Chapter 3

Multiobjective COTS Selection in Fuzzy Environment
The research work presented in this chapter relies on multiple methodologies such as quality models, AHP, and fuzzy mathematical programming for developing fuzzy multiobjective optimization models of COTS selection. We formulate the COTS selection problem by considering the following challenges associated with it: (1) need for hierarchical decision making; (2) availability of alternative COTS components; (3) multiple COTS selection objectives; and (4) selection of one and only one COTS component for each module. To deal with the uncertainty in the COTS selection process, we propose fuzzy multiobjective optimization models. The models belong to the category of flexible programming where fuzzy goals and constraints represent the flexibility in the target values of objective functions and the elasticity of the constraints. The fuzzy goals and constraints in this chapter have been characterized by using both linear and nonlinear membership functions.

This chapter is divided into two sections. In both the sections, we consider the design of a modular software system. The software will perform one or more functions, as is the requirement. A program performs each function. Each program consists of a number of modules which are executed sequentially. Some of these modules are common to different programs. Also, we assume that different alternative COTS components are available for each module. The diagrammatic depiction of such a software system is given in Figure 3.1 that consists of $P$ programs, $m$ modules and several alternative COTS components, for example, COTS components $s_{c11}, s_{c12}, s_{c13}, \ldots, s_{c1m}$ are available to implement the first module and so on.

The hierarchy shown in Figure 3.1 links the user’s view (program level) with the software developer’s view (module level). It may be noted that each module in the above software system has different levels of importance that depend on access frequency. The module which is called more frequently by a program may be relatively more important than the other module(s) whose frequency of calling within a system is less. Users often recognize the quality of a software system based on the quality of the modules. Hence, purchasing expensive high-quality COTS components may be justified by the frequent use of the module. Literature on software reliability suggests two important developments that encourages modeling reliability (one of the attributes of quality) at the modular level. The first is the concept of reusable module [82]. Once a function or
procedure is written, it goes into a library and could be accessed for other pro-
gramming purposes. An error in a module could propagate and contaminate
more than one set of programs. The second concept is the object oriented pro-
gramming paradigm, where one class of objects combines procedure and data
into one independent module [98]. Since classes of objects model physical do-
 mains, they could be used in more than one program. Hence, reliability of a
coded class of objects would be of great interest.

In general, Standards for software quality models define software quality charac-
teristics as composed of six external attributes of interest, namely, functionality,
reliability, usability, maintainability, portability and efficiency. Functionality of
the COTS component is nothing but the ability of the component to perform
according to the specific needs of the organization. Reliability is ability of the
COTS component to run consistently without crashing under specific condi-
tions. It is used to measure the reliability level to which the system satisfies the
functional requirements of the organization. Usability is understandability of
the COTS component as well as easiness to learn and operate it under certain
specific conditions. Maintainability is the ability of the COTS component to be

![Hierarchy of a COTS-based software system](image)

**Figure 3.1:** Hierarchy of a COTS-based software system
modified. Modifications can include corrections, improvements, or adaptation of the software to adjust to changes in the environment in terms of functional requirements and specifications. Portability is the ability of the COTS component to be transferred from one environment to another. Efficiency is ability of the COTS component to provide appropriate performance relative to the amount of resources used under certain conditions. In turn, each of these criteria is decomposed into subcriteria. The decision hierarchy used for quality level estimation of COTS components is depicted in Figure 3.2. The quality levels determined using pairwise comparisons provide values ranging from 0 to 1 where 1 refers to a very high quality level.

![Hierarchy of quality criteria for COTS evaluation](image)

**Figure 3.2:** Hierarchy of quality criteria for COTS evaluation

The notation used in the formulation of fuzzy multiobjective optimization models are as follows:
$i$: 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,

$P$: the number of programs in the software system,

$n_i$: the number of alternative COTS components in the $i$-th module, $i = 1, 2, \ldots, m$,

$w_i$: the weight of the $i$-th module calculated using AHP, $i = 1, 2, \ldots, m$,

$\nu_p$: the probability of use of the $p$-th program, $p = 1, 2, \ldots, P$,

$s_i$: the average number of invocations of the $i$-th module, $i = 1, 2, \ldots, m$,

$S_p$: the set of modules corresponding to the $p$-th program, $p = 1, 2, \ldots, P$,

$q_{ij}$: the quality level of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,

$c_{ij}$: the cost of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,

$l_{ij}$: the size (number of lines of code) of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,

$t_{ij}$: the expected execution time of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,

$d_{ij}$: the delivery time of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,

$\mu_{ij}$: the probability of failure on demand of the $j$-th COTS component for the $i$-th module, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_i$,

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

$$x_{ij} = \begin{cases} 
1, & \text{if the } j \text{-th COTS component of the } i \text{-th module is chosen,} \\
0, & \text{otherwise.}
\end{cases}$$
3.1 COTS selection using hybrid approach

In this section, we formulate a fuzzy biobjective optimization model for COTS selection based on vague aspiration levels of the DM in respect of the weighted quality and cost of COTS components. The model is constrained by limitations including the incompatibility among COTS components and the selection of one and only one COTS component for each module. To express vague aspiration levels of the DM, various types of membership functions have been proposed in the literature. Here, we use a nonlinear S-shape membership function to represent the vague aspiration levels of DM's weighted quality and cost. We use Bellman-Zadeh's maximization principle (the max-min approach) to formulate the fuzzy biobjective optimization model under the assumption that both the fuzzy objectives are treated equivalently. To reflect the relative importance of the DM for the two objectives, a weighted additive model is also presented by assigning different weights to the two objectives. Furthermore, the efficiency of the obtained solutions is verified by using a two-phase approach. The two-phase approach guarantees fuzzy-efficient solution and improves, if possible, the solution obtained by using the max-min approach and the weighted additive approach. The main advantage of the proposed models is that if the DM is not satisfied with any of the COTS selection obtained, more COTS selections can be generated by varying values of shape parameters of nonlinear S-shape membership functions. With the help of numerical illustrations, we demonstrate the efficiency of the proposed model.

The rest of the section is organized as follows. In Section 3.1.1, we formulate the biobjective optimization model for COTS selection, Section 3.1.2 presents the fuzzy mathematical programming models for COTS selection using nonlinear S-shape membership functions. Section 3.1.3 deals with numerical illustrations of the fuzzy biobjective optimization models for COTS selection of a modular software system and discusses the results thus obtained. Finally, some concluding remarks are made in Section 3.1.4.

3.1.1 Biobjective optimization model for COTS selection

For selecting best-fit COTS components, DMs usually consider the following problems: (1) maximize the system quality subject to a cost constraint; or (2)
minimize the cost under system quality constraint. In the proposed biobjective optimization model for COTS selection, we consider the following objectives and constraints.

**Objectives**

We consider two important and conflicting objectives for COTS selection problem: the weighted quality $Q(x)$ and the total cost of purchasing $C(x)$.

- **Weighted quality**

  As discussed in Jung and Choi [54], we consider the overall quality as a single entity. The overall quality of different possible alternatives of each module is computed by combining individual measures of quality listed in Figure 3.2. Furthermore, the individual measures of quality may conflict each other and comparisons may have to be reached, hence, the overall quality measure is an optimum balance of factors rather than an ideal way of representing quality. The objective of maximizing the weighted quality for COTS selection problem is expressed as

  $$\max Q(x) = \sum_{i=1}^{m} w_i \left( \sum_{j=1}^{n_i} q_{ij} x_{ij} \right).$$

- **Cost**

  We consider the cost criterion which is based on procurement and adaptation costs of COTS components. The procurement cost contains licensing arrangement cost, product and technology cost, and consulting cost. The objective of minimizing the total cost for COTS selection problem is expressed as

  $$\min C(x) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij}.$$ 

**Constraints**

- **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

\[ x_{rs} - x_{ut_k} \leq My_k, \ k = 1, 2, \ldots, z, \]
\[ \sum_{k=1}^{z} y_k = z - 1, \]
\[ y_k \in \{0, 1\}, \ k = 1, 2, \ldots, z. \]

• Selection or rejection of a COTS component

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

The decision model

The biobjective optimization model for COTS selection which maximizes the weighted quality and minimizes the cost is formulated as follows:

\[
P(3.1.1) \quad \max Q(x) = \sum_{i=1}^{m} w_i \left( \sum_{j=1}^{n_i} q_{ij} x_{ij} \right) \\
\min C(x) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \\
\text{subject to} \\
\sum_{j=1}^{n_i} x_{ij} = 1, \ i = 1, 2, \ldots, m, \quad (3.1.1) \\
x_{rs} - x_{ut_k} \leq My_k, \ k = 1, 2, \ldots, z, \quad (3.1.2) \\
\sum_{k=1}^{z} y_k = z - 1, \quad (3.1.3) \\
y_k \in \{0, 1\}, \ k = 1, 2, \ldots, z, \quad (3.1.4) \\
x_{ij} \in \{0, 1\}, \ i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n_i. \quad (3.1.5)
\]

3.1.2 COTS selection models based on fuzzy set theory

To express vague aspiration levels of the DM, various types of membership functions have been proposed in literature such as linear membership function [119, 120], tangent-type membership function [71], exponential membership function [131, 132], trapezoidal membership function [129], sigmoid membership function [130], and so on.
A fuzzy LPP with nonlinear membership functions results in an NLPP. Usually, a linear membership function is employed in order to avoid nonlinearity. Nevertheless, certain decision situations may prefer nonlinear membership functions. Also, if membership function is interpreted as the fuzzy utility of the DM, which describes the behavior of indifference, preference or aversion towards uncertainty, a nonlinear membership function may be a better representation than a linear membership function.

We use a logistic-type membership function \([111]\), that is, a nonlinear S-shape membership function to express vague aspiration levels of the DM. The S-shape membership function is given by

\[
f(x) = \frac{1}{1 + \exp(-\alpha x)},
\]

where \(\alpha, 0 < \alpha < \infty\), is a fuzzy parameter which measures the degree of vagueness. The reason why we use this function is that, the logistic-type membership function has similar shape as that of tangent hyperbolic function, but it is more easily handled than the tangent hyperbolic function. Also, the logistic-type membership function preserves linearity even when the operator ‘product’ is used instead of the operator ‘min’ to aggregate the overall satisfaction to arrive at the fuzzy decision. Furthermore, it is known that a trapezoidal membership function is an approximation to a logistic function.

In model P(3.1.1), the two objectives, that is, the weighted quality and the cost are considered to be vague and uncertain. We use the following nonlinear S-shape membership functions to express the vague aspiration levels of the DM’s weighted quality and cost of the software system.

- The membership function of the goal for the weighted quality is given by

\[
\mu_Q(x) = \frac{1}{1 + \exp\left(-\alpha_Q \left(\sum_{i=1}^{m} w_i \left(\sum_{j=1}^{n_i} q_{ij} x_{ij}\right) - Q_m\right)\right)},
\]

where \(Q_m\) is the mid-point (middle aspiration level for the weighted quality) at which the membership function value is 0.5 and \(\alpha_Q\) can be given by the DM based on his own degree of satisfaction for the weighted quality (see Figure 3.1.1).
The membership function of the goal for the cost is given by

\[
\mu_C(x) = \frac{1}{1 + \exp \left( \alpha_C \left( \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} - C_m \right) \right)},
\]

where \( C_m \) is the mid-point (middle aspiration level for the cost) at which the membership function value is 0.5 and \( \alpha_C \) can be given by the DM based on his own degree of satisfaction for the cost (see Figure 3.1.2).

Remark 3.1.1. The shape parameters \( \alpha_Q \) and \( \alpha_C \) determine shapes of the membership functions \( \mu_Q(x) \) and \( \mu_C(x) \), respectively, and are selected in the range \( (0, \infty) \). The larger these parameters get, the less their vagueness becomes.
Remark 3.1.2. The mid-points $Q_m$ and $C_m$ are determined by taking $Q_m = \frac{Q_N + Q_S}{2}$ and $C_m = \frac{C_N + C_S}{2}$. Here, $Q_N$ and $C_N$ are the necessity levels, and $Q_S$ and $C_S$ are the sufficiency levels indicated by the DM. It may be noted that while linear membership functions such as triangular or trapezoidal functions show a necessity level and a sufficiency level at 0 and 1, respectively, on the other hand, for the logistic function on the other hand, a necessity level and/or a sufficiency level may be approximated.

Following Bellman-Zadeh’s maximization principle and using the above defined fuzzy membership functions, the fuzzy biobjective optimization model for COTS selection is formulated as follows:

$$
\text{P}(3.1.2) \quad \max \lambda \\
\text{subject to} \\
\lambda \leq \mu_Q(x), \\
\lambda \leq \mu_C(x), \\
0 \leq \lambda \leq 1, \\
\text{and Constraints (3.1.1)–(3.1.5)}.
$$

The model P(3.1.2) is an NLPP. It can be transformed into an LPP by letting $\theta = \log \frac{\lambda}{1-\lambda}$, so that $\lambda = \frac{1}{1 + \exp(-\theta)}$. Since, the logistic function is monotonically increasing, maximizing $\lambda$ makes $\theta$ maximize. Therefore, model P(3.1.2) can be transformed into an equivalent LPP as follows:

$$
\text{P}(3.1.3) \quad \max \theta \\
\text{subject to} \\
\theta \leq \alpha_Q \left( \sum_{i=1}^{m} w_i \left( \sum_{j=1}^{n_i} q_{ij} x_{ij} \right) - Q_m \right), \\
\theta \leq \alpha_C \left( C_m - \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \right), \\
\theta \geq 0, \\
\text{and Constraints (3.1.1)–(3.1.5)}.
$$

To formulate the fuzzy biobjective optimization model P(3.1.2)/P(3.1.3), we
have used the min-operator to find the fuzzy decision that simultaneously satisfies all the fuzzy objectives. Then the maximizing decision is determined to be the maximum degree of membership for the fuzzy decision. This approach considers that the relationship between various objectives in a fuzzy environment is fully symmetric [120]; that is, all the fuzzy objectives are treated equivalently. Although this approach is efficient in computation, the application of it may produce uniform membership degrees for fuzzy objectives when the achievement of some objectives is stringently required. Depending on the COTS component selection problem, situations wherein fuzzy objectives have unequal importance to the DM, may also be considered. Several approaches have been developed to deal with situations in which the objectives are not equally important. Narasimhan [85] used fuzzy weights approach in which membership functions that represent linguistic priorities are defined on goal values. Here the fuzzy weights represent only the relative importance of the goal values of a certain objective rather than the relative importance of different objectives. Another model that takes into account weights of the objectives is the weighted additive model of Tiwari et al. [104]. They used different weights for the various objectives in order to reflect the relative importance of the objectives, and considered the weights as the coefficients of the objective function. The weighted additive model for COTS selection in a fuzzy environment with unequal importance between the two objectives is written as follows:

\[
\text{P}(3.1.4) \quad \max \sum_{h=1}^{2} \omega_h \lambda_h
\]

subject to
\[
\begin{align*}
\lambda_1 & \leq \mu_Q(x), \\
\lambda_2 & \leq \mu_C(x), \\
0 & \leq \lambda_h \leq 1, \ h = 1, 2,
\end{align*}
\]

and Constraints (3.1.1)–(3.1.5),

where \( \omega_h \) is the relative weight of the \( h \)-th objective function to be specified by the DM such that \( \omega_h > 0 \) and \( \sum_{h=1}^{2} \omega_h = 1 \). The model \( \text{P}(3.1.4) \) is an NLPP which can be solved using LINGO.

The max-min approach used in the formulation of the models \( \text{P}(3.1.2) / \text{P}(3.1.3) \)
and P(3.1.4) possesses some good properties; however, the efficiency of the solution yielded by max-min approach is not guaranteed. To overcome inefficiency, the compromise approach [112] and the two-phase approach [75] have been proposed in literature. Using compromise approach, it is possible for a DM to choose explicitly a desirable achievement degree for each fuzzy objective function. We add a set of desirable achievement degrees as constraints, that is, \( \lambda_h \geq \beta_h \), where \( \beta_h \) is the desirable achievement degree for the \( h \)-th fuzzy objective function, to the constraints of model P(3.1.3) or P(3.1.4). To increase the desirable achievement degree of one objective function means that the value of this objective function is more close to the optimal value but it may result in other objective function values far from their optimal values because of multiobjective nature of the problem. As a result, when the DM requires a very high desirable achievement degree for each fuzzy objective function, there may be no feasible solution. To overcome the difficulty of selecting proper desirable achievement degree for each fuzzy objective function, Li et al. [75] proposed a two-phase approach in which the desirable achievement degree in compromise approach is taken equal to the degree of satisfaction corresponding to the solution of the max-min approach. Consequently, in the two-phase approach, we solve models P(3.1.5) and P(3.1.6) corresponding to models P(3.1.3) and P(3.1.4), respectively, to get an efficient solution of the original crisp model P(3.1.1) for COTS selection.

\[
P(3.1.5) \quad \max \sum_{h=1}^{2} \omega_h \theta_h
\]

subject to

\[
\log \frac{\mu_Q(x^*)}{1 - \mu_Q(x^*)} \leq \theta_1 \leq \alpha_Q \left( \sum_{i=1}^{m} w_i \left( \sum_{j=1}^{n_i} q_{ij} x_{ij} \right) - Q_m \right),
\]

\[
\log \frac{\mu_C(x^*)}{1 - \mu_C(x^*)} \leq \theta_2 \leq \alpha_C \left( C_m - \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \right),
\]

\( \theta_h \geq 0, \ h = 1, 2, \)

and Constraints (3.1.1)–(3.1.5),

where \( x^* \) is an optimal solution of model P(3.1.3), \( \omega_1 = \omega_2 > 0 \) and \( \sum_{h=1}^{2} \omega_h = 1. \)
\[ \text{P(3.1.6)} \quad \max \sum_{h=1}^{2} \omega_h \lambda_h \]

subject to

\[ \mu_Q(x^{**}) \leq \lambda_1 \leq \mu_Q(x), \]
\[ \mu_C(x^{**}) \leq \lambda_2 \leq \mu_C(x), \]
\[ 0 \leq \lambda_h \leq 1, \quad h = 1, 2, \]

and Constraints (3.1.1)–(3.1.5),

where \( x^{**} \) is an optimal solution of model P(3.1.4). Note that model P(3.1.5) is an LPP while model P(3.1.6) is an NLPP.

### 3.1.3 An illustrative example

**System prototype**

Here, we consider a prototype of the software system consisting of four modules \((m=4)\) reproduced from Jung and Choi [54]. Figure 3.1.3 shows the system prototype architecture that consists of three programs, four modules, and eleven COTS components denoted by \( sc_{11}, sc_{12}, \ldots, sc_{43} \). The cost of COTS components is the purchasing price provided by the vendor and their quality level can be estimated using any of the quality models available in the literature such as ISO/IEC 9126 [49] or others. Here, the ISO/IEC 9126 quality model has been used to assign weights of quality characteristics (attributes) in order to evaluate overall quality of the software system. The weights lies between 0 and 1. The data for cost and quality of this system prototype is listed in Table 3.1.1.
### Table 3.1.1: Input data of the system prototype for model P(3.1.1)

<table>
<thead>
<tr>
<th>Module 1 (M1)</th>
<th>Module 2 (M2)</th>
<th>Module 3 (M3)</th>
<th>Module 4 (M4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>Cost</td>
<td>Quality</td>
<td>Cost</td>
</tr>
<tr>
<td>$q_{11} = 0.83$</td>
<td>$c_{11} = 10$</td>
<td>$q_{21} = 0.85$</td>
<td>$c_{21} = 8$</td>
</tr>
<tr>
<td>$q_{12} = 0.82$</td>
<td>$c_{12} = 9$</td>
<td>$q_{22} = 0.88$</td>
<td>$c_{22} = 9$</td>
</tr>
<tr>
<td>$q_{13} = 0.78$</td>
<td>$c_{13} = 8$</td>
<td>$q_{33} = 0.90$</td>
<td>$c_{33} = 9$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

The contingent decision constraints in Table 3.1.1 implies that if the fourth module chooses the second COTS component, then the first module must choose the first COTS component or the third COTS component.

### Assigning weights to modules

In the literature there are various methods for assigning weights to modules. According to the access frequency of each module, we apply the AHP method to assign a weight to each module. Using the hierarchy shown in Figure 3.1, we
obtain weight parameters for the modules according to their access frequencies and also using software developer’s inputs regarding technical specifications of the software system. The use of access frequency in determining the importance of modules is justified from the work of Anderson [5] on software selection.

The hierarchical basis of the system prototype

We begin the hierarchy from the top with the DM’s view, which we define as the overall assessment of the quality of the software system. The DM’s assessment is based on the functionality of several programs of the software system, which are represented at the second level of the hierarchy. The system prototype architecture shown in Figure 3.1.3 consists of three programs denoted by Program 1, Program 2, and Program 3. The third level of the hierarchy consists of the independent modules of which the programs are composed of. Here, we assume that the software system uses four modules denoted by $M_1$, $M_2$, $M_3$, and $M_4$. Here, we consider modules as independent units which themselves may have submodules but each submodule belongs to only one module, that is, a many-to-one relationship. We stop the hierarchical structure at the level of the independent modules.

DM’s relative preferences for modules

Our goal is to identify the relative importance of each program at the second level and each module at the third level of the hierarchy in the DM’s assessment of the quality of the software system at the first level. Clearly, one is unable to ask directly from the DM to express his preference for each program and module because his view of the software is an external one. Hence, to achieve our goal, we use AHP method for identifying the relative importance of the programs and the modules.

An illustrative pairwise comparison matrix of the attributes (Program 1, Program 2, and Program 3) with respect to the goal; that is, quality of the software system is shown in Table 3.1.2. As seen in this table, the preferences of quality of the software system with respect to the functionality of the programs are: for Program 1–0.411, for Program 2–0.261, and for Program 3–0.328.
**Table 3.1.2:** Comparison matrix of the programs with respect to quality

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Program 1</th>
<th>Program 2</th>
<th>Program 3</th>
<th>Relative preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program 1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.411</td>
</tr>
<tr>
<td>Program 2</td>
<td>1/2</td>
<td>1</td>
<td>1</td>
<td>0.261</td>
</tr>
<tr>
<td>Program 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.328</td>
</tr>
</tbody>
</table>

Consistency ratio: $0.0534 < 0.1$.

Since Program 1 require modules $M_1$ and $M_2$, the pairwise comparison matrix of modules $M_1$ and $M_2$ with respect to Program 1 is shown in Table 3.1.3.

**Table 3.1.3:** Comparison matrix of the modules with respect to Program 1(attribute)

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>Relative preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>1</td>
<td>2</td>
<td>0.667</td>
</tr>
<tr>
<td>$M_2$</td>
<td>1/2</td>
<td>1</td>
<td>0.333</td>
</tr>
</tbody>
</table>

On the same lines the pairwise comparison matrix of the modules $M_2$ and $M_3$ with respect to Program 2 and modules $M_1$, $M_3$, and $M_4$ with respect to Program 3 are shown in Tables 3.1.4 and 3.1.5, respectively.

**Table 3.1.4:** Comparison matrix of the modules with respect to Program 2(attribute)

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>Relative preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_2$</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>$M_3$</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 3.1.5:** Comparison matrix of the modules with respect to Program 3(attribute)

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>$M_1$</th>
<th>$M_3$</th>
<th>$M_4$</th>
<th>Relative preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0.429</td>
</tr>
<tr>
<td>$M_3$</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>0.143</td>
</tr>
<tr>
<td>$M_4$</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0.429</td>
</tr>
</tbody>
</table>

Consistency ratio: $0 < 0.1$. 

After finding all the relative weights, the global (final) weight of each module is determined by following what in terms of the AHP hierarchy may be regarded as a bottom-up process of successive multiplication. For example, the relative weights of module $M_1$ in relation to the attributes–Program 1 and Program 3, are multiplied with the corresponding relative weights of these attributes in relation to their parent attribute, that is, the overall goal of the quality of the software system. The final AHP weight of module $M_1$ is then determined by adding these products, that is, the final AHP weight of module $M_1$ is obtained as: $0.667 \times 0.411 + 0.429 \times 0.328 = 0.415$. Similarly, the final weights of modules $M_2$, $M_3$, and $M_4$ can be calculated. The AHP result is listed in the following weight vector

$$W = (0.415, 0.268, 0.177, 0.140).$$

It may be noted that in the above weight vector, the modules having same access frequency are not at the same level of importance because we have combined access frequency with the inputs of the DM in pairwise comparisons. Furthermore, it may be pointed out that when the number of programs or modules is very large at a given level of hierarchy, we can use absolute rating in place of pairwise comparisons to obtain the weights. However, absolute rating of elements is less accurate than pairwise comparisons in eliciting the preferences of the DM.

**COTS selection using optimization model P(3.1.3) and sensitivity analysis with respect to changes in the middle aspiration level of the two objectives**

Initially, we set $Q_m = 0.75$, $C_m = 32.5$, $\alpha_Q = 6$, $\alpha_C = 60$, and use module weight vector, $W = (0.415, 0.268, 0.177, 0.140)$. Using this data, the data given in Table 3.1.1 and by considering that fuzzy goals of the weighted quality and the cost are represented using nonlinear $S$-shape membership functions, model P(3.1.3) is formulated as follows:

$$\max \theta$$

subject to

$$6(0.415(0.83x_{11} + 0.82x_{12} + 0.78x_{13}) + 0.268(0.85x_{21} + 0.88x_{22}) +$$

$$0.177(0.85x_{31} + 0.79x_{32} + 0.90x_{33}) + 0.140(0.90x_{41} + 0.88x_{42} +$$

$$0.81x_{43}) - 0.75) \geq \theta,$$
3.1 COTS selection using hybrid approach

\[60(32.5 - ((10x_{11} + 9x_{12} + 8x_{13}) + (8x_{21} + 9x_{22}) + (8x_{31} + 7x_{32} + 9x_{33}) + (9x_{41} + 8x_{42} + 8x_{43}))) \geq \theta,\]

\[\theta \geq 0,\]

\[x_{11} + x_{12} + x_{13} = 1, \quad (3.1.6)\]

\[x_{21} + x_{22} = 1, \quad (3.1.7)\]

\[x_{31} + x_{32} + x_{33} = 1, \quad (3.1.8)\]

\[x_{41} + x_{42} + x_{43} = 1, \quad (3.1.9)\]

\[x_{42} - x_{11} \leq 5y_1, \quad (3.1.10)\]

\[x_{42} - x_{13} \leq 5y_2, \quad (3.1.11)\]

\[y_1 + y_2 = 1, \quad (3.1.12)\]

\[y_1, y_2 \in \{0, 1\}, \quad (3.1.13)\]

\[x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{31}, x_{32}, x_{33}, x_{41}, x_{42}, x_{43} \in \{0, 1\}. \quad (3.1.14)\]

Using LINGO, we obtain the solution as: \(\theta = 0.4509, Q(x) = 0.825, C(x) = 32, x_{13} = 1, x_{21} = 1, x_{31} = 1, x_{42} = 1,\) and \(y_2 = 0.\) Thus, COTS components \(sc_{13}, sc_{21}, sc_{31},\) and \(sc_{42}\) are selected for modules \(M1, M2, M3,\) and \(M4,\) respectively. As \(y_2 = 0;\) thus, the second constraint of the either-or constraints becomes active, whereas the first constraint is relaxed. Hence, the constraint \(x_{42} \leq x_{13}\) and \(x_{42} = 1\) caused \(x_{13} = 1\) in the optimal solution. If the DM is not satisfied with the COTS selected, more COTS selection strategies can be generated by varying the values of the shape parameters \(\alpha_Q\) and \(\alpha_C.\) We have taken two different values of these parameters. To check the efficiency of the solutions obtained, we apply the two-phase approach and solve the model \(P(3.1.5).\) We find that the solutions obtained are efficient; that is, their criteria vector are nondominated. The corresponding computational results are summarized in Table 3.1.6.

<table>
<thead>
<tr>
<th>(\alpha_Q)</th>
<th>(\alpha_C)</th>
<th>(\lambda)</th>
<th>(\theta)</th>
<th>(Q(x))</th>
<th>(C(x))</th>
<th>COTS components selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>60</td>
<td>0.6109</td>
<td>0.4509</td>
<td>0.825</td>
<td>32</td>
<td>(sc_{13}) (sc_{21}) (sc_{31}) (sc_{42})</td>
</tr>
<tr>
<td>60</td>
<td>6</td>
<td>0.9797</td>
<td>3.8718</td>
<td>0.815</td>
<td>31</td>
<td>(sc_{13}) (sc_{21}) (sc_{32}) (sc_{42})</td>
</tr>
</tbody>
</table>
It may be noted that by changing values of shape parameters in the nonlinear S-shape membership functions, one can reflect the DM’s mind more accurately and suitably. Thus, we can obtain different COTS selections by solving model P(3.1.3). It is important to point out that for some choices of the aspiration levels there may be no solution to the COTS selection model P(3.1.3). In such instances, we will have to modify the aspiration levels for the various scenarios to find a satisfactory solution.

Next, we set $Q_m = 0.70$, $C_m = 35$, $\alpha_Q = 6$, $\alpha_C = 60$, and use module weight vector, $W = (0.415, 0.268, 0.177, 0.140)$. Using this data, the data given in Table 3.1.1 and by considering that fuzzy goals of the weighted quality and the cost are represented using nonlinear S-shape membership functions, we solve the model P(3.1.3) using LINGO. We obtain the solution as: $\theta = 0.8754$, $Q(x) = 0.846$, $C(x) = 34$, $x_{11} = 1$, $x_{21} = 1$, $x_{31} = 1$, $x_{42} = 1$, and $y_1 = 0$. Thus, COTS components $sc_{11}$, $sc_{21}$, $sc_{31}$, and $sc_{42}$ are selected for modules $M_1$, $M_2$, $M_3$, and $M_4$, 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. If the DM is not satisfied with the COTS selected, more COTS selection strategies can be generated by varying the values of shape parameters $\alpha_Q$ and $\alpha_C$ in model P(3.1.3). We have taken two different values of these parameters. To check the efficiency of the solutions obtained, we apply the two-phase approach and solve the model P(3.1.5). We find that the solutions obtained are efficient; that is, their criteria vector are nondominated. The corresponding computational results are summarized in Table 3.1.7.

**Table 3.1.7:** COTS selection corresponding to $Q_m = 0.70$, $C_m = 35$

<table>
<thead>
<tr>
<th>$\alpha_Q$</th>
<th>$\alpha_C$</th>
<th>$\lambda$</th>
<th>$\theta$</th>
<th>$Q(x)$</th>
<th>$C(x)$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$M_4$</th>
<th>COTS components selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>60</td>
<td>0.7058</td>
<td>0.8754</td>
<td>0.846</td>
<td>34</td>
<td>$sc_{11}$</td>
<td>$sc_{21}$</td>
<td>$sc_{31}$</td>
<td>$sc_{42}$</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>6</td>
<td>0.9997</td>
<td>8.1168</td>
<td>0.835</td>
<td>33</td>
<td>$sc_{11}$</td>
<td>$sc_{21}$</td>
<td>$sc_{32}$</td>
<td>$sc_{42}$</td>
<td></td>
</tr>
</tbody>
</table>
COTS selection using optimization model P(3.1.4) and sensitivity analysis with respect to changes in the weights of the two objectives

By varying the weights of the two objectives different solutions can be obtained. For example, let us suppose that initially the DM gives more importance to the goal of quality as compared to the goal of cost by setting $\omega_1 = 0.7$ and $\omega_2 = 0.3$. By taking $Q_m = 0.75$ and $C_m = 37$, model P(3.1.4) is formulated as:

$$\text{max } (0.7\lambda_1 + 0.3\lambda_2)$$
subject to

$$\frac{1}{(1 + \exp(-60(0.415(0.83x_{11} + 0.82x_{12} + 0.78x_{13}) + 0.268(0.85x_{21} + 0.88x_{22}) + 0.177(0.85x_{31} + 0.79x_{32} + 0.90x_{33}) + 0.14(0.90x_{41} + 0.88x_{42} + 0.81x_{43}) - 0.75)))} \geq \lambda_1,$$

$$\frac{1}{(1 + \exp(6((10x_{11} + 9x_{12} + 8x_{13}) + (8x_{21} + 9x_{22}) + (8x_{31} + 7x_{32} + 9x_{33}) + (9x_{41} + 8x_{42} + 8x_{43}) - 37)))} \geq \lambda_2,$$

$$0 \leq \lambda_1 \leq 1,$$

$$0 \leq \lambda_2 \leq 1,$$

and Constraints (3.1.6)–(3.1.14).

Using LINGO, we obtain the solution as: $\lambda_1 = 0.9989$, $\lambda_2 = 0.9975$, $Q(x) = 0.863$, $C(x) = 36$, $x_{11} = 1$, $x_{22} = 1$, $x_{33} = 1$, $x_{42} = 1$, and $y_1 = 0$. Thus, COTS components $sc_{11}$, $sc_{22}$, $sc_{33}$, 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. The efficiency of the solution is verified by solving model P(3.1.6) in the second phase. Note that the achievement levels of both the membership functions are consistent with the DM’s preferences, $\omega_1 = 0.7$ and $\omega_2 = 0.3$. In other words, $(\lambda_1 > \lambda_2)$ agrees with $(\omega_1 > \omega_2)$. The corresponding computational results are listed in Table 3.1.8. We have generated COTS selection strategies for two different values of middle aspiration levels of the two objectives.
Table 3.1.8: COTS selection corresponding to \( \omega_1 = 0.7, \omega_2 = 0.3, \alpha_Q = 60, \alpha_C = 6 \)

<table>
<thead>
<tr>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( Q_m )</th>
<th>( C_m )</th>
<th>( Q(x) )</th>
<th>( C(x) )</th>
<th>COTS components selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9989</td>
<td>0.9975</td>
<td>0.75</td>
<td>37</td>
<td>0.863</td>
<td>36</td>
<td>( sc_{11} ) ( sc_{22} ) ( sc_{33} ) ( sc_{42} )</td>
</tr>
<tr>
<td>0.9997</td>
<td>0.9975</td>
<td>0.70</td>
<td>34</td>
<td>0.835</td>
<td>33</td>
<td>( sc_{11} ) ( sc_{21} ) ( sc_{32} ) ( sc_{42} )</td>
</tr>
</tbody>
</table>

Next, we assume that the DM give less importance to the goal of quality as compared to the goal of cost by setting \( \omega_1 = 0.3 \) and \( \omega_2 = 0.7 \). Again by varying middle aspiration levels of the two objectives we solve model P(3.1.4). The corresponding computational results are listed in Table 3.1.9. The efficiency of the obtained solutions is verified by using the two-phase approach. Here, the achievement levels of both the membership functions are also consistent with the DM’s preferences, \( \omega_1 = 0.3 \) and \( \omega_2 = 0.7 \). In other words, \((\lambda_1 < \lambda_2)\) agrees with \((\omega_1 < \omega_2)\).

Table 3.1.9: COTS selection corresponding to \( \omega_1 = 0.3, \omega_2 = 0.7, \alpha_Q = 60, \alpha_C = 6 \)

<table>
<thead>
<tr>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( Q_m )</th>
<th>( C_m )</th>
<th>( Q(x) )</th>
<th>( C(x) )</th>
<th>COTS components selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9966</td>
<td>0.9999</td>
<td>0.75</td>
<td>37</td>
<td>0.845</td>
<td>35</td>
<td>( sc_{13} ) ( sc_{22} ) ( sc_{33} ) ( sc_{41} )</td>
</tr>
<tr>
<td>0.9995</td>
<td>0.9999</td>
<td>0.70</td>
<td>34</td>
<td>0.825</td>
<td>32</td>
<td>( sc_{13} ) ( sc_{21} ) ( sc_{31} ) ( sc_{42} )</td>
</tr>
</tbody>
</table>

3.1.4 Concluding remarks

In this section, we have presented fuzzy biobjective optimization models for COTS selection in the development of a modular software system. Here, we have assumed that there is no cost in developing interface programs to connect between modules, and the operation of COTS component is statistically independent. Therefore, in actual implementation, the assumptions of the model must be looked into.

It may be appreciated that when different alternatives of the same module are available with variations in the attributes of quality and cost, it involves decision making in an environment that befits more of fuzzy approximation than
3.2 COTS selection using fuzzy interactive approach

deterministic formulation. Therefore, we have drawn on fuzzy methodology for the estimation of quality and cost. Fuzzy methodology allows us to incorporate uncertainty into data and also to incorporate subjective/intuitive characteristics into the COTS selection model. Furthermore, AHP has been used to calculate the weights of the modules which also depends on the access frequencies of the modules. The main advantage of using the proposed models is that if the DM is not satisfied with any of the COTS selection obtained, more COTS selections can be generated by varying values of shape parameters of the nonlinear $S$-shape membership functions. The efficiency of the obtained solutions is verified by using the two-phase approach. The two-phase approach guarantees fuzzy-efficient solution and improves, if possible, the solution obtained by using the max-min approach and the weighted additive approach. Thus, the proposed fuzzy COTS selection models can provide satisfying COTS selection strategies according to the DM’s vague aspiration levels, varying degree of satisfaction, and varying importance of the objectives.

3.2 COTS selection using fuzzy interactive approach

Generally, alternative COTS components that belong to the same module differ from each other with respect to several attributes of quality such as reliability, execution time, and size. Therefore, these attributes should also be considered as decisive criteria in the COTS selection model. In this section, we extend our approach of Section 3.1 to develop a fuzzy multiobjective optimization model by considering these attributes as well. We use Bellman-Zadeh’s maximization principle and linear membership functions to formulate the fuzzy multiobjective optimization model. The proposed fuzzy optimization model minimizes the cost, development efforts (number of lines of code), execution time while maximizing the weighted quality and reliability of the software system. Furthermore, the model is subjected to many realistic constraints including delivery time constraints, selection of one and only one COTS component for each module, and incompatibility among COTS components. We use linear membership functions to describe the fuzzy goals of the multiobjective optimization model. Furthermore, to determine a preferred compromise solution for the fuzzy optimization
model, we use an interactive approach. To satisfy the individual preferences of the DM for various objectives, in this approach, we improve the obtained compromise solution by modifying the lower (upper) bound and the aspiration level(s) of the selected objective(s). As desired by the DM, the selected objective(s) can be improved but because of the multiobjective nature of the problem, this improvement can produce adverse effects on other objective(s). Hence, depending on the preferences of the DM for the different objectives, we can modify the compromise solution and this process is continued until the DM terminates the process, that is, the preferred compromise solution is obtained or no further improvement is made. Numerical illustrations are provided to demonstrate usefulness of the proposed model and the solution approach.

The rest of the section is organized as follows. In section 3.2.1, we formulate the multiobjective optimization model for COTS selection. In Section 3.2.2, we describe the solution methodology and present fuzzy multiobjective optimization model for selecting best-fit COTS components. Section 3.2.3 presents numerical illustrations in support of the multiobjective optimization model. This section also pertains to a discussion of the results obtained. Finally in Section 3.2.4, we furnish our concluding remarks.

### 3.2.1 Multiobjective optimization model for COTS selection

In the proposed multiobjective optimization model, we consider the following important and conflicting objectives, and constraints for COTS selection problem.

**Objectives**

- **Weighted quality**

  The objective of maximizing the weighted quality for COTS selection problem is expressed as

  \[
  \max Q(x) = \sum_{i=1}^{m} w_i \left( \sum_{j=1}^{n_i} q_{ij} x_{ij} \right).
  \]
• Cost

The objective of minimizing the total cost for COTS selection problem is expressed as

$$\min C(x) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij}.$$  

• Reliability

Since, we have assumed that the software system consists of several programs, where each program can call upon a series of modules, therefore, we consider system reliability in terms of the reliability of the modules only. It may be noted that the value of $s_i$ 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. Also, the probability that no failure occurs during the execution of the $i$-th module is given by $\exp(-s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij})$, which represents the probability of no failures in a poisson distribution. Thus, the objective of maximizing the system reliability is expressed as

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

• Size

The objective of minimizing the size (number of lines of code) due to physical limitations on the memory size of computers, is expressed as

$$\min L(x) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} l_{ij} x_{ij}.$$  

• Execution time

Here, we consider that the execution time of the system is the weighted average of the programs execution time. Also, probability of use of programs is the weightage factor. The execution time for all programs in the software system is
not same. The execution time may vary because of the variations in the number of alternative COTS components present in each module that are being called by programs and also the nature of task to be performed by COTS components. Thus, the objective of minimizing the total execution time of the software system is expressed as

$$\min T_e(x) = \sum_{p=1}^{P} \nu_p \sum_{i \in S_p} \left( \sum_{j=1}^{n_i} t_{ij} x_{ij} \right).$$

Constraints

- **Delivery time constraints**
  $$\sum_{j=1}^{n_i} d_{ij} x_{ij} \leq T, \ i = 1, 2, \ldots, m.$$  
  The above constraints ensures that a maximum threshold $T$ is imposed on the delivery time of the whole system [27].

- **Selection of only one COTS component per module**
  $$\sum_{j=1}^{n_i} x_{ij} = 1, \ i = 1, 2, \ldots, m.$$  

- **Contingent decision constraints**
  $$x_{rs} - x_{utk} \leq My_k, \ k = 1, 2, \ldots, z,$$
  $$\sum_{k=1}^{z} y_k = z - 1,$$
  $$y_k \in \{0, 1\}, \ k = 1, 2, \ldots, z.$$  

- **Selection or rejection of a COTS component**
  $$x_{ij} \in \{0, 1\}, \ i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n_i.$$  

The decision model

The proposed multiobjective optimization model for COTS selection of a modular software system depicted in Figure 3.1 which seeks to minimize the cost,
COTS selection using fuzzy interactive approach

Development efforts (number of lines of code), execution time while maximizing the reliability and quality of the whole software system subject to many realistic constraints including delivery time constraints, selection of one and only one COTS component for each module, and incompatibility among the COTS components is formulated as follows:

\[
\begin{align*}
P(3.2.1) \quad & \text{max } Q(x) = \sum_{i=1}^{m} w_i \left( \sum_{j=1}^{n_i} q_{ij} x_{ij} \right) \\
& \text{min } C(x) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \\
& \text{max } R(x) = \prod_{i=1}^{m} \exp \left( -s_i \sum_{j=1}^{n_i} \mu_{ij} x_{ij} \right) \\
& \text{min } L(x) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} l_{ij} x_{ij} \\
& \text{min } T_e(x) = \sum_{p=1}^{P} \nu_p \sum_{i \in S_p} \left( \sum_{j=1}^{n_i} t_{ij} x_{ij} \right) \\
\text{subject to} \\
& \sum_{j=1}^{n_i} d_{ij} x_{ij} \leq T, \ i = 1, 2, \ldots, m, \quad (3.2.1) \\
& \sum_{j=1}^{n_i} x_{ij} = 1, \ i = 1, 2, \ldots, m, \quad (3.2.2) \\
& x_{rs} - x_{ut} \leq M y_k, \ k = 1, 2, \ldots, z, \quad (3.2.3) \\
& \sum_{k=1}^{z} y_k = z - 1, \quad (3.2.4) \\
& y_k \in \{0, 1\}, \ k = 1, 2, \ldots, z, \quad (3.2.5) \\
& x_{ij} \in \{0, 1\}, \ i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n_i. \quad (3.2.6)
\end{align*}
\]

3.2.2 Fuzzy interactive approach

Fuzzy set theory has been implemented in mathematical programming since 1970 when Bellman and Zadeh introduced the basic concepts of fuzzy goals (G), fuzzy constraints (C), and fuzzy decisions (D). Based on these concepts,
the fuzzy decision is defined as

\[ D = G \cap C. \]

This problem is characterized by the membership function

\[ \mu_D(x) = \min (\mu_G(x), \mu_C(x)). \]

The fuzzy goals of maximization-type objective functions \( Z_h(x), h = 1, 3 \) of model P(3.2.1) regarding the weighted quality and reliability, respectively, are characterized using linear membership functions [119, 120] defined as

\[
\mu_{Z_h}(x) = \begin{cases} 
1, & \text{if } Z_h(x) \geq U_h, \\
\frac{Z_h(x) - L_h}{U_h - L_h}, & \text{if } L_h < Z_h(x) < U_h, \ h = 1, 3, \\
0, & \text{if } Z_h(x) \leq L_h,
\end{cases}
\]

where \( U_h, \ h = 1, 3, \) are the best upper bounds and \( L_h, \ h = 1, 3, \) are the worst lower bounds corresponding to the weighted quality and reliability, respectively. These are calculated as follows:

For \( h = 1, 3, \)

\( U_h = \max Z_h(x) \)

subject to

Constraints (3.2.1)–(3.2.6),

and

For \( h = 1, 3, \)

\( L_h = \min Z_h(x) \)

subject to

Constraints (3.2.1)–(3.2.6).

To exemplify, the membership function of the goal for the weighted quality is given by

\[
\mu_Q(x) = \begin{cases} 
1, & \text{if } Q(x) \geq Q_u, \\
\frac{Q(x) - Q_l}{Q_u - Q_l}, & \text{if } Q_l < Q(x) < Q_u, \\
0, & \text{if } Q(x) \leq Q_l,
\end{cases}
\]

where \( Q_l \) is the worst lower bound and \( Q_u \) is the best upper bound of the
weighted quality. A graphical representation of the membership function is given in Figure 3.2.1.

\[
\mu_Z(x) = \begin{cases} 
1, & \text{if } Z_h(x) \leq L_h, \\
\frac{U_h - Z_h(x)}{U_h - L_h}, & \text{if } L_h < Z_h(x) < U_h, \ h = 2, 4, 5, \\
0, & \text{if } Z_h(x) \geq U_h,
\end{cases}
\]

where \( U_h, \ h = 2, 4, 5, \) are the worst upper bounds and \( L_h, \ h = 2, 4, 5, \) are the best lower bounds corresponding to cost, size, and execution time, respectively.

These are calculated as follows:

For \( h = 2, 4, 5, \)
\[
U_h = \max Z_h(x)
\]
subject to
Contraints (3.2.1)-(3.2.6),

and

For \( h = 2, 4, 5, \)
\[
L_h = \min Z_h(x)
\]
subject to
Contraints (3.2.1)-(3.2.6).
To exemplify, the membership function of the goal for the total cost is given by

\[
\mu_C(x) = \begin{cases} 
1, & \text{if } C(x) \leq C_l, \\
\frac{C_u - C(x)}{C_u - C_l}, & \text{if } C_l < C(x) < C_u, \\
0, & \text{if } C(x) \geq C_u,
\end{cases}
\]

where \(C_u\) is the worst upper bound and \(C_l\) is the best lower bound of the cost. A graphical representation of the membership function is given in Figure 3.2.2.

The linear membership function of the goal for the system reliability is similar to that of the goal for the weighted quality. The linear membership functions of the goals corresponding to the size (number of lines of code) and the system execution time are similar to that of the goal for the total system cost.

Following Bellman-Zadeh’s maximization principle and using the above defined membership functions, the fuzzy multiobjective optimization model for COTS selection is formulated as follows:

\[
P(3.2.2) \quad \max \lambda \\
\text{subject to} \\
\lambda \leq \mu_Q(x), \\
\lambda \leq \mu_C(x), \\
\lambda \leq \mu_R(x), \\
\lambda \leq \mu_L(x),
\]
\[ \lambda \leq \mu_{T_e}(x), \]
\[ 0 \leq \lambda \leq 1, \]
and Constraints (3.2.1)–(3.2.6).

To deal with individual preferences among the various objectives, a weighted additive model of fuzzy optimization model can be considered as discussed in Section 3.1.1. Here, we incorporate individual preferences of the DM using the following interactive approach.

**Solution steps of the interactive approach**

The solution methodology of the interactive approach for model P(3.2.1) consists of the following steps:

Step 1: Construct model P(3.2.1).

Step 2: Solve model P(3.2.1) as a single objective optimization problem by considering first objective function. This process is repeated for all remaining objective functions. If all the solutions (that is, \( X^1 = X^2 = \ldots = X^5 = x_{ij}, i = 1,2,\ldots,m, j = 1,2,\ldots,n_i \) are same, select one of them as an optimal compromise solution and stop. Otherwise, go to Step 3.

Step 3: Evaluate the \( h \)-th (\( h = 1,2,\ldots,5 \)) objective function at all solutions obtained and determine the best (worst) lower bound (\( L_h \)) and best (worst) upper bound (\( U_h \)) as the case may be.

Step 4: Define the membership function of each objective function.

Step 5: Develop model P(3.2.2) and solve it. Present the solution to the DM. If the DM accepts it then stop. Otherwise, evaluate each objective function at the obtained solution. Compare the upper (lower) bound of each objective function with the new value of the objective function. If the new value is lower (higher) than the upper (lower) bound, consider it as a new upper (lower) bound. Otherwise, use the old value as is. If there are no changes in current bounds of all the objective functions then stop otherwise go to Step 4.

The solution process terminates when the DM accepts the obtained solution and consider it as the preferred compromise solution which is in fact a compromise feasible solution that meets the DM’s preferences.
3.2.3 An illustrative example

System prototype

Here, we refer to the same system prototype which was considered in Section 3.1.3. The data for different parameters of alternative COTS components and the contingent decision constraints are listed in Table 3.2.1.

Table 3.2.1: Input data of the system prototype for model P(3.2.1)

<table>
<thead>
<tr>
<th>Quality :</th>
<th>$q_{11} = 0.72$</th>
<th>$q_{21} = 0.72$</th>
<th>$q_{31} = 0.81$</th>
<th>$q_{41} = 0.86$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{12} = 0.83$</td>
<td>$q_{22} = 0.88$</td>
<td>$q_{32} = 0.72$</td>
<td>$q_{42} = 0.80$</td>
<td></td>
</tr>
<tr>
<td>$q_{13} = 0.61$</td>
<td>$q_{33} = 0.90$</td>
<td>$q_{43} = 0.75$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost :</td>
<td>$c_{11} = 10$</td>
<td>$c_{21} = 9$</td>
<td>$c_{31} = 7$</td>
<td>$c_{41} = 7$</td>
</tr>
<tr>
<td>$c_{12} = 9$</td>
<td>$c_{22} = 8$</td>
<td>$c_{32} = 8$</td>
<td>$c_{42} = 9$</td>
<td></td>
</tr>
<tr>
<td>$c_{13} = 8$</td>
<td>$c_{33} = 6$</td>
<td>$c_{43} = 8$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probability of failure</td>
<td>$\mu_{11} = 0.0002$</td>
<td>$\mu_{22} = 0.0002$</td>
<td>$\mu_{31} = 0.0002$</td>
<td>$\mu_{41} = 0.0008$</td>
</tr>
<tr>
<td>of failure on demand :</td>
<td>$\mu_{12} = 0.0003$</td>
<td>$\mu_{22} = 0.0004$</td>
<td>$\mu_{32} = 0.0004$</td>
<td>$\mu_{42} = 0.0003$</td>
</tr>
<tr>
<td>$\mu_{13} = 0.0005$</td>
<td>$\mu_{33} = 0.0006$</td>
<td>$\mu_{43} = 0.0006$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average no. of invocations :</td>
<td>$s_1 = 110$</td>
<td>$s_2 = 55$</td>
<td>$s_3 = 80$</td>
<td>$s_4 = 120$</td>
</tr>
<tr>
<td>No. of lines of code :</td>
<td>$l_{11} = 10000$</td>
<td>$l_{21} = 12000$</td>
<td>$l_{31} = 8000$</td>
<td>$l_{41} = 15000$</td>
</tr>
<tr>
<td>$l_{12} = 15000$</td>
<td>$l_{22} = 13000$</td>
<td>$l_{32} = 6000$</td>
<td>$l_{42} = 13000$</td>
<td></td>
</tr>
<tr>
<td>$l_{13} = 17000$</td>
<td>$l_{33} = 5000$</td>
<td>$l_{43} = 11000$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execution Time:</td>
<td>$t_{11} = 1$</td>
<td>$t_{21} = 2.2$</td>
<td>$t_{31} = 3.0$</td>
<td>$t_{41} = 6$</td>
</tr>
<tr>
<td>$t_{12} = 1.6$</td>
<td>$t_{22} = 1.9$</td>
<td>$t_{32} = 3.5$</td>
<td>$t_{42} = 6.6$</td>
<td></td>
</tr>
<tr>
<td>$t_{13} = 2.5$</td>
<td>$t_{33} = 4.6$</td>
<td>$t_{43} = 8$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivery Time:</td>
<td>$d_{11} = 4$</td>
<td>$d_{21} = 7$</td>
<td>$d_{31} = 3$</td>
<td>$d_{41} = 6$</td>
</tr>
<tr>
<td>$d_{12} = 4$</td>
<td>$d_{22} = 4$</td>
<td>$d_{32} = 7$</td>
<td>$d_{42} = 8$</td>
<td></td>
</tr>
<tr>
<td>$d_{13} = 7$</td>
<td>$d_{33} = 4$</td>
<td>$d_{43} = 4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contingent decision constraints:</td>
<td>$x_{42} \leq x_{11}$ or $x_{42} \leq x_{13}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calculation of module weights using AHP

The module weights are calculated using AHP on the same lines as given in Section 3.1.3. The module weight vector is $W = (0.415, 0.268, 0.177, 0.140)$. 
COTS component selection

Step 1: Using the input data and contingent decision constraints from Table 3.2.1, model P(3.2.1) is formulated as follows:

\[
\begin{align*}
\max Q(x) &= (0.415(0.72x_{11} + 0.83x_{12} + 0.61x_{13}) + 0.268(0.72x_{21} + 0.88x_{22}) + 0.177(0.81x_{31} + 0.72x_{32} + 0.90x_{33}) + 0.140(0.86x_{41} + 0.80x_{42} + 0.75x_{43})) \\
\min C(x) &= ((10x_{11} + 9x_{12} + 8x_{13}) + (9x_{21} + 8x_{22}) + (7x_{31} + 8x_{32} + 6x_{33}) + (7x_{41} + 9x_{42} + 8x_{43})) \\
\max R(x) &= (\exp(-110(0.0002x_{11} + 0.0003x_{12} + 0.0005x_{13}))) \ast \exp(-55(0.0002x_{21} + 0.0004x_{22})) \ast \exp(-(80(0.0002x_{31} + 0.0004x_{32} + 0.0006x_{33})) \ast \exp(-120(0.0008x_{41} + 0.0003x_{42} + 0.0006x_{43}))) \\
\min L(x) &= ((10000x_{11} + 15000x_{12} + 17000x_{13}) + (12000x_{21} + 13000x_{22}) + (8000x_{31} + 6000x_{32} + 5000x_{33}) + (15000x_{41} + 13000x_{42} + 11000x_{43})) \\
\min T_e(x) &= (0.411((x_{11} + 1.6x_{12} + 2.5x_{13}) + (2.2x_{21} + 1.9x_{22}) + 0.261((2.2x_{21} + 1.9x_{22}) + (3x_{31} + 3.5x_{32} + 4.6x_{33}) + 0.328((x_{11} + 1.6x_{12} + 2.5x_{13}) + (3x_{31} + 3.5x_{32} + 4.6x_{33}) + (6x_{41} + 6.6x_{42} + 8x_{43})))
\end{align*}
\]

subject to

\[
\begin{align*}
4x_{11} + 4x_{12} + 7x_{13} &\leq 9, \quad (3.2.7) \\
7x_{21} + 4x_{22} &\leq 9, \quad (3.2.8) \\
3x_{31} + 7x_{32} + 4x_{33} &\leq 9, \quad (3.2.9) \\
6x_{41} + 8x_{42} + 4x_{43} &\leq 9, \quad (3.2.10) \\
x_{11} + x_{12} + x_{13} &= 1, \quad (3.2.11) \\
x_{21} + x_{22} &= 1, \quad (3.2.12) \\
x_{31} + x_{32} + x_{33} &= 1, \quad (3.2.13) \\
x_{41} + x_{42} + x_{43} &= 1, \quad (3.2.14) \\
x_{42} - x_{11} &\leq 5y_1, \quad (3.2.15) \\
x_{42} - x_{13} &\leq 5y_2, \quad (3.2.16)
\end{align*}
\]
\[ y_1 + y_2 = 1, \quad (3.2.17) \]
\[ y_1, \ y_2 \in \{0, 1\}, \quad (3.2.18) \]
\[ x_{11}, \ x_{12}, \ x_{13}, \ x_{21}, \ x_{22}, \ x_{31}, \ x_{32}, \ x_{33}, \ x_{41}, \ x_{42}, \ x_{43} \in \{0, 1\}. \quad (3.2.19) \]

Step 2: The solution of each single objective problem is obtained as

\[ X^1 = (0, 1, 0, 0, 1, 0, 0, 1, 0, 0), \quad X^2 = (0, 0, 1, 0, 1, 0, 0, 1, 0, 0), \]
\[ X^3 = (1, 0, 0, 1, 0, 0, 0, 1, 0, 0), \quad X^4 = (1, 0, 0, 1, 0, 0, 0, 1, 0, 0), \]
\[ X^5 = (1, 0, 0, 1, 0, 0, 0, 1, 0, 0). \]

Step 3: The objective function values are

<table>
<thead>
<tr>
<th>Objective Function</th>
<th>( X^1 )</th>
<th>( X^2 )</th>
<th>( X^3 )</th>
<th>( X^4 )</th>
<th>( X^5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q(x) ) (( Z_1 ))</td>
<td>0.85999</td>
<td>0.76869</td>
<td>0.74713</td>
<td>0.75606</td>
<td>0.79841</td>
</tr>
<tr>
<td>( C(x) ) (( Z_2 ))</td>
<td>30</td>
<td>29</td>
<td>35</td>
<td>33</td>
<td>32</td>
</tr>
<tr>
<td>( R(x) ) (( Z_3 ))</td>
<td>0.81955</td>
<td>0.8017</td>
<td>0.91851</td>
<td>0.85813</td>
<td>0.85556</td>
</tr>
<tr>
<td>( L(x) ) (( Z_4 ))</td>
<td>48000</td>
<td>50000</td>
<td>43000</td>
<td>38000</td>
<td>46000</td>
</tr>
<tr>
<td>( T_e(x) ) (( Z_5 ))</td>
<td>7.1366</td>
<td>7.8017</td>
<td>6.1492</td>
<td>7.5508</td>
<td>5.7508</td>
</tr>
</tbody>
</table>

Thus, the upper and lower bounds of each objective function are

\[ 0.74713 \leq Q(x) \leq 0.85999, \quad 29 \leq C(x) \leq 35, \quad 0.8017 \leq R(x) \leq 0.91851, \]
\[ 38000 \leq L(x) \leq 50000, \quad 5.7508 \leq T_e(x) \leq 7.8017. \]

Step 4: The membership functions of the five objectives using the lower and upper bounds from Step 3 and as per the details in Section 3.2.2 are constructed as follows:

- The membership function of the goal for the weighted quality is given by

\[ \mu_Q(x) = \begin{cases} 
1, & \text{if } Q(x) \geq 0.85999, \\
\frac{Q(x) - 0.74713}{0.11286}, & \text{if } 0.74713 < Q(x) < 0.85999, \\
0, & \text{if } Q(x) \leq 0.74713.
\end{cases} \]

- The membership function of the goal for the total cost is given by
\[ \mu_C(x) = \begin{cases} 
1, & \text{if } C(x) \leq 29, \\
\frac{35 - C(x)}{6}, & \text{if } 29 < C(x) < 35, \\
0, & \text{if } C(x) \geq 35.
\end{cases} \]

- The membership function of the goal for the system reliability is given by

\[ \mu_R(x) = \begin{cases} 
1, & \text{if } R(x) \geq 0.91851, \\
\frac{R(x) - 0.8017}{0.11681}, & \text{if } 0.8017 < R(x) < 0.91851, \\
0, & \text{if } R(x) \leq 0.8017.
\end{cases} \]

- The membership function of the goal for the size (number of lines of code) is given by

\[ \mu_L(x) = \begin{cases} 
1, & \text{if } L(x) \leq 38000, \\
\frac{50000 - L(x)}{12000}, & \text{if } 38000 < L(x) < 50000, \\
0, & \text{if } L(x) \geq 50000.
\end{cases} \]

- The membership function of the goal for the system execution time is given by

\[ \mu_C(x) = \begin{cases} 
1, & \text{if } T_e(x) \leq 5.7508, \\
\frac{7.8017 - T_e(x)}{2.0509}, & \text{if } 5.7508 < T_e(x) < 7.8017, \\
0, & \text{if } T_e(x) \geq 7.8017.
\end{cases} \]

Step 5: The model \( P(3.2.2) \) is developed as follows:

\[
\max \lambda \\
\text{subject to}
\]

\[
0.415(0.72x_{11} + 0.83x_{12} + 0.61x_{13}) + 0.268(0.72x_{21} + 0.88x_{22}) + \\
0.177(0.81x_{31} + 0.72x_{32} + 0.90x_{33}) + 0.140(0.86x_{41} + 0.80x_{42} + \\
0.75x_{43}) - 0.11286\lambda \geq 0.74713, \\
(10x_{11} + 9x_{12} + 8x_{13}) + (9x_{21} + 8x_{22}) + (7x_{31} + 8x_{32} + 6x_{33}) + \\
(7x_{41} + 9x_{42} + 8x_{43})) + 4\lambda \leq 33,
\]
\[
\exp(-110(0.0002x_{11} + 0.0003x_{12} + 0.0005x_{13})) \cdot \exp(-55(0.0002x_{21} + 0.0004x_{22})) \cdot \exp(-80(0.0002x_{31} + 0.0004x_{32} + 0.0006x_{33})) \cdot \exp(-120(0.0008x_{41} + 0.0003x_{42} + 0.0006x_{43}))
\]

\[
-0.11681\lambda \geq 0.8017,
\]

\[
(10000x_{11} + 15000x_{12} + 17000x_{13}) + (12000x_{21} + 13000x_{22}) + (8000x_{31} + 6000x_{32} + 5000x_{33}) + (15000x_{41} + 13000x_{42} + 11000x_{43}) + 12000\lambda \leq 50000,
\]

\[
0.411((x_{11} + 1.6x_{12} + 2.5x_{13}) + (2.2x_{21} + 1.9x_{22})) + 0.261((2.2x_{21} + 1.9x_{22}) + (3x_{31} + 3.5x_{32} + 4.6x_{33})) + 0.328((x_{11} + 1.6x_{12} + 2.5x_{13}) + (3x_{31} + 3.5x_{32} + 4.6x_{33}) + (6x_{41} + 6.6x_{42} + 8x_{43})) + 2.0509\lambda \leq 7.8017,
\]

\[
0 \leq \lambda \leq 1,
\]

and Constraints (3.2.7)–(3.2.19).

Using LINGO, we obtain the solution as: \(\lambda = 0.3333, \, Q(x) = 0.78578, \, C(x) = 33, \, R(x) = 0.8763, \, L(x) = 46000, \, T_e(x) = 7.0518, \, x_{11} = 1, \, x_{21} = 1, \, x_{31} = 1, \, x_{43} = 1\). Thus, the COTS components \(s_{c11}, \, s_{c21}, \, s_{c31}, \) and \(s_{c43}\) which contributed towards this solution are selected for modules \(M1, \, M2, \, M3, \) and \(M4, \) respectively.

**Sensitivity Analysis**

Suppose the DM is not satisfied with the preferred compromise solution obtained and wants to improve it further. As desired by the DM, an individual objective can be improved but due to the multiobjective nature of the problem this improvement can produce adverse effects on other objective(s). Hence, depending upon the preferences of the DM for the various objectives, we can modify the preferred compromise solution. In this process, the lower (upper) bound and aspiration level of the selected objective function are modified. The model \(P(3.2.2)\) is resolved with the new parameters and this process is continued until the DM terminates the process. To exemplify, let us suppose that the DM wishes to improve cost objective. The upper bound for the cost objective
function is changed corresponding to the obtained solution and consequently, the new upper and lower bounds for cost are $29 \leq C(x) \leq 33$. Thus, the membership function of the goal for the total cost becomes

$$
\mu_C(x) = \begin{cases} 
1, & \text{if } C(x) \leq 29, \\
\frac{33-C(x)}{4}, & \text{if } 29 < C(x) < 33, \\
0, & \text{if } C(x) \geq 33.
\end{cases}
$$

The upper and lower bounds, and the membership functions of the other objectives remain unchanged. We solve the model $P(3.2.2)$ using LINGO and obtain the solution as: $\lambda = 0.25$, $Q(x) = 0.79841$, $C(x) = 32$, $R(x) = 0.8556$, $L(x) = 46000$, $T_\epsilon(x) = 5.7508$, $x_{11} = 1$, $x_{22} = 1$, $x_{31} = 1$, $x_{41} = 1$. Thus, the COTS components $sc_{11}$, $sc_{22}$, $sc_{31}$, and $sc_{41}$ which contributed towards this solution are selected for modules $M_1$, $M_2$, $M_3$, and $M_4$, respectively. We can further improve the cost objective in the similar way. But due to the multiobjective nature of the problem improvement in one or more objective value(s) produces adverse effect on other objective(s). Table 3.2.2 lists some sample preferred compromise solutions obtained by varying bounds of various objectives.
<table>
<thead>
<tr>
<th>Case</th>
<th>Objective Function</th>
<th>Bounds</th>
<th>COTS component selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cost</td>
<td>29 ≤ C(x) ≤ 330.25</td>
<td>0.79841 32 0.8556 46000 5.7508</td>
</tr>
<tr>
<td>2</td>
<td>Quality</td>
<td>78578 ≤ Q(x) ≤ 85999</td>
<td>0.3203472 0.81273 33 0.853 45000 7.1447</td>
</tr>
<tr>
<td>3</td>
<td>Reliability</td>
<td>8763 ≤ R(x) ≤ 91851</td>
<td>0.16667 0.79001 0.9085 44000 5.9476</td>
</tr>
<tr>
<td>4</td>
<td>Size</td>
<td>38000 ≤ L(x) ≤ 46000</td>
<td>0.33333 0.80594 33 0.8799 41000 6.89</td>
</tr>
<tr>
<td>5</td>
<td>Execution Time</td>
<td>7508 ≤ T(e)(x) ≤ 70518</td>
<td>0.2304144 0.81434 31 0.8286 43000 6.6932</td>
</tr>
<tr>
<td>6</td>
<td>Quality &amp; Cost</td>
<td>78578 ≤ Q(x) ≤ 85999</td>
<td>0.2304144 0.81434 31 0.8286 43000 6.6932</td>
</tr>
<tr>
<td>7</td>
<td>Cost &amp; Reliability</td>
<td>29 ≤ C(x) ≤ 330.1528113</td>
<td>0.85999 30 0.8195 48000 7.1366</td>
</tr>
<tr>
<td>8</td>
<td>Size &amp; Execution Time</td>
<td>38000 ≤ L(x) ≤ 39000</td>
<td>0.08333 0.82813 32 0.8328 39000 7.3492</td>
</tr>
<tr>
<td>9</td>
<td>Reliability &amp; Cost</td>
<td>8763 ≤ R(x) ≤ 85999</td>
<td>0.08418335 0.80594 33 0.8799 41000 6.89</td>
</tr>
<tr>
<td>10</td>
<td>Cost &amp; Reliability</td>
<td>29 ≤ C(x) ≤ 330.85999</td>
<td>0.09985 32 0.8668 47000 6.8502</td>
</tr>
</tbody>
</table>

**Table 3.2.2:** Compromise solutions with respect to improvements desired in various objectives.
3.2.4 Concluding remarks

In real COTS selection decisions, the DM must simultaneously handle conflicting objectives. This section presented an interactive fuzzy approach for solving a multiobjective COTS selection model that provides a strategy for COTS selection from the pool of alternative COTS components, differing in their desirable properties such as quality, cost, and reliability. Each objective of the COTS selection model is basically the desirable selection criterion. The solution approach controls the search direction via updating both upper and lower bounds and aspiration level of each objective function. The obtained solution is a preferred compromise solution which is more realistic from the DM’s point-of-view keeping in mind the multiple conflicting nature of objectives considered. Using sensitivity analysis, individual preferences of the DM for various objective functions were considered in the solution process. The numerical illustrations clearly shows that the combination of quality model, AHP, fuzzy mathematical programming and interactive approach as solution methodology is quite capable of providing a systematic framework for COTS selection where the DM has the freedom of modifying the bounds and aspiration levels of various objectives interactively in order to obtain a preferred compromise solution.