robust optimization models in (2.4) and (2.6) are both two-stage optimization. Here we give a general form of two-stage optimization as (2.19) [10].

$$\begin{aligned} & \min_{x \in X} g(x) + \min_{y \in Y} f(y) \\ \text{s.t. } & l(x) \leq 0, h(y) \leq 0 \\ & G(x, y) \leq 0 \end{aligned} \quad (2.19)$$

In the above formulation,  $g(x)$  and  $f(y)$  are the objective functions of the first stage and the second stage, respectively; and  $x, y$  are the corresponding decision variables;  $l(x) \leq 0, h(y) \leq 0$  are the corresponding constraints and  $G(x, y) \leq 0$  is the coupling constraints. It should be noted that, two-stage means  $x, y$  cannot be determined in the same time. To clarify this problem, Example 2.5 is given below.

#### Example 2.5: Two-stage Knapsack problem

Based on all the assumptions of Example 2.1, we assume that the  $i$ th good when  $i = 1, 2, \dots, n_1$  is available now and the  $i$ th good when  $i = n_2, \dots, n$  will be available after some times, and  $n_2 \leq n_1$ . The objective is still the maximization of the total values, but each good can only be selected one time. Then the optimization problem becomes (2.20).

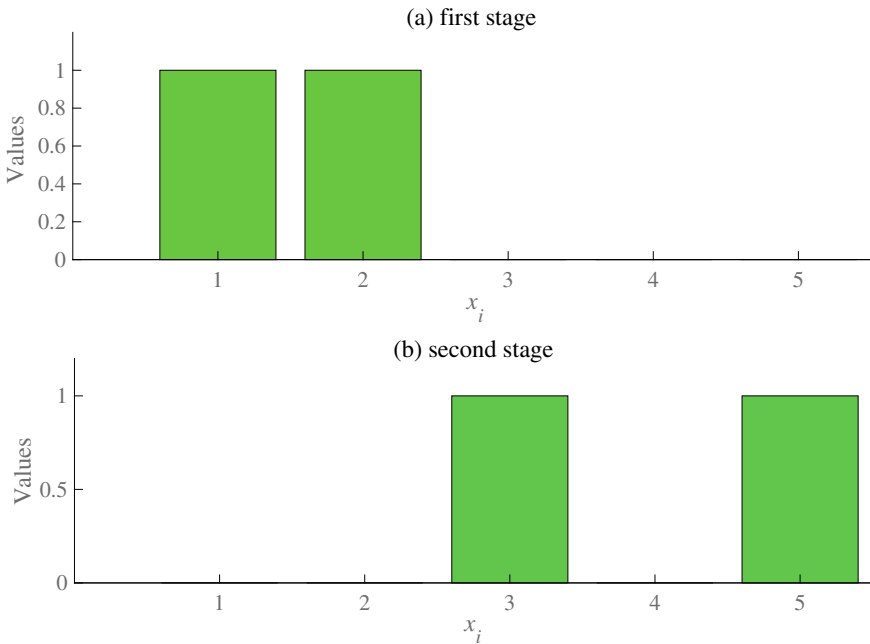
$$\begin{aligned} & \min_{x \in X} \sum_{i=1}^{n_1} W_i \cdot x_i + \sum_{j=n_2}^n W_j \cdot y_j \\ \text{s.t. } & \sum_{i=1}^{n_1} S_i \cdot x_i \leq C, \sum_{j=n_2}^n S_j \cdot y_j \leq C, x_i \in \{0, 1\}, y_j \in \{0, 1\} \\ & x_i + y_j \leq 1, i \in n_1, \dots, n_2 \end{aligned} \quad (2.20)$$

In the above formulation,  $\sum_{i=1}^{n_1} W_i \cdot x_i$  and  $\sum_{j=n_2}^n W_j \cdot y_j$  are the objective functions of the first stage and the second stage, respectively, and  $\sum_{i=1}^{n_1} S_i \cdot x_i \leq C, \sum_{j=n_2}^n S_j \cdot y_j \leq C$  are their corresponding constraints, and  $x_i + y_j \leq 1, i \in n_1, \dots, n_2$  is the coupling constraints.

#### Case study for Example 2.5

Here we test a simple case, and the parameters are shown as follows:  $n = 5, n_1 = 3$  and  $n_2 = 2$ , and  $\{W_i | i = 1, 2, \dots, 5\} = [2, 3, 1, 4, 7]$ , and  $\{S_i | i = 1, 2, \dots, 5\} = [2, 2, 1, 2, 3]$ , and  $C = 6$ . The simulation results are shown in Fig. 2.6.

From Fig. 2.6, the final objective function is 13 by the final selections of the 1st, 2nd, 3rd, and 5th goods. In the first stage, the capacity of the knapsack is 4, and



**Fig. 2.6** Simulation results of Example 2.5

the 1st and 2nd goods are selected, then in the second stage, the 3rd and 5th goods are selected, and no good has been selected for twice. If the coupling constraint  $x_i + y_j \leq 1, i \in n_1, \dots, n_2$  is eliminated and the value of the 3rd good comes to 2, then the final selections are the 2nd good, and 3rd good for twice and the 5th good, and the objective function comes to 14.

In summary, the coupling constraint in two-stage optimization is essential which could influence the final results. Which is proved by many practical cases, the modifications on the coupling constraints benefit the objective function [10, 11].

### 2.4.2 Bi-level Optimization

Bi-level optimization is a special type of two-stage optimization and has a general formulation as following (2.21) [12]. In the following formulation,  $F(x, y)$  represents the upper-level objective function and  $f(x, z)$  represents the lower-level objective function. Similarly,  $x$  represents the upper-level decision vector and  $y$  represents the lower-level decision vector.  $G_i(x, y)$  and  $g(x, z)$  represents the inequality constraint functions at the upper and lower levels respectively. We can find that  $y$  is the decision variable of  $F(x, y)$  and also the optimal decision variable to minimize  $f(x, z)$ . The upper and lower levels are coupled to achieve the overall optimum. Here we also give Example 2.6.

$$\begin{aligned}
& \min_{x \in X, y \in Y} F(x, y) \\
& \text{s.t. } G_i(x, y) \leq 0, i = 1, 2, \dots, I \\
& y \in \text{arg min}_{z \in Y} \{f(x, z) : g(x, z) \leq 0\}
\end{aligned} \tag{2.21}$$

### Example 2.6: Bi-level Knapsack problem

Based on all the assumptions of Example 2.1, we assume inside the outer knapsack with the capacity of  $C_o$ , there is a small bag which holds the most valuable goods, and the capacity is  $C_s$ , and the objective is to maximize the total values and also in the small bag. Then the optimization problem becomes (2.22).

$$\begin{aligned}
& \max \left( \sum_{i=1}^{n_f} W_i \cdot x_i + \sum_{i=n_f}^n W_i \cdot x_i \right) \\
& \text{s.t. } \sum_{i=1}^{n_f} S_i \cdot x_i \leq C_o - C_s, x_i \in \{0, 1\} \\
& \{x_i | i = n_f, n_f + 1, \dots, n\} = \text{argmax} \left\{ \sum_{i=n_f}^n W_i \cdot x_i \mid \sum_{i=n_f}^n S_i \cdot x_i \leq C_s \right\}
\end{aligned} \tag{2.22}$$

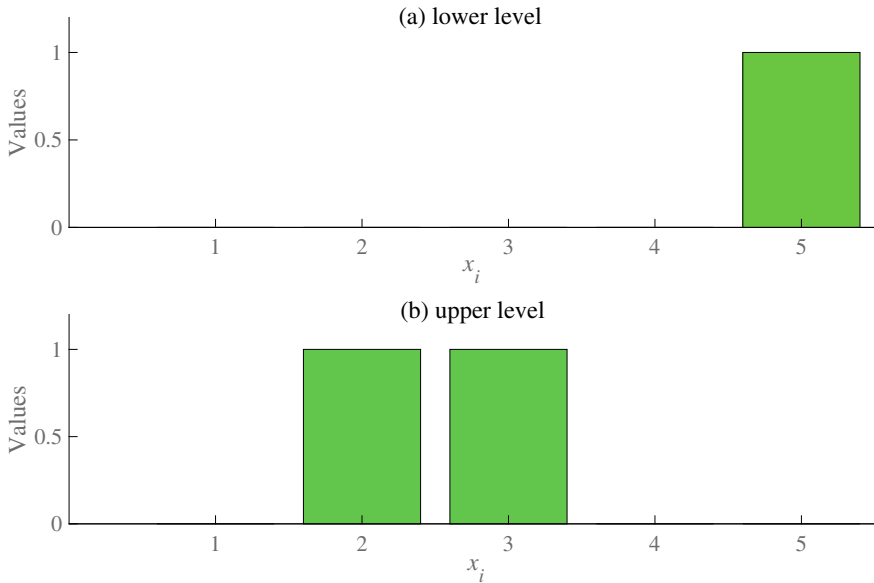
In the above formulation,  $\sum_{i=1}^{n_f} W_i \cdot x_i + \sum_{i=n_f}^n W_i \cdot x_i$  and  $\sum_{i=n_f}^n W_i \cdot x_i$  are the objective functions of the upper level and lower level, respectively.  $\sum_{i=1}^{n_f} S_i \cdot x_i \leq C_o - C_s$  and  $\sum_{i=n_f}^n S_i \cdot x_i \leq C_s$  are their constraints, respectively.

### Case study for Example 2.6

Here we test a simple case, and the parameters are shown as follows:  $n = 5, n_f = 2$ , and  $\{W_i | i = 1, 2, \dots, 5\} = [2, 3, 1, 4, 7]$ , and  $\{S_i | i = 1, 2, \dots, 5\} = [2, 2, 1, 2, 3]$ , and  $C_s = 3$ , and  $C_o = 6$ . The simulation results are shown in Fig. 2.7. From the results, the lower level selects the 5th good in the first place and then in the upper level, the 2nd and 3rd goods are selected, and the final objective function comes to 11.

## 2.5 Summary

This chapter has briefly introduced the frequently used optimization models in engineering, and listed several important literature in the references for the readers. Simple testcases are also given to show different types of optimization models, and the models above will be used in the rest chapters of this book.



**Fig. 2.7** Simulation results of Example 2.6

## References

1. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press (2004)
2. Heyman, D., Sobel, J.: *Stochastic Models in Operations Research: Stochastic Optimization*. Courier Corporation (2004)
3. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: *Robust Optimization*. Princeton University Press (2009)
4. Bhurjee, A.K., Panda, G.: Efficient solution of interval optimization problem. *Math. Methods Oper. Res.* **76**(3), 273–288 (2012)
5. Vandenberghe, L., Boyd, S.: Semidefinite programming. *SIAM Rev.* **38**(1), 49–95 (1996)
6. Boyd, S., Vandenberghe, L.: Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. In: *Communications, Computation, Control, and Signal Processing*. Springer, Boston, MA, pp. 279–287 (1997)
7. Lobo, M., Vandenberghe, L., Boyd, S., et al.: Applications of second-order cone programming. *Linear Algebra Appl.* **284**(1–3), 193–228 (1998)
8. Zamzam, S., Dall’Anese, E., Zhao, C., et al.: Optimal water–power flow–problem: formulation and distributed optimal solution. *IEEE Trans. Control Netw. Syst.* **6**(1), 37–47 (2018)
9. Lipp, T., Boyd, S.: Variations and extension of the convex–concave procedure. *Optim. Eng.* **17**(2), 263–287 (2016)
10. Zeng, B., Zhao, L.: Solving two-stage robust optimization problems using a column-and-constraint generation method. *Oper. Res. Lett.* **41**(5), 457–461 (2013)
11. Zhao, C., Wang, J., Watson, J., et al.: Multi-stage robust unit commitment considering wind and demand response uncertainties. *IEEE Trans. Power Syst.* **28**(3), 2708–2717 (2013)
12. Dempe, S.: *Foundations of Bilevel Programming*. Springer Science & Business Media (2002)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

