Chapter 2

Cost-Reliability Trade-Off of COTS Selection in Fuzzy Environment
In CBSD, to develop a software system that has high reliability, rapid delivery, and low cost, an extensive optimization analysis among their trade-offs is essential. In practice, system reliability, system delivery time, and the cost of COTS components are often imprecise. Thus, to deal with the imprecise parameters, we introduce a fuzzy cost-reliability trade-off model of COTS selection for the development of modular software systems. The proposed fuzzy optimization model minimizes the total cost of the software system satisfying the constraints of minimum threshold on system reliability, maximum threshold on the delivery time of the software, and incompatibility among COTS components. The model considers uncertainty in the coefficients of the objective function and constraints. Generally, the fuzzy data in fuzzy mathematical programming are expressed in terms of fuzzy numbers. Here, we consider the coefficients of the cost objective function, delivery time constraints, and minimum threshold on reliability as TFNs. In this chapter, we develop a solution approach by using Zadeh’s extension principle to provide fuzzy solutions instead of a crisp solution of the fuzzy optimization model. In the recent past, many researchers, for example, Chen [21], Chen and Tsai [22], Liu and Kao [76], and Liu [77, 78], have used this approach in many important applications of real-world decision making.

To obtain the membership function of the fuzzy total cost, the fuzzy optimization model is transformed into two conventional crisp mathematical programming problems that calculate the lower and upper bounds of the fuzzy objective value at different possibility (feasibility) levels $\alpha$. The lower and upper bounds of the fuzzy objective value at different possibility levels $\alpha$ provide the left-shape function and the right-shape function, respectively, of the membership function of the fuzzy total cost. The proposed solution approach completely preserves the fuzziness of model parameters and provides more information for cost-reliability trade-off analysis in COTS selection.

The rest of the chapter is organized as follows. Section 2.1 describes the COTS selection problem along with the notation used, related assumptions, and formulation of the fuzzy optimization model of COTS selection. Section 2.2 presents the solution methodology using membership function approach. Section 2.3 deals with numerical illustration of the fuzzy optimization model by discussing
a real-world case study of a COTS-based modular software system. This section also includes a discussion of the results obtained and comparison with well-known ranking function approach from the literature. Finally, Section 2.4 concludes the chapter.

2.1 Fuzzy COTS selection model

Here, we consider the design of a modular software system under single application development task. The software system consists of several programs, where a specific function of each program can call upon a series of modules. A failure occurs in the software system if a module fails to carry out a designated operation. We assume that several alternative COTS components are available for each module. The modules are developed independently using best-fit COTS components among available alternatives.

We use the following notation throughout this chapter where the tilde ($\sim$) sign corresponds to fuzziness in the model parameters:

- $i$: the index for modules, $i = 1, 2, \ldots, m$,
- $j$: the index for alternative COTS components for the $i$-th module, $j = 1, 2, \ldots, n_i$,
- $m$: the number of modules in the software system,
- $n_i$: the number of alternative COTS components in the $i$-th module, $i = 1, 2, \ldots, m$,
- $s_i$: the average number of invocations of the $i$-th module, $i = 1, 2, \ldots, m$,
- $p_i$: the probability of occurrence of at least one failure when the $i$-th module is executed, $i = 1, 2, \ldots, m$,
- $\mu_{ij}$: the probability of failure on demand of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,
- $\tilde{C}_{ij}$: the imprecise cost of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,
- $\tilde{D}_{ij}$: the imprecise delivery time of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$. 

\( \tilde{T}_i \): the imprecise delivery time of the \( i \)-th module, \( i = 1, 2, \ldots, m \),
\( \tilde{T} \): the imprecise overall system delivery time,
\( \tilde{R}^* \): the imprecise minimum system reliability threshold,
\( \tilde{Z} \): the fuzzy total cost of the software system,

\( x_{ij} \): the binary variable indicating whether the \( j \)-th COTS component of the \( i \)-th module is chosen or not,

\[
x_{ij} = \begin{cases} 
1, & \text{if the } j \text{-th COTS component of the } i \text{-th module is chosen}, \\
0, & \text{otherwise.}
\end{cases}
\]

Our goal is to select the best-fit COTS component for each module satisfying both the system requirements and the budget when many important parameters are not known precisely. We propose a fuzzy optimization model for COTS selection under the following assumptions:

(i) the coefficients of the cost objective function and delivery time constraints, and targeted threshold on system reliability are considered TFNs;

(ii) the objective function and the constraints are linear;

(iii) the cost of COTS components is based on purchasing price which includes the procurement and the adaptation cost;

(iv) the delivery time and the probability of failure on demand of COTS components are provided by vendor.

In the proposed fuzzy optimization model for COTS selection, we consider the following objective and constraints.

### 2.1.1 Objective

- **Cost**

The objective function corresponding to the total cost of COTS components given that the cost coefficients are imprecise, is expressed as

\[
\min \tilde{Z} = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \tilde{C}_{ij} x_{ij}.
\]
2.1.2 Constraints

- **Delivery time constraints**

  The delivery time $\bar{T}_i$ of the $i$-th module is expressed as

  \[
  \bar{T}_i = \sum_{j=1}^{n_i} \bar{D}_{ij} x_{ij}, \quad i = 1, 2, \ldots, m.
  \]

  Therefore, the delivery time constraint on the software system is obtained as

  \[
  \max_{i=1,2,\ldots,m} (\bar{T}_i) \leq \bar{T},
  \]

  which can be decomposed in the set of constraints $\bar{T}_1 \leq \bar{T}, \bar{T}_2 \leq \bar{T}, \ldots, \bar{T}_m \leq \bar{T}$.

  Alternatively, delivery time constraints may also be reformulated as

  \[
  \sum_{j=1}^{n_i} \bar{D}_{ij} x_{ij} \leq \bar{T}, \quad i = 1, 2, \ldots, m.
  \]

- **Reliability constraint**

  Assuming that the failure occurrence in a module is equivalent to the failure in the software system, the reliability of the software system can be obtained as a function of the probability of failure on demand of its modules which are developed using COTS components. Referring to the work of Berman [10], the effect of failure from a module can be represented as an exponential function.

  Thus, the probability that no failure occurs during the execution of the $i$-th module is obtained using a poisson distribution as

  \[
  1 - p_i = \exp \left( - s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \right), \quad i = 1, 2, \ldots, m,
  \]

  where $\sum_{j=1}^{n_i} s_i \mu_{ij} x_{ij}$ is the average number of failures in the execution of the $i$-th module.

  It may be noted that the value of the parameter $s_i$ does not depend on the COTS component, because we assume that the pattern of interactions within each scenario does not change by changing the COTS component. This value is obtained by processing the various execution scenarios and the number of
invocations is averaged over all the scenarios by using the probability of each scenario to be executed.

In view of the poisson probability principle, which states that the probability of failure occurring is proportional to the length of sojourn time and the assumption that an operation of each module is independent since the modules are developed independently using COTS components, the probability of failure-free software system may be formulated as

$$\prod_{i=1}^{m} (1 - p_i) = \prod_{i=1}^{m} \exp \left( - s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \right).$$

Therefore, the system reliability constraint corresponding to the given minimum threshold $\tilde{R}^*$ may be expressed as

$$\prod_{i=1}^{m} \exp \left( - s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \right) \geq \tilde{R}^*.$$

It may be noted that due to the presence of exponential function in the above stated reliability constraint, it is nonlinear in nature. To avoid nonlinearity in the mathematical formulation of COTS selection problem, we rewrite the above defined constraint in an equivalent form as

$$\sum_{i=1}^{m} s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \leq \ln \frac{1}{\tilde{R}^*} \leq \tilde{R},$$

where $\tilde{R} = \left( \frac{1}{R^*} - 1 \right)$. Note that for a TFN $\tilde{R}^* = (a, b, c)$, $\frac{1}{\tilde{R}^*} = \left( \frac{1}{c}, \frac{1}{b}, \frac{1}{a} \right)$.

- **Selection of only one COTS component for each module**

$$\sum_{j=1}^{n_i} x_{ij} = 1, \ i = 1, 2, \ldots, m.$$

- **Contingent decision constraints**

In the development of a modular software system, a COTS component for one module may be incompatible with the alternative COTS components for other modules, due to problems such as implementation technology, interfaces, and licensing. For example, $x_{rs} \leq x_{at1}$ means that, if the $r$-th module chooses the
s-th COTS component, then the u-th module must choose the t1-th COTS component. This condition is called the contingent decision constraint [54]. Suppose there are two contingent decisions in the model, such as, the COTS component for the module r is only compatible with the COTS components t1 and t2 for the module u; that is, either \( x_{ut1} = 1 \) if \( x_{rs} = 1 \) or \( x_{ut2} = 1 \) if \( x_{rs} = 1 \). These constraints can be represented as either \( x_{rs} \leq x_{ut1} \) or \( x_{rs} \leq x_{ut2} \).

Since the presence of the either-or constraint makes the optimization problem nonlinear, it can be linearized by introducing a binary variable \( y_i \) defined as

\[
y_i = \begin{cases} 
0, & \text{if the } i \text{-th constraint is active,} \\
1, & \text{if the } i \text{-th constraint is not active.} 
\end{cases}
\]

The constraints corresponding to contingent decision in the COTS selection problem can be restated as

\[
x_{rs} - x_{ut_k} \leq M y_k, \quad k = 1, 2, \ldots, z,
\]

\[
\sum_{k=1}^{z} y_k = z - 1,
\]

\[
y_k \in \{0, 1\}, \quad k = 1, 2, \ldots, z.
\]

The above conversion guarantees that only one out of the \( z \) contingent decision constraints for any COTS components between two modules is active if \( M \) is large.

- **Selection or rejection of a COTS component**

\[
x_{ij} \in \{0, 1\}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n_i.
\]

### 2.1.3 The decision model

The fuzzy optimization model for COTS selection is now formulated as follows:
\[ P(2.1) \quad \min \tilde{Z} = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \tilde{C}_{ij} x_{ij} \]

subject to

\[ \sum_{j=1}^{n_i} \tilde{D}_{ij} x_{ij} \leq \tilde{T}, \quad i = 1, 2, \ldots, m, \quad (2.1) \]

\[ \sum_{i=1}^{m} s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \leq \tilde{R}, \quad (2.2) \]

\[ \sum_{j=1}^{n_i} x_{ij} = 1, \quad i = 1, 2, \ldots, m, \quad (2.3) \]

\[ x_{rs} - x_{ut} \leq M y_k, \quad k = 1, 2, \ldots, z, \quad (2.4) \]

\[ \sum_{k=1}^{z} y_k = z - 1, \quad (2.5) \]

\[ y_k \in \{0, 1\}, \quad k = 1, 2, \ldots, z, \quad (2.6) \]

\[ x_{ij} \in \{0, 1\}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n_i. \quad (2.7) \]

### 2.2 Membership function approach

Let \( \mu_{\tilde{C}_{ij}}(c_{ij}), \mu_{\tilde{D}_{ij}}(d_{ij}), \mu_{\tilde{T}}(T), \) and \( \mu_{\tilde{R}}(R) \) be the membership functions of \( \tilde{C}_{ij}, \tilde{D}_{ij}, \tilde{T}, \) and \( \tilde{R} \), respectively. Then we have the following fuzzy sets:

\[ \tilde{C}_{ij} = \{(c_{ij}, \mu_{\tilde{c}_{ij}}(c_{ij})) \mid c_{ij} \in S(\tilde{C}_{ij})\}, \]

\[ \tilde{D}_{ij} = \{(d_{ij}, \mu_{\tilde{d}_{ij}}(d_{ij})) \mid d_{ij} \in S(\tilde{D}_{ij})\}, \]

\[ \tilde{T} = \{(T, \mu_{\tilde{T}}(T)) \mid T \in S(\tilde{T})\}, \]

\[ \tilde{R} = \{(R, \mu_{\tilde{R}}(R)) \mid R \in S(\tilde{R})\}, \]

where \( S(\tilde{C}_{ij}), S(\tilde{D}_{ij}), S(\tilde{T}), \) and \( S(\tilde{R}) \) are the supports of \( \tilde{C}_{ij}, \tilde{D}_{ij}, \tilde{T}, \) and \( \tilde{R} \), respectively, which denote their crisp universal sets.

Since the model \( P(2.1) \) involves fuzzy numbers in the objective function and constraints (2.1) and (2.2), the minimum objective value of \( \tilde{Z} \) is also a fuzzy number rather than a real number. Most of the existing techniques for solving fuzzy problems provides crisp solutions; however, if the obtained objective value
is crisp then the software developers may lose some helpful information. To preserve the fuzziness of the model parameters, we must derive the membership function of $\bar{Z}$, denoted by $\mu_{\bar{Z}}(z')$. To solve the proposed model, we adopt a solution approach that combines the concepts of $\alpha$-cut, Zadeh’s extension principle [113, 117, 122], and two-level mathematical programming.

We can derive the membership function $\mu_{\bar{Z}}(z')$ of $\bar{Z}$ using $\alpha$-cut approach. The $\alpha$-cut of $\bar{Z}$ is defined as

$$Z_\alpha = \{z' \in S(\bar{Z}) | \mu_{\bar{Z}}(z') \geq \alpha\},$$

where $S(\bar{Z})$ is the support of $\bar{Z}$. It may be noted that $Z_\alpha$ is a crisp set rather than a fuzzy set. Using $\alpha$-cut, $\bar{Z}$ can be represented by different levels of confidence intervals. Since $\bar{Z}$ is a fuzzy number, its $\alpha$-cut defined as above is a crisp interval which can be expressed as

$$Z_\alpha = [\min_{z'}\{z' \in S(\bar{Z}) | \mu_{\bar{Z}}(z') \geq \alpha\}, \max_{z'}\{z' \in S(\bar{Z}) | \mu_{\bar{Z}}(z') \geq \alpha\}] = [Z_\alpha^L, Z_\alpha^U].$$

Following this, the $\alpha$-cuts of fuzzy numbers $\tilde{C}_{ij}$, $\tilde{D}_{ij}$, $\tilde{T}$, and $\tilde{R}$ can be represented as follows:

$$(C_{ij})_\alpha = \{c_{ij} \in S(\tilde{C}_{ij}) | \mu_{\tilde{C}_{ij}}(c_{ij}) \geq \alpha\} = [(C_{ij})^L_\alpha, (C_{ij})^U_\alpha],$$

$$(D_{ij})_\alpha = \{d_{ij} \in S(\tilde{D}_{ij}) | \mu_{\tilde{D}_{ij}}(d_{ij}) \geq \alpha\} = [(D_{ij})^L_\alpha, (D_{ij})^U_\alpha],$$

$$T_\alpha = \{T \in S(\tilde{T}) | \mu_{\tilde{T}}(T) \geq \alpha\} = [T^L_\alpha, T^U_\alpha],$$

$$R_\alpha = \{R \in S(\tilde{R}) | \mu_{\tilde{R}}(R) \geq \alpha\} = [R^L_\alpha, R^U_\alpha].$$

Based on the extension principle, the membership function $\mu_{\bar{Z}}(z')$ is defined as

$$\mu_{\bar{Z}}(z') = \sup_{c,d,T,R} \min \{\mu_{\tilde{C}_{ij}}(c_{ij}), \mu_{\tilde{D}_{ij}}(d_{ij}), \mu_{\tilde{T}}(T), \mu_{\tilde{R}}(R) \forall i,j \}$$

$$|z' = Z(c,d,T,R)| \tag{2.8}$$

where $Z(c,d,T,R)$ is the objective value of crisp equivalent model of P(2.1) which can be obtained using the crisp parameters $c_{ij}$, $d_{ij}$, $T$, and $R$. If the $\alpha$-cuts of $\bar{Z}$ at all $\alpha$ values degenerate to the same value, then the minimum total cost is a crisp number; otherwise, it is a fuzzy number. Since several membership functions are involved in (2.8), it is difficult to derive $\mu_{\bar{Z}}(z')$ in closed form. Also,
\(\mu_Z(z')\) is the minimum of \(\mu_{\tilde{c}_{ij}}(c_{ij}), \mu_{\tilde{d}_{ij}}(d_{ij}), \mu_{\tilde{p}}(T), \mu_{\tilde{r}}(R), \forall i, j\), therefore, in order to satisfy \(\mu_Z(z') = \alpha\), we need \(\mu_{\tilde{c}_{ij}}(c_{ij}) \geq \alpha, \mu_{\tilde{d}_{ij}}(d_{ij}) \geq \alpha, \mu_{\tilde{p}}(T) \geq \alpha, \mu_{\tilde{r}}(R) \geq \alpha\), and at least one \(\mu_{\tilde{c}_{ij}}(c_{ij}), \mu_{\tilde{d}_{ij}}(d_{ij}), \mu_{\tilde{p}}(T), \text{ or } \mu_{\tilde{r}}(R), \forall i, j\), equal to \(\alpha\). To find the membership function \(\mu_Z(z')\), it is sufficient to find the left-shape function and the right-shape function of \(\mu_Z(z')\), which is equivalent to finding the lower bound \(Z^L_\alpha\) and the upper bound \(Z^U_\alpha\) of \(\tilde{Z}\) at different possibility level \(\alpha\). Since \(Z^L_\alpha\) is the minimum of \(Z(c, d, T, R)\) and \(Z^U_\alpha\) is the maximum of \(Z(c, d, T, R)\), we have

\[
Z^L_\alpha = \min \{Z(c, d, T, R) | (C_{ij})^L_\alpha \leq c_{ij} \leq (C_{ij})^U_\alpha, (D_{ij})^L_\alpha \leq d_{ij} \leq (D_{ij})^U_\alpha, T^L_\alpha \leq T \leq T^U_\alpha, R^L_\alpha \leq R \leq R^U_\alpha, \forall i, j\},
\]

\[
Z^U_\alpha = \max \{Z(c, d, T, R) | (C_{ij})^L_\alpha \leq c_{ij} \leq (C_{ij})^U_\alpha, (D_{ij})^L_\alpha \leq d_{ij} \leq (D_{ij})^U_\alpha, T^L_\alpha \leq T \leq T^U_\alpha, R^L_\alpha \leq R \leq R^U_\alpha, \forall i, j\}.
\]

Also, the different values of \(c_{ij}, d_{ij}, T, \text{ and } R\) produces different objective values. Thus, the values of \(c_{ij}, d_{ij}, T, \text{ and } R\) which produces smallest value for \(Z^L_\alpha\) at possibility level \(\alpha\) can be determined from the following two-level mathematical programming problem:

\[
Z^L_\alpha = \min \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \right\} \text{ subject to } \sum_{j=1}^{m} d_{ij} x_{ij} \leq T, i = 1, 2, \ldots, m, \quad \sum_{i=1}^{m} s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \leq R, \text{ and Constraints (2.3)-(2.7)} (2.9)
\]

Similarly, to find the values of \(c_{ij}, d_{ij}, T, \text{ and } R\) that produces the largest objective value, a two-level mathematical programming problem is formulated by replacing the outer-level program of (2.9) from ‘min’ to ‘max’.

\[
Z^U_\alpha = \max \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \right\} \text{ subject to } \sum_{j=1}^{m} d_{ij} x_{ij} \leq T, i = 1, 2, \ldots, m, \quad \sum_{i=1}^{m} s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \leq R, \text{ and Constraints (2.3)-(2.7)} (2.10)
\]
2.2.1 One-level mathematical programming models

To derive the lower bound of the objective value in model P(2.1) for a desired \( \alpha \), we can directly set \( c_{ij} \) in the model P(2.1) to its lower bound \( (C_{ij})_\alpha^L \), \( \forall i, j \), to find the minimum objective value. Since, two-level mathematical programming model (2.9) determines the minimum of all the minimum objective values, we can insert the constraints of level 1 into level 2 and simplify the two-level mathematical programming model to the conventional one-level model as follows:

\[
P(2.2) \quad Z_\alpha^L = \min \sum_{i=1}^{m} \sum_{j=1}^{n_i} (C_{ij})_\alpha^L x_{ij}
\]

subject to

\[
\sum_{j=1}^{n_i} d_{ij} x_{ij} \leq T, \quad i = 1, 2, \ldots, m,
\]

\[
\sum_{i=1}^{m} s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \leq R,
\]

\[
(D_{ij})_\alpha^L \leq d_{ij} \leq (D_{ij})_\alpha^U, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n_i,
\]

\[
T_\alpha^L \leq T \leq T_\alpha^U,
\]

\[
R_\alpha^L \leq R \leq R_\alpha^U,
\]

and Constraints (2.3)–(2.7).

Note that model P(2.2) is an NLPP due to presence of the product terms \( d_{ij} x_{ij}, \forall i, j \). Although, for medium or large-sized problems, it is expected that solving the NLPP could be computationally difficult, this is not the case, as many excellent softwares are available to solve them. We can use LINGO to solve model P(2.2). In this model, since all \( c_{ij} \) have been set to the lower bounds of their possibility level \( \alpha \), that is, \( \mu C_{ij} (c_{ij}) = \alpha \), this assures \( \mu Z (z') = \alpha \) as required by (2.8).

To convert the two-level mathematical programming model (2.10) into a one-level mathematical programming model, the dual of the level 2 problem is formulated to get a maximization problem in order to be consistent with the maximization operation of level 1. It is well known from the duality theorems of linear programming that the primal problem and the dual problem have the same optimal objective values. Thus, two-level mathematical programming model (2.10) becomes:
\[ Z_U = \max \quad (C_{ij}) \leq c_{ij} \leq (C_{ij})^U \]

\[ (D_{ij}) \leq d_{ij} \leq (D_{ij})^U \]

\[ r_k \leq r_k \leq (r_k)^U \]

\[ \forall i, j \]

\[ \begin{align*}
\max & \quad \left( \sum_{i=1}^{m} u_i - \sum_{i=1}^{m} T \cdot v_i - R \cdot w - \sum_{i=1}^{m} \rho_{ij} - \sum_{k=1}^{z} \gamma_k + \tau \right) \\
\text{subject to} & \quad u_i - d_{ij} \cdot v_i - s_i \cdot \mu_{ij} \cdot w - \rho_{ij} \leq c_{ij}, \quad i = 1, 2, \ldots, m, \\
& \quad j = 1, 2, \ldots, n_i, j \neq t_k, k = 1, 2, \ldots, z; \text{ if } i = u \\
& \quad \text{and } j \neq s; \text{ if } i = r, \\
& \quad u_i - d_{ij} \cdot v_i - s_i \cdot \mu_{ij} \cdot w - \rho_{ij} + \lambda_k \leq c_{ij}, \quad i = u, \\
& \quad j = t_k, \quad k = 1, 2, \ldots, z, \\
& \quad u_i - d_{ij} \cdot v_i - s_i \cdot \mu_{ij} \cdot w - \rho_{ij} - \sum_{k=1}^{z} \lambda_k \leq c_{ij}, \quad i = r, \\
& \quad j = s, \\
& \quad \tau + M \cdot \lambda_k - \gamma_k \leq 0, \quad k = 1, 2, \ldots, z, \\
& \quad v_i \geq 0, \quad w \geq 0, \quad \rho_{ij} \geq 0, \quad \gamma_k \geq 0, \quad \lambda_k \geq 0, \quad \forall i, j, k.
\end{align*} \]

Note that \( u_i, v_i, w, \rho_{ij}, \gamma_k, \lambda_k, \) and \( \tau, \forall i, j, k \) are dual variables used in the construction of dual problem corresponding to the level 2 problem of model (2.10). Since \((C_{ij})_a^L \leq c_{ij} \leq (C_{ij})_a^U, \forall i, j \) in model (2.11), we can derive the upper bound of the objective function by setting \( c_{ij} \) to its upper bound \((C_{ij})_a^U, \forall i, j \). Since, model (2.11) involves the maximum of all the maximum objective values, we can insert the constraints of level 1 into level 2 and simplify the two-level mathematical programming model to the conventional one-level model as follows:

\[ P(2.3) \quad Z_\alpha^U = \max \quad \left( \sum_{i=1}^{m} u_i - \sum_{i=1}^{m} T \cdot v_i - R \cdot w - \sum_{i=1}^{m} \sum_{j=1}^{n_i} \rho_{ij} - \sum_{k=1}^{z} \gamma_k + \tau \right) \]

\[ \text{subject to} \]

\[ u_i - d_{ij} \cdot v_i - s_i \cdot \mu_{ij} \cdot w - \rho_{ij} \leq (C_{ij})_a^U, \quad i = 1, 2, \ldots, m, \]

\[ j = 1, 2, \ldots, n_i, j \neq t_k, k = 1, 2, \ldots, z; \text{ if } i = u \]

\[ \text{and } j \neq s; \text{ if } i = r, \]
2.2 Membership function approach

\[ u_i - d_{ij} \cdot v_i - s_i \cdot \mu_{ij} \cdot w - \rho_{ij} + \lambda_k \leq (C_{ij})^U, \quad i = u, \ j = t_k, \]
\[ k = 1, 2, \ldots, z, \]
\[ u_i - d_{ij} \cdot v_i - s_i \cdot \mu_{ij} \cdot w - \rho_{ij} - \sum_{k=1}^z \lambda_k \leq (C_{ij})^U, \quad i = r, \ j = s, \]
\[ \tau + M \cdot \lambda_k - \gamma_k \leq 0, \quad k = 1, 2, \ldots, z, \]
\[ (D_{ij})^L \leq d_{ij} \leq (D_{ij})^U, \quad i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n, \]
\[ T_a^L \leq T \leq T_a^U, \]
\[ R_a^L \leq R \leq R_a^U, \]
\[ v_i \geq 0, \ w \geq 0, \ \rho_{ij} \geq 0, \ \gamma_k \geq 0, \ \lambda_k \geq 0, \ \forall i, j, k. \]

It may be noted that model P(2.3) is also an NLPP due to presence of the product terms \( T \cdot v_i, \ R \cdot w, \) and \( d_{ij} \cdot v_i, \ \forall i, j. \) In the above model, since all \( c_{ij} \) have been set to the upper bounds of their possibility level \( \alpha, \) that is, \( \mu_{c_{ij}}(c_{ij}) = \alpha, \) this assures \( \mu_{Z}(z') = \alpha \) as required by (2.8).

An important feature of the \( \alpha \)-cut approach is that all \( \alpha \)-cuts form a nested structure with respect to \( \alpha [60], \) that is, for two possibility levels \( \alpha_1 \) and \( \alpha_2 \) such that \( 0 < \alpha_2 < \alpha_1 \leq 1, \) the feasible regions defined by \( \alpha_1 \) in models P(2.2) and P(2.3) are smaller than those defined by \( \alpha_2. \) Consequently, \( Z_a^L \geq Z_{a_2}^L \) and \( Z_a^U \leq Z_{a_2}^U; \) in other words, the left-shape function is non-decreasing and the right-shape function is non-increasing. This property, based on the definition of convex fuzzy sets [122], assures the convexity of \( \widetilde{Z}. \) Consequently, the membership function of \( \widetilde{Z} \) can be constructed.

In order to find the analytic form of the membership function whenever possible, or the approximate membership function via enumerating different \( \alpha \) values, define an increasing function \( Z_a^L : \alpha \rightarrow Z_a^L \) and a decreasing function \( Z_a^U : \alpha \rightarrow Z_a^U. \) If both \( Z_a^L \) and \( Z_a^U \) are invertible with respect to \( \alpha, \) then a left-shape function \( L(z') = (Z_a^L)^{-1} \) and a right-shape function \( R(z') = (Z_a^U)^{-1} \) can be obtained. From \( L(z') \) and \( R(z'), \) the membership function \( \mu_{\widetilde{Z}}(z') \) is constructed as follows:

\[
\mu_{\widetilde{Z}}(z') = \begin{cases} 
L(z'), & Z_{a=0}^L \leq z' \leq Z_{a=1}^L; \\
1, & Z_{a=1}^L \leq z' \leq Z_{a=1}^U; \\
R(z'), & Z_{a=1}^U \leq z' \leq Z_{a=0}^U.
\end{cases}
\]  \hspace{1cm} (2.12)

Alternatively, the shape of \( \mu_{\widetilde{Z}}(z') \) can be approximated from the set of intervals
\{[Z^L_\alpha, Z^U_\alpha] \mid \alpha \in [0,1]\}, although exact function may not be known explicitly. The numerical solutions for \(Z^L_\alpha\) and \(Z^U_\alpha\) at different possibility levels \(\alpha\) can be collected to approximate the shapes of \(L(z')\) and \(R(z')\).

### 2.2.2 Crisp transformation

To find a representative crisp value for \(\tilde{Z}\), we defuzzify the fuzzy number \(\tilde{Z}\) to a crisp value. Many defuzzification approaches [20] have been proposed in the literature. The center of gravity (COG) method is the most trivial weighted average. We use the COG method to defuzzify the fuzzy total cost to a crisp value. Let \(Z^*\) be the defuzzified value of \(\tilde{Z}\). The COG method calculates \(Z^*\) as

\[
Z^* = \frac{\int_{Z^L_{\alpha=0}}^{Z^U_{\alpha=0}} z' \mu_{\tilde{Z}}(z') dz'}{\int_{Z^L_{\alpha=0}}^{Z^U_{\alpha=0}} \mu_{\tilde{Z}}(z') dz'}.
\] (2.13)

If the analytical form of \(\mu_{\tilde{Z}}(z')\) is not obtained, then numerical methods of approximation can be used to approximate \(Z^*\). Several numerical approximation methods have been proposed, of which, a simple and efficient one called the trapezoidal rule [8] is adopted in this chapter. The trapezoidal rule method approximates a continuous function \(f(x)\) on the interval \([a, b]\) using the formula

\[
\int_{a}^{b} f(x) dx \approx \frac{\Delta x}{2} [f(x_0) + 2f(x_1) + \ldots + 2f(x_{n-1}) + f(x_n)],
\] (2.14)

where \(\Delta x = (b - a)/n\), \(n\) is the number of trapezoids which is approximately pre-specified, \(a = x_0\), \(b = x_n\) and \(x_{i+1} = x_i + \Delta x, i = 0,1,\ldots,n-2\). In general, using a larger \(n\) yields a better approximation to the definite integral in (2.14).

### 2.3 An illustrative example

In order to demonstrate the validity and practicality of the proposed COTS selection model in a fuzzy environment and the solution method, an industrial
2.3 An illustrative example

An illustrative example case scenario inspired from CBSD is presented. A local software system supplier undertakes the development of an Enterprise Resource Planning (ERP) software system for small and medium-size enterprises. An ERP system is an integrated application software package composed by a set of standard functional requirements such as planning production, sales, human resources, maintaining inventories, finance, providing customer service, and tracking orders. Application software packages are designed to provide a set of standard functions that can be adapted to the specific needs of each customer. Here, the COTS term refers to application software package. Three functional requirements of the ERP system, namely, Finance (Program 1), Operations (Program 2) and Marketing (Program 3) are considered. The software development team of the company has defined four modules, namely, Accounts ($M_1$), Inventory ($M_2$), Sales Order ($M_3$), and Sales Promotion ($M_4$). Program 1 calls modules $M_1$ and $M_2$ (once each), Program 2 calls modules $M_2$ and $M_3$ (twice each), Program 3 calls modules $M_1$ (twice), $M_3$ (once), and $M_4$ (twice). Module $M_1$ has three alternative COTS components denoted as $sc_{11}$, $sc_{12}$, and $sc_{13}$. Module $M_2$ has two alternative COTS components denoted as $sc_{21}$ and $sc_{22}$. Module $M_3$ has three alternative COTS components denoted as $sc_{31}$, $sc_{32}$, and $sc_{33}$. Module $M_4$ has three alternative COTS components denoted as $sc_{41}$, $sc_{42}$, and $sc_{43}$.

The diagrammatic depiction of an ERP software system is shown in Figure 2.1.

![Figure 2.1: Hierarchy of an ERP software system](image)


2.3.1 Data description

Because of confidentiality as well as the lack of some exact required data, we have used fuzzy parameters characterized by TFNs in accordance with the past data and integrating the judgment of the DM for the numerical experiments. The TFNs of the individual cost and delivery time of COTS components, probability of failure on demand of COTS components, average number of invocations of modules, maximum threshold on the delivery time of the software expressed as TFN, minimum threshold on reliability expressed as TFN, and contingent decision constraints are given in Table 2.1.

**Table 2.1: Input data of the system prototype**

<table>
<thead>
<tr>
<th>Module</th>
<th>$s_i$</th>
<th>COTS component</th>
<th>$\tilde{C}_{ij}$</th>
<th>$\tilde{D}_{ij}$</th>
<th>$\mu_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>110</td>
<td>$sc_{11}$</td>
<td>(11.5,15.5,18)</td>
<td>(4.5,7)</td>
<td>0.0002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$sc_{12}$</td>
<td>(12.5,14.5,16.5)</td>
<td>(3.4,6)</td>
<td>0.0003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$sc_{13}$</td>
<td>(12,15,18)</td>
<td>(5.8,10)</td>
<td>0.0003</td>
</tr>
<tr>
<td>$M_2$</td>
<td>55</td>
<td>$sc_{21}$</td>
<td>(2.5,5.5,7)</td>
<td>(5.7,10)</td>
<td>0.0003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$sc_{22}$</td>
<td>(3.5,8)</td>
<td>(3.6,8)</td>
<td>0.0002</td>
</tr>
<tr>
<td>$M_3$</td>
<td>80</td>
<td>$sc_{31}$</td>
<td>(12,15,18)</td>
<td>(1.4,6)</td>
<td>0.0002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$sc_{32}$</td>
<td>(10,14,17)</td>
<td>(3.7,9)</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$sc_{33}$</td>
<td>(12,14,18)</td>
<td>(3.6,8)</td>
<td>0.0004</td>
</tr>
<tr>
<td>$M_4$</td>
<td>120</td>
<td>$sc_{41}$</td>
<td>(10,12,14)</td>
<td>(3.5,8)</td>
<td>0.0004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$sc_{42}$</td>
<td>(6.8,11)</td>
<td>(4.7,10)</td>
<td>0.0003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$sc_{43}$</td>
<td>(7.8,9)</td>
<td>(3.5,8)</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

$\bar{T}$: (3, 8, 10), $\bar{R}$: (0.5, 0.7, 0.9)

Contingent decision constraints: $x_{42} \leq x_{11}$ or $x_{42} \leq x_{13}$

The contingent decision constraints given in Table 2.1 implies that, if the fourth module chooses the second COTS component; that is, $sc_{42}$, then the first module must choose the first COTS component; that is, $sc_{11}$ or the third COTS component, that is, $sc_{13}$. 
2.3 An illustrative example

2.3.2 Fuzzy solutions of COTS selection model

• Lower bound calculation

To derive the lower bounds of the cost objective at different $\alpha$-cuts, model $P(2.2)$ is formulated as follows:

$$Z^L_\alpha = \min \ ((11.5 + 4\alpha)x_{11} + (12.5 + 2\alpha)x_{12} + (12 + 3\alpha)x_{13} + (2.5 + 3\alpha)x_{21}$$
$$+ (3 + 2\alpha)x_{22} + (12 + 3\alpha)x_{31} + (10 + 4\alpha)x_{32} + (12 + 2\alpha)x_{33} +$$
$$+ (10 + 2\alpha)x_{41} + (6 + 2\alpha)x_{42} + (7 + \alpha)x_{43})$$

subject to

$$d_{11}x_{11} + d_{12}x_{12} + d_{13}x_{13} \leq T,$$
$$d_{21}x_{21} + d_{22}x_{22} \leq T,$$
$$d_{31}x_{31} + d_{32}x_{32} + d_{33}x_{33} \leq T,$$
$$d_{41}x_{41} + d_{42}x_{42} + d_{43}x_{43} \leq T,$$
$$(110(0.0002x_{11} + 0.0003x_{12} + 0.0003x_{13})) + (55(0.0003x_{21} + 0.0002x_{22})))$$
$$+ (80(0.0002x_{31} + 0.0001x_{32} + 0.0004x_{33})) + (120(0.0004x_{41} + 0.0003x_{42})$$
$$+ 0.0003x_{43})) \leq R,$$
$$4 + \alpha \leq d_{11} \leq 7 - 2\alpha, \ 3 + \alpha \leq d_{12} \leq 6 - 2\alpha, \ 5 + 3\alpha \leq d_{13} \leq 10 - 2\alpha,$$
$$5 + 2\alpha \leq d_{21} \leq 10 - 3\alpha, \ 3 + 3\alpha \leq d_{22} \leq 8 - 2\alpha,$$
$$1 + 3\alpha \leq d_{31} \leq 6 - 2\alpha, \ 3 + 4\alpha \leq d_{32} \leq 9 - 2\alpha, \ 3 + 3\alpha \leq d_{33} \leq 8 - 2\alpha,$$
$$3 + 2\alpha \leq d_{41} \leq 8 - 3\alpha, \ 4 + 3\alpha \leq d_{42} \leq 10 - 3\alpha, \ 3 + 2\alpha \leq d_{43} \leq 8 - 3\alpha,$$
$$3 + 5\alpha \leq T \leq 10 - 2\alpha, \ 0.111 + 0.318\alpha \leq R \leq 1 - 0.571\alpha,$$
$$x_{11} + x_{12} + x_{13} = 1,$$
$$x_{21} + x_{22} = 1,$$
$$x_{31} + x_{32} + x_{33} = 1,$$
$$x_{41} + x_{42} + x_{43} = 1,$$
$$x_{42} - x_{11} \leq 5y_1,$$
$$x_{42} - x_{13} \leq 5y_2,$$
$$y_1 + y_2 = 1,$$
\[ y_1, y_2 \in \{0, 1\}, \]
\[ x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{31}, x_{32}, x_{33}, x_{41}, x_{42}, x_{43} \in \{0, 1\}, \]
\[ 0 \leq \alpha \leq 1. \]

The above model is solved using LINGO. For \( \alpha = 0 \) cut of \( \tilde{Z} \), the value of \( Z_{\alpha=0}^{T} \) is 30 with \( x_{11} = 1, x_{21} = 1, x_{32} = 1, x_{42} = 1, d_{11} = 4, d_{12} = 3, d_{13} = 5, d_{21} = 5, d_{22} = 3, d_{31} = 1, d_{32} = 3, d_{33} = 3, d_{41} = 3, d_{42} = 4, d_{43} = 3, T = 10, R = 0.9, y_2 = 1 \), and other decision variables are 0. Thus, COTS components \( sc_{11}, sc_{21}, sc_{32} \), and \( sc_{42} \) are selected for modules \( M1, M2, M3 \), and \( M4 \), respectively. As \( y_1 = 0 \); thus, the first constraint of the either-or constraints becomes active, whereas the second constraint is relaxed. Hence, the constraint \( x_{42} \leq x_{11} \) and \( x_{42} = 1 \) caused \( x_{11} = 1 \) in the optimal solution.

At the other extreme end of \( \alpha = 1 \), the value of \( Z_{\alpha=1}^{T} \) is 41.5 with \( x_{12} = 1, x_{22} = 1, x_{33} = 1, x_{43} = 1, d_{11} = 5, d_{12} = 4, d_{13} = 8, d_{21} = 7, d_{22} = 6, d_{31} = 4, d_{32} = 7, d_{33} = 6, d_{41} = 5, d_{42} = 7, d_{43} = 5, T = 8, R = 0.7, y_1 = 1 \), and other decision variables are 0. Thus, COTS components \( sc_{12}, sc_{22}, sc_{33} \), and \( sc_{43} \) are selected for modules \( M1, M2, M3 \), and \( M4 \), respectively. Similar interpretations hold for other cases as well. Table 2.2 lists the selection of best-fit COTS components for different modules corresponding to eleven distinct \( \alpha \) values: 0, 0.1, 0.2, \ldots, 1.0.

**Upper bound calculation**

To derive the upper bounds of the cost objective at different \( \alpha \)-cuts, model \( P(2.3) \) is formulated as follows:

\[
Z_{\alpha}^{U} = \max (u_1 + u_2 + u_3 + u_4 - T(v_1 + v_2 + v_3 + v_4) - Rw - (\rho_{11} + \rho_{12} + \rho_{13} + \rho_{21} + \rho_{22} + \rho_{31} + \rho_{32} + \rho_{33} + \rho_{41} + \rho_{42} + \rho_{43}) - \gamma_1 - \gamma_2 + \tau)
\]

subject to

\[
\lambda_1 + u_1 - d_{11}v_1 - 110 * 0.0002w - \rho_{11} \leq 18 - 2.5\alpha, \\
u_1 - d_{12}v_1 - 110 * 0.0003w - \rho_{12} \leq 16.5 - 2\alpha, \\
\lambda_2 + u_1 - d_{13}v_1 - 110 * 0.0003w - \rho_{13} \leq 18 - 3\alpha, \\
u_2 - d_{21}v_2 - 55 * 0.0003w - \rho_{21} \leq 7 - 1.5\alpha, \\
u_2 - d_{22}v_2 - 55 * 0.0002w - \rho_{22} \leq 8 - 3\alpha,
\]
\[ u_3 - d_{31} v_3 - 80 \times 0.0002 w - \rho_{31} \leq 18 - 3\alpha, \]
\[ u_3 - d_{32} v_3 - 80 \times 0.0001 w - \rho_{32} \leq 17 - 3\alpha, \]
\[ u_3 - d_{33} v_3 - 80 \times 0.0004 w - \rho_{33} \leq 18 - 4\alpha, \]
\[ u_4 - d_{41} v_4 - 120 \times 0.0004 w - \rho_{41} \leq 14 - 2\alpha, \]
\[ -\lambda_1 - \lambda_2 + u_4 - d_{42} v_4 - 120 \times 0.0003 w - \rho_{42} \leq 11 - 3\alpha, \]
\[ u_4 - d_{43} v_4 - 120 \times 0.0003 w - \rho_{43} \leq 9 - \alpha, \]
\[ \tau + 5\lambda_1 - \gamma_1 \leq 0, \]
\[ \tau + 5\lambda_2 - \gamma_2 \leq 0, \]
\[ 4 + \alpha \leq d_{11} \leq 7 - 2\alpha, \]
\[ 3 + \alpha \leq d_{12} \leq 6 - 2\alpha, \]
\[ 5 + 3\alpha \leq d_{13} \leq 10 - 2\alpha, \]
\[ 5 + 2\alpha \leq d_{21} \leq 10 - 3\alpha, \]
\[ 3 + 3\alpha \leq d_{22} \leq 8 - 2\alpha, \]
\[ 1 + 3\alpha \leq d_{31} \leq 6 - 2\alpha, \]
\[ 3 + 4\alpha \leq d_{32} \leq 9 - 2\alpha, \]
\[ 3 + 3\alpha \leq d_{33} \leq 8 - 2\alpha, \]
\[ 3 + 2\alpha \leq d_{41} \leq 8 - 3\alpha, \]
\[ 4 + 3\alpha \leq d_{42} \leq 10 - 3\alpha, \]
\[ 3 + 2\alpha \leq d_{43} \leq 8 - 3\alpha, \]
\[ 3 + 5\alpha \leq T \leq 10 - 2\alpha, \]
\[ 0.111 + 0.318\alpha \leq R \leq 1 - 0.571\alpha, \]
\[ w \geq 0, \quad v_i \geq 0, \quad \gamma_k \geq 0, \quad \lambda_k \geq 0, \quad i = 1, 2, 3, 4, \quad k = 1, 2, \]
\[ \rho_{11}, \quad \rho_{12}, \quad \rho_{13}, \quad \rho_{21}, \quad \rho_{22}, \quad \rho_{31}, \quad \rho_{32}, \quad \rho_{33}, \quad \rho_{41}, \quad \rho_{42}, \quad \rho_{43} \geq 0, \]
\[ 0 \leq \alpha \leq 1. \]

The above model is solved using LINGO. For \( \alpha = 0 \) cut of \( \tilde{Z} \), the value of \( Z_\alpha^{U=0} \) is 49.5 with \( x_{12} = 1, \ x_{21} = 1, \ x_{32} = 1, \ x_{43} = 1, \ d_{11} = 7, \ d_{12} = 6, \ d_{13} = 10, \ d_{21} = 10, \ d_{22} = 8, \ d_{31} = 6, \ d_{32} = 9, \ d_{33} = 8, \ d_{41} = 8, \ d_{42} = 10, \ d_{43} = 8, \ T = 10, \ R = 0.9, \ u_1 = 16.5, \ u_2 = 7, \ u_3 = 17, \ u_4 = 9, \) and other decision variables are 0. Thus, COTS components \( sc_{12}, \ sc_{21}, \ sc_{32}, \) and \( sc_{43} \) are selected for modules \( M1, \ M2, \ M3, \) and \( M4, \) respectively. At the other extreme end of \( \alpha = 1, \) the value of \( Z_\alpha^{U=1} \) is 41.5 with \( x_{12} = 1, \ x_{22} = 1, \ x_{33} = 1, \ x_{43} = 1, \ d_{11} = 5, \ d_{12} = 4, \ d_{13} = 8, \ d_{21} = 7, \ d_{22} = 6, \ d_{31} = 4, \ d_{32} = 7, \ d_{33} = 6, \ d_{41} = 5, \ d_{42} = 7, \ d_{43} = 5, \ T = 8, \ R = 0.7, \ u_1 = 14.5, \ u_2 = 5, \ u_3 = 14, \ u_4 = 8, \) and other decision variables are 0. Thus, COTS components \( sc_{12}, \ sc_{22}, \ sc_{33}, \) and \( sc_{43} \) are selected for modules \( M1, \ M2, \ M3, \) and \( M4, \) respectively. Similar interpretations hold for other cases as well. Table 2.2 lists the selection of best COTS components for different modules corresponding to eleven distinct
α values: 0, 0.1, 0.2, \ldots, 1.0.

The possibility level α represents the possibility that the total system cost will appear in the associated range. Specifically, the α = 1 cut shows the cost that is most likely to be and the α = 0 cut shows the range that the total cost could be. In this case study, while the total cost of the software system is fuzzy, its most likely value falls at 41.5, and its value is impossible to fall outside the range of 30 and 49.5, see Table 2.2. This additional information might prove beneficial to the DM when deciding the selection of COTS components.

Table 2.2: COTS selection corresponding to various α-cuts of the total software system cost

<table>
<thead>
<tr>
<th>α</th>
<th>( Z^L_\alpha )</th>
<th>COTS components selected</th>
<th>( Z^U_\alpha )</th>
<th>COTS components selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>30 ( sc_{11} )</td>
<td>( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td>49.5 ( sc_{12} ) ( sc_{21} ) ( sc_{32} ) ( sc_{43} )</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>31.3 ( sc_{11} )</td>
<td>( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td>48.75 ( sc_{12} ) ( sc_{21} ) ( sc_{32} ) ( sc_{43} )</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>32.6 ( sc_{11} )</td>
<td>( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td>48 ( sc_{12} ) ( sc_{21} ) ( sc_{32} ) ( sc_{43} )</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>33.9 ( sc_{11} )</td>
<td>( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td>47.25 ( sc_{12} ) ( sc_{21} ) ( sc_{32} ) ( sc_{43} )</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>35.2 ( sc_{11} )</td>
<td>( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td>46.5 ( sc_{12} ) ( sc_{21} ) ( sc_{32} ) ( sc_{43} )</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>36.5 ( sc_{13} )</td>
<td>( sc_{22} ) ( sc_{32} ) ( sc_{42} )</td>
<td>45.75 ( sc_{11} ) ( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>37.6 ( sc_{13} )</td>
<td>( sc_{22} ) ( sc_{32} ) ( sc_{42} )</td>
<td>45 ( sc_{11} ) ( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>38.7 ( sc_{13} )</td>
<td>( sc_{22} ) ( sc_{32} ) ( sc_{42} )</td>
<td>44.2 ( sc_{11} ) ( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>39.7 ( sc_{12} )</td>
<td>( sc_{22} ) ( sc_{32} ) ( sc_{43} )</td>
<td>43.3 ( sc_{13} ) ( sc_{22} ) ( sc_{32} ) ( sc_{42} )</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>40.6 ( sc_{12} )</td>
<td>( sc_{22} ) ( sc_{32} ) ( sc_{43} )</td>
<td>42.4 ( sc_{13} ) ( sc_{22} ) ( sc_{32} ) ( sc_{42} )</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>41.5 ( sc_{12} )</td>
<td>( sc_{22} ) ( sc_{33} ) ( sc_{43} )</td>
<td>41.5 ( sc_{12} ) ( sc_{22} ) ( sc_{33} ) ( sc_{43} )</td>
<td></td>
</tr>
</tbody>
</table>

- Membership function of \( \tilde{Z} \)

The membership function of \( \tilde{Z} \) can now be derived as

\[
\mu_{\tilde{Z}}(z') = \begin{cases} 
\frac{z' - 30}{11.5}, & 30 \leq z' \leq 41.5, \\
1, & z' = 41.5, \\
\frac{49.5 - z'}{8}, & 41.5 \leq z' \leq 49.5.
\end{cases} 
\] (2.15)

The associated membership function of \( \tilde{Z} \) is depicted approximately in Figure
2.2. It may be noted that the shape of $\mu_{\tilde{Z}}(z')$ is similar to that of the fuzzy cost coefficients represented by TFNs; however, it just look likes triangular, yet it is actually not.

![Figure 2.2](image)

Figure 2.2: Membership function of $\tilde{Z}$ using the proposed approach

- **Crisp transformation**

By taking $n = 100$, the representative crisp value $Z^*$ of the fuzzy total cost obtained using (2.13) and (2.14) is 40.33333.

2.3.3 **Comparison with ranking function approach**

We compare the results obtained from the proposed approach with the one obtained by using ranking function approach applied to model P(2.1). Each of the fuzzy coefficient in model P(2.1) is represented by a TFN given as $\tilde{A} = (a_l, a_m, a_u)$, where $a_l$ and $a_u$ are the lower and upper bounds of the support of $\tilde{A}$ and $a_m$ is the modal value. By the actual substitution of the membership function of the TFN, the ranking function approach [113] provides crisp equivalent of $\tilde{A}$ given by

$$A = \frac{a_l + a_m + a_u}{3}. \quad (2.16)$$

Using (2.16), we transform each of the fuzzy coefficients in the constraints of model P(2.1) into crisp ones. The fuzzy optimization model with fuzzy cost coefficients and crisp constraints is formulated as follows:
\[
\begin{align*}
\text{min } & \quad (11.5, 15.5, 18)x_{11} + (12.5, 14.5, 16.5)x_{12} + (12, 15, 18)x_{13} + (2.5, 5.5, 7)x_{21} \\
& \quad + (3.5, 8)x_{22} + (12, 15, 18)x_{31} + (10, 14, 17)x_{32} + (12, 14, 18)x_{33} \\
& \quad + (10, 12, 14)x_{41} + (6, 8, 11)x_{42} + (7, 8, 9)x_{43} \\
\text{subject to } & \quad 5.333x_{11} + 4.333x_{12} + 7.667x_{13} \leq 7, \\
& \quad 7.333x_{21} + 5.667x_{22} \leq 7, \\
& \quad 3.667x_{31} + 6.333x_{32} + 5.667x_{33} \leq 7, \\
& \quad 5.333x_{41} + 7x_{42} + 5.333x_{43} \leq 7, \\
& \quad (110(0.0002x_{11} + 0.0003x_{12} + 0.0003x_{13})) + (55(0.0003x_{21} + 0.0002x_{22})) \\
& \quad + (80(0.0002x_{31} + 0.0001x_{32} + 0.0004x_{33})) + (120(0.0004x_{41} + 0.0003x_{42} \\
& \quad + 0.0003x_{43})) \leq 0.513, \\
& \quad x_{11} + x_{12} + x_{13} = 1, \\
& \quad x_{21} + x_{22} = 1, \\
& \quad x_{31} + x_{32} + x_{33} = 1, \\
& \quad x_{41} + x_{42} + x_{43} = 1, \\
& \quad x_{42} - x_{11} \leq 5y_1, \\
& \quad x_{42} - x_{13} \leq 5y_2, \\
& \quad y_1 + y_2 = 1, \\
& \quad y_1, y_2 \in \{0, 1\}, \\
& \quad x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{31}, x_{32}, x_{33}, x_{41}, x_{42}, x_{43} \in \{0, 1\}. 
\end{align*}
\]

The three representative corner points of the triangular fuzzy cost coefficients in the above model are used to form three crisp LPPs. The solutions of these three problems define a membership function of the fuzzy total cost as shown in Figure 2.3. Corresponding to the two extreme ends \(\alpha = 0\) and \(\alpha = 1\) cuts from Table 2.2, it can be seen clearly that the results obtained from crisp model overestimate the fuzzy minimal cost \(\tilde{Z}\) obtained using the proposed solution approach.
2.4 Concluding remarks

In real-world applications of COTS selection, input data and/or related model parameters are often uncertain in non-stochastic sense because much of the information is incomplete or unavailable over the planning horizon. This chapter proposed a solution method for the uncertain COTS selection problem whose imprecise parameters are expressed by TFNs. The solution approach consists of using Zadeh’s extension principle to transform fuzzy model into two conventional crisp mathematical programming problems parameterized by possibility level $\alpha$. The resulting fuzzy solutions of the two problems at different possibility levels $\alpha$ can provide the DM enough flexibility to meet his/her goals. The effectiveness of the fuzzy optimization model and the solution approach is demonstrated by performing numerical experiments on a real-world case study of COTS-based software development. The proposed approach can also be used if the uncertain data of the COTS selection problem are represented using TrFNs. If the imprecise numbers are represented using general functional fuzzy forms, one would need to approximate the nonlinear forms first and then use the proposed approach. The membership function of the fuzzy total cost is derived numerically using $\alpha$-cuts. Consequently, the membership degree of a specific objective value must be approximated from the $\alpha$-cuts. A challenging task would be to derive the mathematical form of the membership function so that the membership degree can be calculated directly.