A Fuzzy Optimization Framework for COTS Products Selection of Modular Software Systems

Pankaj Gupta, Hoang Pham, Mukesh Kumar Mehlawat, and Shilpi Verma

Abstract

In this paper, we discuss a decision-making situation under uncertainty related to software creation through Commercial-Off-The-Shelf (COTS) based modules. We propose a bi-objective fuzzy optimization model of the COTS selection problem. The proposed optimization model simultaneously maximize the weighted quality and minimize the total cost of a modular software system subject to many limitations including maximum threshold on delivery time of the software and incompatibility among COTS products. The coefficients of both the objective functions and the delivery time constraints are characterized by fuzzy restrictions with triangular possibility distributions. Analytical hierarchy process technique is used to assign weights to the various modules according to their access frequencies and also using preferences of the software developer regarding technical specifications of the software system. Using possibility theory, an auxiliary linear multiple objective 0-1 programming model parameterized by possibility (feasibility) level \( \beta \) is obtained corresponding to the fuzzy COTS selection model. The resultant model is converted into an equivalent crisp 0-1 linear programming model using max-min approach employing linear membership functions for all the fuzzy sets and we solved the equivalent model using an interactive approach that provides fuzzy solutions corresponding to different possibility levels \( \beta \). A real-world case study of ERP software system is included for numerical illustrations to deal with many situations of the imprecise data in the fuzzy COTS selection model.

Keywords: Possibility theory, COTS selection, Multiple criteria optimization, Fuzzy optimization, Modular software system.

1. Introduction

Today's need for rapid software development has generated a great interest in employing Commercial-Off-The-Shelf (COTS) software products as a way of managing cost, developing time and effort [1]. In order to realize the benefits of COTS products, it is important that the “right” products are selected for various software systems. The selection of best COTS products has thus become the key to the quality and cost of the modular software systems based on COTS. Selecting an appropriate COTS product is often a non-trivial task and is considered as a multiple criteria decision making process. With an abundance of COTS products to choose from, the COTS selection problem becomes how to systematically evaluate, rank and select a COTS product that best meets the software project requirements. Several COTS selection methods [8, 10, 22, 23, 26, 30, 37] have been proposed in literature. However, it may be noted that there is no single method which is accepted as a standard COTS selection method. A detailed list of the COTS selection methods has been provided by Mohamed et al. [27].

Alternatively, optimization techniques have been used in the COTS selection process to achieve the different attributes of quality along with the objective of minimizing the cost or keeping cost to a specified budgetary level. Berman and Ashrafi [4] discussed optimization models for reliability of modular software systems. Chi et al. [7] presented a software reliability optimization model. Cortellessa et al. [9] developed an optimization model that supports “build-or-buy” decisions in selecting software components based on cost-reliability trade-off. Jung and Choi [20] introduced two optimization models for the COTS selection in the development of modular software systems considering cost-reliability trade-off. Kwong et al. [24] presented an optimization model for determining the optimal selection of software components for component-based software system development. Neubauer and Stummer [28] presented a two-phase decision support approach based on multiobjective optimization for the COTS selection.
Tang et al. [35] presented an optimization model for software component selection under multiple applications development. Zachariah and Rattihalli [39] used goal-programming approach in a multi-criteria optimization framework for the COTS selection of modular software systems. Zahedi and Ashrafi [42] discussed software reliability allocation using optimization approach based on structure, utility, price and cost. All the above mentioned optimization models are based on the assumption that in the software development process, the decision maker has complete information. It may be noted that since software development is not an exact science, there are often plenty of indefinite and uncertain factors in the estimation of model parameters. Hence, the various model parameters are often imprecise or the process of estimation of these input parameters is subjected to uncertainty in a non-stochastic sense. Thus, it is desirable to incorporate uncertainty in the mathematical model itself. Corresponding to the various possible scenarios of uncertainty in the model parameters, we must explore different possible outcomes and then select the best outcome in a given decision making situation. The optimization models in the software development may benefit greatly from the fuzzy set theory [40] in terms of integrating quantitative and qualitative information, subjective preferences of the decision maker and knowledge of the software experts. The crisp optimization models of the COTS selection may be extended to imprecise/uncertain/fuzzy COTS selection model in order to provide satisfactory solutions to the decision maker in different scenarios of input data uncertainty. Gupta et al. [11] formulated bi-objective fuzzy optimization model for the COTS selection using nonlinear S-shaped functions describing vague aspiration levels of the decision maker in respect of the weighted quality and cost. Gupta et al. [12] proposed a membership function approach for cost-reliability trade-off of COTS selection in fuzzy environment. Recently, Gupta et al. [13] developed an interactive fuzzy approach using linear membership functions for the COTS selection problem considering multiple objectives. Shen et al. [33] presented a fuzzy optimization model subject to a parameterized budgetary level constraint for selecting the best COTS product considering weighted quality as an objective function. Jha et al. [19] presented a fuzzy approach for optimal selection of COTS components for modular software system under consensus recovery block scheme incorporating execution time.

In this paper, we propose a bi-objective fuzzy optimization framework for the COTS selection under the following assumptions:

1. Modular programming is used for software development.
2. Module versions are developed independently.

It may be noted that because of assumption (2) the proposed COTS selection model is applicable only to those software systems that are developed using COTS based modules which are independently generated and tested. The proposed research can be considered as a generalization and extension of the optimization models proposed by Jung and Choi [20] in terms of providing a systematic framework for COTS selection that facilitates software development process of a modular software. The proposed bi-objective COTS selection model simultaneously maximize the weighted quality and minimize the total cost of a modular software system subject to many limitations including maximum threshold on the delivery time of the software and incompatibility among COTS products. We use imprecise coefficients characterized by triangular possibility distributions in both the objective functions and the delivery time constraints. The possibility theory proposed by Zadeh [41] is used to develop an interactive approach yielding fuzzy solutions corresponding to different possibility levels $\beta$.

The key differences between our study and the relevant studies from literature are as follows: (i) Unlike the use of crisp model parameters for COTS selection in extant literature [11, 13, 20, 33], we use imprecise model parameters. In real-world problems, various model parameters, namely, cost, quality, and delivery time cannot be determined exactly because these may be affected by various indefinite and uncertain factors which cannot be measured precisely. Also, most of the parameters embedded in COTS selection problem are frequently fuzzy in nature because of incompleteness and/or unavailability of required data over the planning horizon and can be just obtained subjectively. Thus, the use of imprecise model parameters provide the decision maker great flexibility in dealing with different possible scenarios of input data uncertainty; (ii) the proposed approach can easily solve the imprecise objective functions and obtain the whole possibilistic distribution of the satisfying objective value. Furthermore, it can solve rather general imprecise optimization problems through the involvement with the decision maker thereby capturing his preferences in the solution process. Three critical values (the most possible, the pessimistic, and the optimistic values) are only necessary for each imprecise objective function; (iii) by making little effort it is really possible to obtain a lot of information about the solution obtained; and (iv) the proposed approach solves COTS selection problem with imprecise parameters more efficiently since it preserves original linear form for the fuzzy goals. The proposed method can be easily implemented using commercially available
software such as LINDO and other related mathematical programming software.

The paper is organized as follows. Section 2 presents the bi-objective fuzzy optimization model of the COTS software selection problem. In Section 3, we develop the interactive approach and present the solution procedure. Section 4 presents numerical illustrations of a real-world case study inspired from component based software development to test the effectiveness of the proposed approach. Section 5, presents comparison and benefits of the proposed interactive possibilistic programming approach and other existing fuzzy programming approaches. Finally, we furnish our concluding remarks in Section 6.

2. COTS Selection Formulation

We consider the design of a software system that is developed using COTS based modules and is required to perform one or more programs as specified by the user. It is assumed that 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 products are available for each module. The modules are developed independently using the best COTS products available for each module. The diagrammatic depiction of such a software system is given in Figure 1 that consists of m programs, k modules and several COTS alternatives, for example, $s_{c_{11}}$, $s_{c_{12}}$, ..., $s_{c_{1n}}$ are available to implement module 1 and so on.

The hierarchy shown in Figure 1 links the user’s view (program level) with the software developer’s view (module level). Note 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 products may be justified by the frequent use of the module.

Using the above shown hierarchy, 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 [2] on software selection. In this paper, we use the Analytic Hierarchy Process (AHP) technique [31] to assign weights to the modules. One of the most important advantages of the AHP is that it use pair-wise comparisons based on the decision maker’s view of the system. Besides, the AHP calculates the inconsistency index which is the ratio of the decision maker’s inconsistency in pair-wise comparisons. The AHP technique is used in many applications [11, 13, 18, 20, 33, 42] to evaluate the quality of modular software systems.

![Figure 1. The hierarchy of a modular software system.](image)

There are many factors and issues that must be taken into consideration during selection of the COTS products. Any generic stage-based methodology for selection of COTS products broadly consists of the following stages [17]:

1) Determining the need for purchasing the product and preliminary investigation of the availability of COTS products that might be suitable candidates, including high level investigation of software features and capabilities provided by vendor.

2) Short listing of alternative COTS products.

3) Eliminating those alternative products that do not have required features or do not work with the existing hardware, operating system and database management software or network.

4) Using an evaluation technique to evaluate remaining products and obtain a score or overall ranking of them.

5) Purchasing and implementing most appropriate COTS products.

In this paper, our main focus is on stages 4) and 5) of COTS selection process. We use quality criteria for evaluation of the alternative COTS products using ISO/IEC 9126-I [16] quality standard. Further, cost/quality trade-off is used for selecting best-fit COTS products for modular software systems using fuzzy optimization model.

### A. COTS Evaluation Using Quality Criteria

COTS products are available with the built-in functionality and hence the end user has no or little

---

**References**

[11, 13, 18, 20, 33, 42] to evaluate the quality of modular software systems.
control on their functionality. Thus, the non-functional requirements which define the general quality of the software system have received more attention from researchers in the recent past. We use quality criteria derived from the quality standard ISO/IEC 9126-I [16] to access quality of the COTS products. Currently, ISO/IEC 9126-I standard is widely used by researchers and practitioners to evaluate software quality. ISO/IEC 9126-I quality model is defined by means of general software criteria which are further represented by sub-criteria which in turn are decomposed into attributes, yielding to a multilevel hierarchy. At the bottom of hierarchy appear the measurable software alternatives, whose scores are computed by using some metric, for example, pairwise comparisons which are scale independent.

![Figure 2. The hierarchy of quality criteria for COTS evaluation.](image)

Standards for the software quality models defines software quality criteria as composed of six external criteria of interest, namely, functionality, reliability, usability, maintainability, portability and efficiency. Functionality of the COTS product is nothing but the ability of the COTS product to perform according to the specific needs of the organization. Reliability is ability of the product to run consistently without crashing under specific conditions. 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 products as well as easiness to learn and operate it under certain specific conditions. Maintainability is the ability of the product to be modified. Modifications can include corrections, improvements or adaptations of the software to adjust to changes in the environment, in terms of functional requirements and specifications. Portability is the ability of the product to be transferred from one environment to another. Efficiency is ability of the COTS product to provide appropriate performance, relative to the amount of resources used under certain conditions. Note that, each of these criteria may be decomposed into sub-criteria. The decision hierarchy used for quality level estimation of COTS products is depicted in Figure 2. The quality level determined using pairwise comparison metric provides values ranging from 0 to 1 where 1 refers to a very high quality level.

B. COTS Selection Using Fuzzy Decision Theory

Today’s need for rapid software development has generated a great interest in employing COTS software products as a way of managing cost, developing time and effort. To realize the benefits of COTS products, it is important that the “right” products are selected for a software system. The selection of best-fit COTS products has now become the key to the quality and cost of the modular software systems based on COTS. Decision making in COTS selection requires considering conflicting objectives as well as different system constraints.

We study a bi-objective fuzzy COTS selection problem that has two objectives, namely, maximization of the quality of the software system and minimization of the purchasing cost. The task of COTS selection is said to be accomplished optimally if the selected COTS products meet the realistic constraints, namely, delivery time, contingent decision and one and only one COTS product for each module, providing maximum quality for the software system requiring minimum purchasing cost.

We formulate the bi-objective fuzzy optimization model of COTS selection by considering the following challenges associated with the problem.

1) Need for hierarchical decision-making
2) Many similar COTS products
3) Two COTS selection objectives
4) Selection of one COTS product for each module
5) Delivery time restriction on the software
6) Incompatibility issues among COTS products

Operationally, formulating a COTS selection problem requires an estimate of quality, cost, and delivery time for the various alternative COTS in the modules. In real-world problems, these estimates cannot be determined exactly because cost, quality, and delivery time are affected by various indefinite and uncertain factors which cannot be measured precisely. Also, the decision makers assessment about these estimates may be based on incomplete knowledge about the COTS product itself.
and other aspects (e.g. vendors credentials, etc.), which may affect the decision of selecting COTS products. Therefore, a decision based on a crisp model is not the best decision. Under such circumstances, the issue of COTS selection becomes one of a choice from a “fuzzy” set of subjective preferences. The fuzzy optimization models can be classified on the basis of two concepts: (i) the limits of fuzziness that the decision maker sets—decision maker’s aspirations— with respect to the objective and/or constraints, and (ii) the ambiguity in the coefficients of the objective functions and/or constraints (i.e., uncertainty in data). The uncertainty in data can be classified into two groups: (a) randomness, that comes from the random nature of parameters and usually stochastic programming approaches are used to model this kind of uncertainty; (b) epistemic uncertainty that deals with ill-known parameters and usually possibilistic programming approaches are used to handle the epistemic uncertainty [14]. The research work proposed in this paper deals with imprecision in the parameters of the objective functions and constraints instead of dealing with vagueness in the aspiration levels of the decision maker studied in our earlier works on fuzzy COTS selection [11, 12, 13].

According to the above-mentioned description, and because we are confronting imprecise parameters in the COTS selection problem, possibilistic programming approach is used to handle the uncertain parameters in the proposed model. Each ill-known parameter is assumed to follow a possibility distribution in our COTS selection problem: the weighted quality (1), the imprecise quality level of the j-th COTS alternative in the i-th module, the imprecise cost of the j-th COTS alternative in the i-th module, the imprecise delivery time of the j-th COTS alternative in the i-th module, the binary variable indicating whether the j-th COTS alternative of the i-th module is chosen or not, i.e.:

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

B.2. Objective Functions

We consider two important and conflicting objectives in our COTS selection problem: the weighted quality (z1) and the total cost of purchasing (z2). Several other studies [11, 13, 20, 33] also considered similar objectives but not with imprecise coefficients. The use of imprecise coefficients provides the decision maker great flexibility in dealing with different possible scenarios of input data uncertainty.

Weighted Quality

Similar to the work of Jung and Choi [20], 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 Section 2.1. The individual measures of quality may conflict each other and comparisons may have to be reached and hence the overall quality measure is an optimum balance of factors rather than an ideal way of representing quality. Since the calculations for quality levels using quality standard model involves subjective preferences of the decision maker and qualitative information, we consider them as imprecise numbers. The objective of maximizing the weighted quality of the COTS selection problem is expressed as:

$$\text{max } z_1 = \sum_{i=1}^{n} w_i \left( \sum_{j=1}^{M} \tilde{d}_{ij} x_{ij} \right)$$

Cost

In this paper, the cost criterion is based on procurement and adaptation costs of COTS products. The procurement cost contains licensing arrangement cost, product & technology cost and consulting cost. The objective of minimizing the total cost of the COTS selection problem is expressed as:

$$\text{min } z_2 = \sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij} x_{ij}$$

B.3. Model Constraints

Delivery Time Constraints

The delivery time of each COTS alternative and also maximum threshold $\bar{T}$ on the delivery time of the software system is considered uncertain. The delivery time $\bar{T}_i$ of the $i$-th module is expressed as:

$$\bar{T}_i = \sum_{j=1}^{n} \tilde{d}_{ij} x_{ij}, i = 1, 2, ..., k.$$ 

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

$$\text{max } (\bar{T}_i) \leq \bar{T}$$

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

Alternatively, the delivery time constraints may also be reformulated as:

$$\sum_{j=1}^{n} \tilde{d}_{ij} x_{ij} \leq \tilde{T}, i = 1, 2, ..., k.$$ 

One and only one COTS product is selected for each module

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, ..., k.$$ 

Contingent Decision Constraints

In the development of a modular software system, sometimes a COTS product for one module is incompatible with the COTS alternatives for other modules, due to problems such as implementation technology, interfaces and licensing. This incompatibility can be denoted by $x_{ij} \leq x_{u_{ij}}$, that is, if the $r$-th module chooses the $s$-th COTS alternative, then $u$-th module must choose the $t$-th COTS alternative. This condition is called the contingent decision constraint [20]. Suppose there are two contingent decisions in the model, such as, the COTS alternative for the module $r$ is only compatible with the COTS products $t_1$ and $t_2$ for the module $u$, that is, either $x_{u_{ij}} = 1$ if $x_{ij} = 1$ or $x_{u_{ij}} = 1$ if $x_{ij} = 1$. These constraints can be represented as either $x_{u_{ij}} \leq x_{u_{ij}}$ or $x_{u_{ij}} \leq x_{u_{ij}}$. Since the presence of the “either-or” constraint makes the optimization problem nonlinear, it can be linearized by introducing a binary variable $y_i$ as follows:

$$y_i = \begin{cases} 0, & \text{if the } i\text{-th constraint is active}, \\ 1, & \text{if the } i\text{-th constraint is inactive}, \end{cases}$$

Thus, the constraints corresponding to contingent decisions in the COTS selection problem can be restated as follows:

$$x_{ij} - x_{u_{ij}} \leq My_i, h = 1, 2, ..., z.$$ 

$$\sum_{h=1}^{z} y_i = z - 1,$$

$$y_i \in \{0, 1\}, h = 1, 2, ..., z.$$ 

The above conversion guarantees that only one out of the $z$ contingent decision constraints for any COTS product between two modules is active if $M$ is large (larger than the coefficients used in the corresponding constraints).

Constraints on Variables

$$x_{ij} \in \{0, 1\}, i = 1, 2, ..., k, j = 1, 2, ..., n_i.$$ 

B.4. The Decision Model

The fuzzy optimization model of the COTS software selection is formulated as follows:

(BOFOP)

$$\text{max } z_1 = \sum_{i=1}^{k} \sum_{j=1}^{n} \tilde{d}_{ij} x_{ij}$$

$$\text{min } z_2 = \sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^{n} \tilde{d}_{ij} x_{ij} \leq \tilde{T}, i = 1, 2, ..., k,$$ 

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, ..., k,$$ 

$$x_{ij} - x_{u_{ij}} \leq My_i, h = 1, 2, ..., z.$$ 

$$\sum_{h=1}^{z} y_i = z - 1,$$ 

$$y_i \in \{0, 1\}, h = 1, 2, ..., z.$$ 

$$x_{ij} \in \{0, 1\}, i = 1, 2, ..., k, j = 1, 2, ..., n_i.$$
\[ \sum_{k=1}^{z} y_k = z - 1, \quad (4) \]

\[ y_k \in \{0,1\}, h = 1, 2, ..., z, \quad (5) \]

\[ x_{ij} \in \{0,1\}, i = 1, 2, ..., k, j = 1, 2, ..., n. \quad (6) \]

3. Possibilistic Programming Approach

In real-world decision making problems, unavailability or incompleteness of data is an important challenge that imposes a high degree of uncertainty; hence, in such cases the use of probability distribution is not desirable. Further, despite the availability of historical data, the behaviour of model parameters does not necessarily comply with their historical pattern in future. To deal with this challenge in the concerned COTS selection problem, the uncertain model parameters are represented by fuzzy numbers characterized by their possibility distribution. The importance of the possibility theory lies in the fact that much of the information on which human decisions are based is possibilistic rather than probabilistic in nature [41]. It also provides an effective method for proceeding with inherent ambiguous phenomena in determining imprecise environment coefficients and related parameters [5, 15, 25, 36]. Recalling BOFOP model, we convert it into an equivalent auxiliary crisp multi-objective 0-1 programming model using the possibilistic approach.

A. Model the Imprecise Data with Triangular Possibility Distribution

In this study, the pattern of triangular possibility distribution is adopted to represent each fuzzy parameter with a view to improve the computational efficiency and flexibility of fuzzy arithmetic operations. In practical situations, the decision maker can construct the triangular possibility distribution of \( \tilde{z}_i = (c_i^p, c_i^o, c_i^m) \), see Figure 3, based on the three prominent data as follows:

1. The most possible value \( c_i^p \) that definitely belongs to the set of available values (possibility degree = 1 if normalized).
2. The optimistic value \( c_i^o \) that has a very low likelihood of belonging to the set of available values (possibility degree = 0 if normalized).
3. The pessimistic value \( c_i^m \) that has a very low likelihood of belonging to the set of available values (possibility degree = 0 if normalized).

B. An Auxiliary Multi-Objective 0-1 Programming Model

Given the imprecise coefficients in the objective functions and constraint (1) of BOFOP model, in general, one cannot guarantee an optimal solution to the problem. There are several approaches for obtaining compromise solutions in the literature. We apply an extended version of a well-known approach proposed by Lai and Hwang [25] to transform the BOFOP model into an auxiliary crisp linear multiple objective 0-1 programming model. To do so, we apply appropriate strategies for converting the fuzzy objective functions and fuzzy constraints of BOFOP model into the equivalent crisp forms.

![Figure 3. The triangular possibility distribution of \( \tilde{z}_i \).](image)

B.1. Strategy for Treating the Imprecise Objective Functions

We first consider the weighted quality objective function \( (\tilde{Z}_i) \). Geometrically, this imprecise objective function is fully defined by three corner points \( (z_i^m, 1) \), \( (z_i^o, 0) \), and \( (z_i^p, 0) \) and it can be maximized by pushing the three corner points of triangular possibility distribution to the right. Since, the vertical coordinates of the corner points are fixed at either 1 or 0, the three horizontal coordinates are the only considerations. Using Lai and Hwang [25], the strategy adopted here for treating the imprecise weighted quality objective function is to simultaneously maximize the most possible value of weighted quality \( z_i^m \), minimize the risk of obtaining lower quality levels \( (z_i^m - z_i^p) \) (region I of the possibility distribution in Figure 4) and maximize the possibility of obtaining higher quality levels \( (z_i^o - z_i^m) \) (region II of the possibility distribution in Figure 4).

![Figure 4. The strategy to solve fuzzy quality objective.](image)
Using \( \tilde{q}_{ij} = (q_{ij}^p, q_{ij}^m, q_{ij}^u) \), the imprecise quality level of the \( j \)-th COTS alternative in the \( i \)-th module, the fuzzy weighted quality objective function in BOFOP model is replaced by the following three crisp objective functions to obtain a compromise solution:

\[
\begin{align*}
\max z_1 &= z_1^m = \sum_{i=1}^{n} w_i \left( \sum_{j=1}^{k} q_{ij}^m x_{ij} \right) \\
\max z_2 &= (z_2^o - z_2^m) = \sum_{i=1}^{n} w_i \left( \sum_{j=1}^{k} (q_{ij}^o - q_{ij}^m) x_{ij} \right) \\
\min z_3 &= (z_3^p - z_3^m) = \sum_{i=1}^{n} w_i \left( \sum_{j=1}^{k} (q_{ij}^p - q_{ij}^m) x_{ij} \right)
\end{align*}
\]

Similarly, the strategy adopted for treating the imprecise total cost objective function \( (\tilde{z}_2) \) is to simultaneously minimize the most possible value of total cost \( z_2^m \), minimize the risk of obtaining higher costs \( (z_2^o - z_2^m) \) (region II of the possibility distribution in Figure 5) and maximize the possibility of obtaining lower costs \( (z_2^p - z_2^m) \) (region I of the possibility distribution in Figure 5).

\[
\mu_{\tilde{z}_2}
\]

Figure 5. The strategy to solve fuzzy cost objective.

Using \( \tilde{c}_{ij} = (c_{ij}^p, c_{ij}^m, c_{ij}^u) \), the imprecise cost of the \( j \)-th COTS alternative for the \( i \)-th module, the fuzzy total cost objective function in BOFOP model is replaced by the following three crisp objective functions to obtain a compromise solution:

\[
\begin{align*}
\min z_1 &= z_1^m = \sum_{i=1}^{n} \sum_{j=1}^{k} c_{ij}^m x_{ij} \\
\max z_2 &= (z_2^o - z_1^m) = \sum_{i=1}^{n} \sum_{j=1}^{k} (c_{ij}^o - c_{ij}^m) x_{ij} \\
\min z_3 &= (z_3^p - z_1^m) = \sum_{i=1}^{n} \sum_{j=1}^{k} (c_{ij}^p - c_{ij}^m) x_{ij}
\end{align*}
\]

B.2. Strategy for Treating the Imprecise Constraints

To deal with the fuzzy constraint (1) of BOFOP model that have imprecise parameters both in the left-hand side and right-hand side, we use the fuzzy ranking concept [25, 29, 34]. Accordingly, for a given minimum acceptable possibility value \( \beta (0 \leq \beta \leq 1) \) specified by the decision maker, we replace the fuzzy constraint with three equivalent auxiliary inequality constraints. Using \( \tilde{d}_{ij} = (d_{ij}^p, d_{ij}^m, d_{ij}^u) \), the imprecise delivery time of the \( j \)-th COTS alternative for the \( i \)-th module, the following auxiliary crisp delivery time constraints are obtained as:

\[
\begin{align*}
\sum_{j=1}^{n} (d_{ij}^p) x_{ij} &\leq T_p^m, i = 1, 2, \ldots, k, \\
\sum_{j=1}^{n} (d_{ij}^m) x_{ij} &\leq T_p^m, i = 1, 2, \ldots, k, \\
\sum_{j=1}^{n} (d_{ij}^u) x_{ij} &\leq T_p^m, i = 1, 2, \ldots, k,
\end{align*}
\]

where \( (d_{ij}^p) = d_{ij}^m + \beta (d_{ij}^u - d_{ij}^m) \), \( (d_{ij}^m) = d_{ij}^o - \beta (d_{ij}^o - d_{ij}^m) \), and \( T_p^m = T_p^o + \beta (T_p^u - T_p^o) \), \( T_p^o = T_p^o - \beta (T_p^o - T_p^m) \). It may be noted that \( \beta \) is the minimum satisfaction level of the fuzzy constraints or the minimum desired degree of feasibility and its membership function is elicited on the basis of the preference concept. Thus, the desired degree of feasibility may be strongly dependent upon the psychological value system.

B.3. The Multi-Objective 0-1 Model

The multiple objective problem is now presented as follows:

\[
\text{(MOP)}
\]

\[
\begin{align*}
\max z_1 &= z_1^m = \sum_{i=1}^{n} w_i \left( \sum_{j=1}^{k} q_{ij}^m x_{ij} \right) \\
\max z_2 &= (z_2^o - z_1^m) = \sum_{i=1}^{n} w_i \left( \sum_{j=1}^{k} (q_{ij}^o - q_{ij}^m) x_{ij} \right) \\
\min z_3 &= (z_3^p - z_1^m) = \sum_{i=1}^{n} w_i \left( \sum_{j=1}^{k} (q_{ij}^p - q_{ij}^m) x_{ij} \right) \\
\min z_4 &= z_4^m = \sum_{i=1}^{n} \sum_{j=1}^{k} c_{ij}^m x_{ij} \\
\max z_5 &= (z_5^o - z_4^m) = \sum_{i=1}^{n} \sum_{j=1}^{k} (c_{ij}^o - c_{ij}^m) x_{ij} \\
\min z_6 &= (z_6^p - z_4^m) = \sum_{i=1}^{n} \sum_{j=1}^{k} (c_{ij}^p - c_{ij}^m) x_{ij}
\end{align*}
\]

subject to Constraints (2)-(9)

C. Solving the MOP Model

There are several methods in the literature for solving MOP problem; among them the fuzzy programming approaches are being increasingly applied. The main advantage of fuzzy approaches is that they are capable to
measure the satisfaction degree of each objective function explicitly which helps the decision maker to choose a preferred compromise solution according to the satisfaction degree and preference (relative importance) of each objective function. The MOP model developed above can be converted into an equivalent crisp 0-1 linear programming model using Zimmermann’s approach [44]. For the purpose, we first determine the positive ideal solution (PIS) and negative ideal solution (NIS) for each objective function of the MOP model as follows:

\[ z_{11}^{PIS} = \max z_1^n, z_{11}^{NIS} = \min z_1^n \]
\[ z_{12}^{PIS} = \max (z_1^n - z_1^w), z_{12}^{NIS} = \min (z_1^n - z_1^w) \]
\[ z_{13}^{PIS} = \min (z_1^n - z_1^w), z_{13}^{NIS} = \max (z_1^n - z_1^w) \]
\[ z_{21}^{PIS} = \min z_2^n, z_{21}^{NIS} = \max z_2^n \]
\[ z_{22}^{PIS} = \max (z_2^n - z_2^w), z_{22}^{NIS} = \min (z_2^n - z_2^w) \]
\[ z_{23}^{PIS} = \min (z_2^n - z_2^w), z_{23}^{NIS} = \max (z_2^n - z_2^w) \]

subject to Constraints (2)-(9).

It may be noted that determining the above ideal solutions require solving many integer linear programming problems which could be computationally difficult especially in large-sized problem instances. In order to reduce the computational complexity, we may apply the heuristic rules wherein we solve the corresponding integer linear programming problem heuristically to obtain a satisfactory feasible integer solution [21].

Next, we specify the linear membership functions for each objective function of the MOP model as follows:

\[ \mu_{z_1} = \begin{cases} 
1, & \text{if } z_{11} \geq z_{11}^{PIS} \\
\frac{z_{11} - z_{11}^{NIS}}{z_{11}^{PIS} - z_{11}^{NIS}}, & \text{if } z_{11}^{NIS} < z_{11} < z_{11}^{PIS} \\
0, & \text{if } z_{11} \leq z_{11}^{NIS} 
\end{cases} \]  
(10)

\[ \mu_{z_2} = \begin{cases} 
1, & \text{if } z_{12} \geq z_{12}^{PIS} \\
\frac{z_{12} - z_{12}^{NIS}}{z_{12}^{PIS} - z_{12}^{NIS}}, & \text{if } z_{12}^{NIS} < z_{12} < z_{12}^{PIS} \\
0, & \text{if } z_{12} \leq z_{12}^{NIS} 
\end{cases} \]  
(11)

Note that the linear membership functions of \( z_{12} \) and \( z_{22} \) are similar to \( z_{11} \); while \( z_{21} \) and \( z_{23} \) are similar to \( z_{13} \) and hence are not specified explicitly. The graphical representation of linear membership functions defined by (10) and (11) are shown in Figure 6.

We now transform the MOP model into an equivalent crisp single objective 0-1 linear programming model using the max-min approach proposed by Bellman and Zadeh [3]:

\[ \text{(LP)} \max \lambda \]
subject to \( \lambda \leq \mu_{z_g}, g = 1, 2; l = 1, 2, 3, \)
\( 0 \leq \lambda \leq 1, \)
and Constraints (2)-(9),

where the auxiliary variable \( \lambda (0 \leq \lambda \leq 1) \) represents the overall degree of satisfaction with determined objective values. The LP formulation is capable of yielding compromise solutions for a given possibility (feasibility) value \( \beta \) based on the decision maker’s preferences.

D. Solution Procedure

The solution procedure for the fuzzy optimization model of COTS selection problem is summarized as under.

Step 1: Calculate the weight of each module using AHP.

Step 2: Determine appropriate triangular possibility distributions for the imprecise coefficients of both the objective functions and the fuzzy constraint and formulate the BOFOP model of COTS selection.

Step 3: Convert the fuzzy objective functions of the BOFOP model into the crisp objective functions of the MOP model.

Step 4: Given the minimum acceptable possibility (feasibility) value, \( \beta \), convert the fuzzy constraint of the BOFOP model into the...
Step 5: Calculate the PIS and NIS for each objective function of the MOP model and define the linear membership functions of the auxiliary objective functions.

Step 6: Convert the MOP model into its crisp equivalent LP model. Solve and modify the LP model interactively. If the decision maker is not satisfied with the solution obtained then the model may be adjusted until a set of satisfactory solutions is obtained by varying values of the minimum acceptable possibility (feasibility) value $\beta$.

E. Crisp Transformation of Fuzzy Objectives

To find a representative crisp value for the fuzzy number $\tilde{Z}$, we defuzzify $\tilde{Z}$ to a crisp value. Many defuzzification approaches have been proposed in the literature [6]. The center of gravity (COG) method is the most trivial weighted average. We use the COG method to defuzzify the fuzzy total cost and fuzzy weighted quality. Let $Z'$ be the defuzzified value of the fuzzy number $\tilde{Z}$. The COG method calculates $Z'$ as follows:

$$Z' = \frac{\int_{\beta}^{U} z \mu_z(z)dz}{\int_{\beta}^{U} \mu_z(z)dz} \quad (12)$$

If $\mu_z(z)$ is represented by triangular fuzzy number $(z_1, z_2, z_3)$ where $z_1$ and $z_3$ are the lower and upper limits of the support of $\tilde{Z}$ and $z_2$ is the modal value, then we obtain

$Z' = \frac{z_1 + z_2 + z_3}{3} \quad (13)$

by actual substitution of the membership function of the triangular fuzzy number. Using equations (12) and (13), we can obtain crisp equivalents of the fuzzy objective values of the BOFOP model.

4. Numerical Illustrations

In order to demonstrate the validity and practicability of the proposed COTS selection model and the solution method, an industrial case scenario inspired from a component-based software system development is presented. A local software system supplier planned to develop 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 defined by a vendor to provide a set of standard functions that can be adapted to the specific needs of each customer. COTS term refers to application software package. In this paper, we have identified three functional requirements of the ERP system, namely, Finance (Program 1), Operations (Program 2), and Marketing (Program 3). 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 $m_4$ (twice each); Program 3 calls modules $m_1$ (twice), $m_3$ (once), and $m_4$ (twice). Module $m_1$ has three COTS alternatives denoted as $sc_{11}$, $sc_{12}$, and $sc_{13}$. Module $m_2$ has two COTS alternatives denoted as $sc_{21}$ and $sc_{22}$. Module $m_3$ has three COTS alternatives denoted as $sc_{31}$, $sc_{32}$, and $sc_{33}$. Module $m_4$ has three COTS alternatives denoted as $sc_{41}$, $sc_{42}$, and $sc_{43}$. The diagrammatic depiction of ERP software system considered in this paper is given in Figure 7.

A. Data Description

Because of confidentiality as well as the lack of some exact required data, we have used fuzzy parameters characterized by triangular possibility distributions in accordance with the past data and integrating the judgement of the decision maker for the numerical experiments. The triangular possibility distributions of the individual quality level, cost, and delivery time of all the COTS components are shown in Table 1.

B. Module Weights using AHP

The optimization model used in this paper for COTS selection requires systematic evaluation of the user’s and software developer’s preferences about the software in terms of the programs and modules.

B.1. The Hierarchical Basis of the System Prototype

Referring to Figure 7, we begin the hierarchy from the top with the user’s view, which we define as the user’s overall assessment of the quality of the software system. The user’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 of this study consists of three programs denoted by Program 1, Program 2, and Program 3. The third level of the hierarchy lists the four independent modules from software developer’s view of which the programs are composed of. These modules may have submodules but each submodule belongs to only one module, i.e. a many-to-one relationship. We
stop the hierarchical structure at the level of the independent modules.

Figure 7. The hierarchical structure of the ERP software system.

B.2. Decision Maker’s Relative Preference

Our goal is to identify the relative importance of each program at the second level (user’s view) and each module (software developer’s view) at the third level of the hierarchy in the assessment of the quality of the software system at the first level. Clearly, one is unable to ask directly from the user or software developer alone to express his preferences for each program and each module because his view of the software is only partial view of the system. Hence, to achieve our goal, we use AHP technique [31] for identifying the relative importance of the programs and the modules.

In AHP, the elements of each level of the decision hierarchy are rated using pair-wise comparison. After all the elements have been compared pair by pair, a paired comparison matrix is formed. The order of the matrix depends on the number of elements at each level. The number of such matrices at each level depends on the number of elements at the immediate upper level that it links to. After developing all the paired comparison matrices, the eigenvectors or the relative weights representing the degree of the relative importance amongst the elements and the maximum eigenvalue ($\lambda_{\text{max}}$) are calculated for each matrix.

The ($\lambda_{\text{max}}$) value is an important validating parameter in AHP. It is used as a reference index to screen information by calculating the consistency ratio of the estimated vector in order to validate whether the paired comparison matrix provides a completely consistent evaluation. The consistency ratio is calculated as per the following steps:

1. Calculate the eigenvector or the relative weights and ($\lambda_{\text{max}}$) for each matrix of order $n$
2. Compute the consistency index ($CI$) for each matrix of order $n$ as follows:

$$CI = \frac{\lambda_{\text{max}} - n}{(n-1)}$$

3. The consistency ratio ($CR$) is calculated as follows:

$$CR = \frac{CI}{RI}$$

Table 1. Input data of the system prototype.

<table>
<thead>
<tr>
<th>COTS products</th>
<th>Quality</th>
<th>Cost</th>
<th>Delivery time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$sc_{11}$</td>
<td>$\tilde{q}_{11} = (0.80, 0.83, 0.90)$</td>
<td>$\tilde{c}_{11} = (5, 10, 13)$</td>
<td>$\tilde{d}_{11} = (4, 5, 7)$</td>
</tr>
<tr>
<td>$sc_{12}$</td>
<td>$\tilde{q}_{12} = (0.78, 0.82, 0.87)$</td>
<td>$\tilde{c}_{12} = (7, 9, 14)$</td>
<td>$\tilde{d}_{12} = (3, 4, 6)$</td>
</tr>
<tr>
<td>$sc_{13}$</td>
<td>$\tilde{q}_{13} = (0.74, 0.78, 0.82)$</td>
<td>$\tilde{c}_{13} = (5, 8, 11)$</td>
<td>$\tilde{d}_{13} = (5, 7, 10)$</td>
</tr>
<tr>
<td>$sc_{21}$</td>
<td>$\tilde{q}_{21} = (0.83, 0.85, 0.88)$</td>
<td>$\tilde{c}_{21} = (1, 8, 12)$</td>
<td>$\tilde{d}_{21} = (5, 7, 10)$</td>
</tr>
<tr>
<td>$sc_{22}$</td>
<td>$\tilde{q}_{22} = (0.82, 0.88, 0.93)$</td>
<td>$\tilde{c}_{22} = (3, 9, 11)$</td>
<td>$\tilde{d}_{22} = (3, 6, 8)$</td>
</tr>
<tr>
<td>$sc_{31}$</td>
<td>$\tilde{q}_{31} = (0.82, 0.85, 0.90)$</td>
<td>$\tilde{c}_{31} = (7, 8, 14)$</td>
<td>$\tilde{d}_{31} = (1, 3, 6)$</td>
</tr>
<tr>
<td>$sc_{32}$</td>
<td>$\tilde{q}_{32} = (0.77, 0.79, 0.84)$</td>
<td>$\tilde{c}_{32} = (4, 7, 11)$</td>
<td>$\tilde{d}_{32} = (3, 7, 9)$</td>
</tr>
<tr>
<td>$sc_{33}$</td>
<td>$\tilde{q}_{33} = (0.85, 0.90, 0.98)$</td>
<td>$\tilde{c}_{33} = (6, 9, 13)$</td>
<td>$\tilde{d}_{33} = (3, 5, 8)$</td>
</tr>
<tr>
<td>$sc_{41}$</td>
<td>$\tilde{q}_{41} = (0.84, 0.90, 0.95)$</td>
<td>$\tilde{c}_{41} = (3, 9, 11)$</td>
<td>$\tilde{d}_{41} = (3, 6, 8)$</td>
</tr>
<tr>
<td>$sc_{42}$</td>
<td>$\tilde{q}_{42} = (0.81, 0.88, 0.96)$</td>
<td>$\tilde{c}_{42} = (2, 8, 12)$</td>
<td>$\tilde{d}_{42} = (4, 8, 10)$</td>
</tr>
<tr>
<td>$sc_{43}$</td>
<td>$\tilde{q}_{43} = (0.77, 0.81, 0.87)$</td>
<td>$\tilde{c}_{43} = (6, 8, 11)$</td>
<td>$\tilde{d}_{43} = (3, 6, 8)$</td>
</tr>
</tbody>
</table>
where $R_I$ is a known random consistency index that has been obtained from a large number of simulation runs and varies according to the order of matrix.

If $CI$ is sufficiently small then pair-wise comparisons are probably consistent enough to give useful estimates of the weights. If $\frac{CI}{R_I} \leq 0.10$, then the degree of consistency is satisfactory. However, if $\frac{CI}{R_I} > 0.10$, then serious inconsistencies may exist and hence AHP may not yield meaningful results. The evaluation process should, therefore, be reviewed and improved. Also, the number of pairwise comparisons depends on (i) number of alternatives to be evaluated and (ii) number of criteria considered for evaluation. If user requirements of the COTS products changes then relative score of each alternative over the other with respect to each evaluation criteria also changes. If number of alternatives considered for evaluation increases then number of pairwise comparisons also increases. Therefore in both the cases: (i) if user needs of the components changes, or (ii) number of alternatives to be evaluated changes, pairwise comparison must be done again.

An illustrative pair-wise comparison matrix of the attributes (Program 1, Program 2, and Program 3) with respect to the goal, i.e. quality of software system is shown in Table 2. As seen in this table, the preference scores of quality of 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 2. Comparison matrix of the programs w.r.t. the 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>½</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

As seen from Figure 7, Program 1 calls upon modules 1 and 2. The pair-wise comparison matrix of the modules 1 and 2 with respect to Program 1 is shown in Table 3.

Table 3. Comparison matrix of the modules w.r.t. the Program 1. (attribute)

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Module 1</th>
<th>Module 2</th>
<th>Relative preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 1</td>
<td>1</td>
<td>2</td>
<td>0.667</td>
</tr>
<tr>
<td>Module 2</td>
<td>½</td>
<td>1</td>
<td>0.333</td>
</tr>
</tbody>
</table>

On the same lines the pair-wise comparison matrix of the modules 2 and 3 with respect to Program 2 and modules 1, 3, and 4 with respect to Program 3 are shown in Tables 4 and 5, respectively.

Table 4. Comparison matrix of the modules w.r.t. the Program 2. (attribute)

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Module 2</th>
<th>Module 3</th>
<th>Relative preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 2</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>Module 3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 5. Comparison matrix of the modules w.r.t. the Program 3. (attribute)

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Module 1</th>
<th>Module 3</th>
<th>Module 4</th>
<th>Relative preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0.429</td>
</tr>
<tr>
<td>Module 3</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>0.143</td>
</tr>
<tr>
<td>Module 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 local 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 local weights of Module 1 in relation to the attributes -Program 1 and Program 3, are multiplied with the corresponding local weights of these attributes in relation to their parent attribute, i.e. quality of software system. The AHP weight of Module 1 is then determined by adding these products, i.e. the AHP weight of Module 1 is obtained as: 0.667*0.411+0.429*0.328=0.415. Similarly, the final weights of modules 2, 3, and 4 can be calculated. Thus, we obtain the following AHP 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 software developer in pair-wise comparisons. Further, 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 pair-wise comparisons to obtain the weights. However, absolute rating of elements is less accurate than pair-wise comparisons in eliciting the preferences of the decision maker.

B.3. The Auxiliary MOP Model

We use fuzzy input data of the system prototype from Section 4.A, module weight vector $W= (0.415, 0.268, 0.177, 0.140)$ from Section 4.B.2 and maximum threshold on the delivery time of the software system as $\bar{T}= (3,8,12)$ . Consider the following contingent decision constraints in our COTS selection problem:
\[ x_{42} \leq x_{11} \text{ or } x_{42} \leq x_{13} \tag{14} \]

Constraint (14) implies that if Module 4 chooses the second COTS product then Module 1 must choose the first COTS product or the third COTS product. Considering \( \beta = 0.5 \), we obtain the following auxiliary MOP model:

\[
\begin{align*}
\text{max } z_{11} &= 0.415(0.83x_{41} + 0.82x_{42} + 0.78x_{43}) + 0.268(0.85x_{21} + 0.88x_{22}) + 0.177(0.85x_{31} + 0.79x_{32} + 0.90x_{33}) + 0.140(0.90x_{41} + 0.88x_{42} + 0.81x_{43}) \\
\text{max } z_{12} &= 0.415(0.07x_{11} + 0.05x_{12} + 0.04x_{13}) + 0.268(0.03x_{21} + 0.05x_{22}) + 0.177(0.05x_{31} + 0.05x_{32} + 0.08x_{33}) + 0.140(0.05x_{41} + 0.08x_{42} + 0.06x_{43}) \\
\text{min } z_{13} &= 0.415(0.03x_{11} + 0.04x_{12} + 0.04x_{13}) + 0.268(0.02x_{21} + 0.06x_{22}) + 0.177(0.03x_{31} + 0.02x_{32} + 0.05x_{33}) + 0.140(0.06x_{41} + 0.07x_{42} + 0.04x_{43}) \\
\text{min } z_{21} &= 10x_{11} + 9x_{12} + 8x_{13} + 8x_{21} + 9x_{22} + 8x_{31} + 7x_{32} + 9x_{41} + 8x_{42} + 8x_{43} \\
\text{max } z_{22} &= 5x_{11} + 2x_{12} + 3x_{13} + 7x_{21} + 6x_{22} + x_{31} + 3x_{32} + 3x_{33} + 6x_{41} + 6x_{42} + 2x_{43} \\
\text{min } z_{23} &= 3x_{11} + 5x_{12} + 3x_{13} + 4x_{21} + 2x_{22} + 6x_{31} + 4x_{32} + 4x_{33} + 2x_{41} + 4x_{42} + 3x_{43} \\
\text{subject to} \\
x_{11} + x_{12} + x_{13} &= 1, \quad x_{21} + x_{22} + 1, \quad x_{31} + x_{32} + x_{33} = 1, \quad x_{41} + x_{42} + x_{43} = 1, \\
4.5x_{11} + 3.5x_{12} + 6x_{13} &\leq 5.5, \\
5x_{21} + 4x_{22} + 7x_{23} &\leq 8, \\
6x_{31} + 5x_{32} + 8.5x_{33} &\leq 10, \\
6x_{41} + 4.5x_{42} &\leq 5.5, \\
7x_{41} + 6x_{42} &\leq 8, \\
8.5x_{41} + 7x_{42} &\leq 10, \\
2x_{31} + 5x_{32} + 4x_{33} &\leq 5.5, \\
3x_{41} + 7x_{42} + 5x_{43} &\leq 8, \\
4.5x_{41} + 8x_{42} + 6.5x_{43} &\leq 10, \\
4.5x_{41} + 6x_{42} + 4.5x_{43} &\leq 5.5, \\
6x_{41} + 8x_{42} + 6x_{43} &\leq 8, \\
7x_{41} + 9x_{42} + 7x_{43} &\leq 10, \\
x_{42} - x_{11} &\leq 5y_{1}, \\
x_{42} - x_{13} &\leq 5y_{2}, \\
y_{1} + y_{2} &= 1, \\
y_{1}, y_{2} &\in \{0, 1\}, \\
x_{y} &\in \{0, 1\}, \forall i, j. 
\end{align*}
\]

Next, we calculate PIS and NIS for each objective function of the MOP model. The computational results are presented in Table 6.

**Table 6. PIS and NIS for fuzzy objective functions.**

<table>
<thead>
<tr>
<th>Objectives</th>
<th>( z_{11} )</th>
<th>( z_{12} )</th>
<th>( z_{13} )</th>
<th>( z_{21} )</th>
<th>( z_{22} )</th>
<th>( z_{23} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIS</td>
<td>0.86559</td>
<td>0.06501</td>
<td>0.03767</td>
<td>33</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>NIS</td>
<td>0.82937</td>
<td>0.05</td>
<td>0.04993</td>
<td>37</td>
<td>11</td>
<td>16</td>
</tr>
</tbody>
</table>

B.4. The Equivalent Crisp LP Model

The linear membership function for each objective function in the auxiliary MOP model can be defined using (10) and (11). The equivalent LP model is obtained using the max-min approach to aggregate all the fuzzy sets.

\[
\begin{align*}
\text{max } \lambda \\
\text{subject to} \\
0.415(0.83x_{41} + 0.82x_{42} + 0.78x_{43}) + 0.268 (0.85x_{21} + 0.88x_{22}) + 0.177(0.85x_{31} + 0.79x_{32} + 0.90x_{33}) + 0.140(0.90x_{41} + 0.88x_{42} + 0.81x_{43}) &\geq 0.036222\lambda \geq 0.82937; \\
0.415(0.07x_{11} + 0.05x_{12} + 0.04x_{13}) + 0.268 (0.03x_{21} + 0.05x_{22}) + 0.177(0.05x_{31} + 0.05x_{32} + 0.08x_{33}) + 0.140(0.05x_{41} + 0.08x_{42} + 0.06x_{43}) &\geq 0.01501\lambda \geq 0.05; \\
0.415(0.03x_{11} + 0.04x_{12} + 0.04x_{13}) + 0.268 (0.02x_{21} + 0.06x_{22}) + 0.177(0.03x_{31} + 0.02x_{32} + 0.05x_{33}) + 0.140(0.06x_{41} + 0.07x_{42} + 0.04x_{43}) &\geq 0.01226\lambda \leq 0.04993; \\
10x_{11} + 9x_{12} + 8x_{13} + 8x_{21} + 9x_{22} + 8x_{31} + 7x_{32} + 9x_{33} + 9x_{41} + 8x_{42} + 8x_{43} + 4x_{43} &\leq 37, \\
5x_{11} + 2x_{12} + 3x_{13} + 7x_{21} + 6x_{22} + x_{31} + 3x_{32} + 3x_{33} + 6x_{41} + 6x_{42} + 2x_{43} - 9\lambda &\leq 11, \\
3x_{11} + 5x_{12} + 3x_{13} + 4x_{21} + 2x_{22} + 6x_{31} + 4x_{32} + 4x_{33} + 2x_{41} + 4x_{42} + 3x_{43} + 5\lambda &\leq 16, \\
0 \leq \lambda \leq 1, \\
\text{and Constraints (15)–(35).} 
\end{align*}
\]

We use LINDO software [32] to solve the above LP model. The obtained solutions have the following triangular possibility distributions:
\( \bar{z}_1 \) (weighted quality) = (0.80565, 0.84612, 0.90442),
\( \bar{z}_2 \) (cost) = (15, 35, 46),
and \( \lambda \) (overall degree of satisfaction) = 0.462452.

The solutions variables are \( x_{11}, x_{22}, x_{32}, \) and \( x_{41} \). The selected COTS products for modules 1, 2, 3 and 4 are \( SC_{11}, SC_{22}, SC_{32}, \) and \( SC_{41} \), respectively. It may be noted that the quality of the software system is fuzzy, its most likely value falls at 0.84612, and its value is impossible to fall outside the range of 0.80565 and 0.90442. Similarly, the total cost of the software system is fuzzy, but its most likely value falls at 35, and its value is impossible to fall outside the range of 15 and 46. Thus, the obtained quality and cost ranges provide a useful reference for the software development team of the ERP software system.

B.5. Sensitivity Analysis w.r.t. \( \beta \)-values

If the decision maker is not satisfied with the obtained COTS selection, more COTS selection strategies can be generated by varying value of the minimum acceptable possibility (feasibility) parameter, \( \beta \), in the MOP model. Table 7 lists the results of implementing different \( \beta \)-values.

The possibility distributions of the obtained solutions using different \( \beta \)-values for the fuzzy quality and cost objective functions are shown in Figure 8 and Figure 9, respectively.

![Figure 8. The possibility distributions for quality objective.](image)

![Figure 9. The possibility distributions for cost objective.](image)

Table 7. Results of sensitivity analysis on \( \beta \)-values.

<table>
<thead>
<tr>
<th>( \beta )-value</th>
<th>Degree of satisfaction</th>
<th>Objective values</th>
<th>COTS products for modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>( \lambda = 0.3333 )</td>
<td>( \mu_{11} = 0.60711 )</td>
<td>( (0.80171, 0.84884, 0.90555) )</td>
</tr>
<tr>
<td></td>
<td>( \mu_{12} = 0.34525 )</td>
<td>( \mu_{13} = 0.3333 )</td>
<td>( \mu_{14} = 0.6667 )</td>
</tr>
<tr>
<td>0.5</td>
<td>( \mu_{21} = 0.46245 )</td>
<td>( \mu_{22} = 0.55296 )</td>
<td>( \mu_{23} = 0.7716 )</td>
</tr>
<tr>
<td></td>
<td>( \mu_{24} = 0.5 )</td>
<td>( \mu_{31} = 1 )</td>
<td>( \mu_{32} = 0.5 )</td>
</tr>
<tr>
<td>0.9</td>
<td>( \mu_{33} = 1 )</td>
<td>( \mu_{34} = 0.5 )</td>
<td>( \mu_{41} = 1 )</td>
</tr>
<tr>
<td></td>
<td>( \mu_{42} = 0.54798 )</td>
<td>( \mu_{43} = 0.50775 )</td>
<td>( \mu_{44} = 0.8782 )</td>
</tr>
</tbody>
</table>
obtained by solving the single objective linear programming problem corresponding to each objective function. It may be noted that all the obtained solutions are efficient solutions. As shown in Table 7, the change in $\beta$-value influences the overall degree of satisfaction $\lambda$, individual degree of satisfaction of the auxiliary objective functions $\mu_{z_p}, g = 1, 2; l = 1, 2, 3$, objective function values and solution variables. The $\lambda$-value increases from 0.33 to 0.5, when the $\beta$-value increases from 0.1 to 0.9. Generally, the achievement level may not be large enough to satisfy the decision maker because of multiple objective nature of the problem and also considering the fact that we are dealing with 0-1 programming problem.

Next, by defuzzifying the fuzzy total cost and fuzzy weighted quality using (12) and (13), the crisp values of the two fuzzy objectives $\tilde{z}_1$ and $\tilde{z}_2$ are provided in Table 8.

<table>
<thead>
<tr>
<th>Objective</th>
<th>$\beta$-value</th>
<th>Fuzzy value</th>
<th>Crisp value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>0.1</td>
<td>(22,35,49)</td>
<td>35.33</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>(15,35,46)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>(13,34,47)</td>
<td>31.33</td>
</tr>
<tr>
<td>Weighted</td>
<td>0.1</td>
<td>(0.80171,0.84884,0.90555)</td>
<td>0.85203</td>
</tr>
<tr>
<td>quality</td>
<td>0.5</td>
<td>(0.80565,0.84612,0.90442)</td>
<td>0.85206</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>(0.80833,0.83808,0.89102)</td>
<td>0.84581</td>
</tr>
</tbody>
</table>

Table 8. Crisp values of the fuzzy objective functions.

It can be seen from Table 8 that accepting deterioration in one of the objective function may provide improvement in the other objective function. In most cases, it is possible that poor performance on one criterion may be compensated by good performance on other criteria. Thus, corresponding to decision maker’s preferences, many different solutions can be reached at different satisfaction degrees. In the proposed methodology, both the quality and cost objectives are treated equivalently in the LP model. However, the decision maker can choose different solutions according to his needs in different situations. For example, cost (quality) can be the most important objective for the decision maker in a determined period of software development. Then, the decision maker will choose the solution which satisfies the cost (quality) objective function the most. However, this can cause poor performance of the satisfaction degree of other objective(s). Therefore, different solutions at different $\beta$-values will enable the decision maker to determine appropriate COTS selection strategies using quality and cost objectives. The proposed solution method provides a wide range of information and flexibility in the sense that by changing possibility (feasibility) values interactively provides different scenario analysis for imprecise situations. Also, it is possible that the lower (upper) bounds of the objective functions can be changed using decision maker’s preferences so that new membership functions are generated. This will lead to the formulation of a new crisp LP model. Hence, possibly a new compromise solution may be obtained.

5. Comparison and Benefits of the Proposed Approach

To better justify the applicability of the proposed interactive possibilistic approach for COTS selection, we present the performance analysis and the benefits of the proposed approach.

A. Performance Analysis

We evaluate performance of the proposed interactive possibilistic programming approach and the fuzzy interactive approach developed by Gupta et al. [13] for COTS selection.

To analyze and compare performance of the interactive approaches, we use a well-known performance measure, namely, distance measure. The distance measure is used for determining the degree of closeness of each solution to the corresponding best solution. In this regard, we define the following family of distance functions [14, 38, 43]:

$$d_p(x) = \left[ \sum_{g=1}^{2} \sum_{l=1}^{3} (1 - \mu_{z_p}(x))^p \right]^{1/p};$$

$p \geq 1$ and integer, \(36\)

where $\gamma_{gl}, g = 1, 2; l = 1, 2, 3$ indicates relative importance of the objective functions.

Since the satisfaction degree of each objective is defined as the relative closeness of the solution to the best point or the relative remoteness to the worst point, they are used explicitly in (36). The power $p$ represents a distance parameter and especially $p = 1, 2$, and $\infty$ are operationally important so that $d_1$ (the Manhattan distance) and $d_2$ (the Euclidean distance) are the longest and shortest distances in the geometrical sense; and $d_\infty$ (the Tchebycheff distance) is the shortest distance in the numerical sense. Generally, when $p$ increases, the amount of distance $d_p$ and also the credibility of the distance function $d_p$ decreases [14, 38, 43]. It may be noted that based on the definition of $d_p$, the fuzzy approach with minimum $d_p$ (especially for $p = 1$), would be preferred to the other methods.

To evaluate performance of both the interactive approaches, we solve the biobjective model BOFOP using the input data provided in Table 1, AHP weight vector $W = (0.415, 0.268, 0.177, 0.140)$ and $\beta = 0.5$ with...
both the interactive approaches. It may be noted that to solve the BOFOP model using the interactive approach of Gupta et al. [13] we defuzzify the imprecise data (triangular fuzzy numbers) in respect of the fuzzy total cost and fuzzy weighted quality using (13). Note that the data in respect of delivery time is used in imprecise form (triangular fuzzy numbers) only. The computational results are summarized in Table 9. For the given illustration, it is assumed that $\gamma_1 = \gamma_2 = \gamma_3 = 1$, $\gamma_21 = \gamma_22 = \gamma_23 = \frac{1}{6}$. It may be noted that the above defined family of distance functions needs to be modified appropriately in order to apply on the interactive approach of Gupta et al. [13]. From Table 9, it is clear that the proposed interactive possibilistic programming approach provides a compromise solution better than the solution obtained by the fuzzy interactive approach [13] for the distance functions $d_1$ and $d_2$.

B. Benefits of the Proposed Approach

The managerial implications in applying the proposed methodology to practical COTS selection are as follows.

The proposed approach yields a compromise solution for a fuzzy COTS selection problem. For a given acceptable possibility (feasibility) level $\beta$, the obtained solutions provide a range in which the objectives, that is, the quality of the software system and the total cost of the software system will appear. To exemplify, for $\beta = 0.5$, the obtained objective value of the quality of the software system is $(0.80565, 0.84612, 0.90442)$ and the total cost of the software system is $(15, 35, 46)$. The triangular distribution range of quality so obtained is interpreted as: the most likely value of the quality is 0.84612, the best value of the quality is 0.90442, and the worst value of the quality is 0.80565. Similarly, the most likely value of the cost is 35, the best value of the cost is 15, and the worst value of the cost is 46. Also, it is impossible to have the quality $< 0.80565$ and $> 0.90442$ and cost $< 15$ and $> 46$.

The solutions presented in Table 7 reveal that the changes in the acceptable possibility (feasibility) $\beta$ value influence both the overall degree of satisfaction $\lambda$ and objective values. If the aspiration levels are improperly given, they may create a more complicated procedure. For this reason, we use the PIS and NIS values in the solution approach.

The benefits of the proposed solution approach are summarized as under.

1. The proposed approach effectively handles vagueness and imprecision in the goal values using triangular fuzzy numbers to model the imprecise data. Several different patterns including triangular, trapezoid, S-curve, exponential, hyperbolic are used to model imprecise data in the existing literature. Among them, the triangular distribution is used most often to represent imprecise data owing to the ease it offers in defining the maximum and minimum limits of deviation of a fuzzy number from its central value; although, certain practical applications may prefer other patterns.

2. The proposed approach exhibits greater computational efficiency and flexibility by employing linear membership functions to specify the fuzzy goal values. The main advantage of using a linear membership function is that it produces a computationally tractable function that closely reflects the real-world structure of the decision

<table>
<thead>
<tr>
<th>Degree of satisfaction</th>
<th>Objective values</th>
<th>COTS products for modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda = 0.462452$</td>
<td>$\mu_{z_1} = 0.46245$</td>
<td>$\mu_{z_2} = 0.55296$</td>
</tr>
<tr>
<td></td>
<td>$(0.80565, 0.84612, 0.90442)$</td>
<td>$15, 35, 46$</td>
</tr>
<tr>
<td>Fuzzy interactive approach [13]</td>
<td>$\lambda = 0.00015$</td>
<td>$\mu_{z_1} = 0.00015$</td>
</tr>
<tr>
<td></td>
<td>0.871533</td>
<td>34</td>
</tr>
</tbody>
</table>
3. problem. A linear membership function is most commonly used because it is simple and it is defined by fixing two points: the upper and lower levels of acceptability; however, certain decision-situations may prefer nonlinear patterns.

4. The proposed approach solves COTS selection problem with imprecise coefficients more efficiently since it preserves original linear form for the fuzzy goals. The proposed method can be easily implemented using commercially available software such as LINDO [32] and other related mathematical programming software.

6. Conclusions

In real-world COTS selection problem, the decision maker must simultaneously handle conflicting objectives. Moreover, input data and/or related model parameters are often uncertain in non-stochastic sense because either information is incomplete or unavailable over the planning horizon. This work has developed an interactive approach for solving a bi-objective fuzzy COTS selection model considering imprecise quality, cost and delivery time coefficients with triangular possibility distributions. The interactive approach based on possibility theory developed in this paper provide a systematic framework that facilitates software development process of a modular software, enabling the decision maker to interactively modify the possibility (feasibility) levels until a set of satisfactory solutions is obtained. A real-world case study for numerical illustrations has been provided to demonstrate the feasibility of applying the interactive approach to the COTS selection problem.

The main limitation of the approach developed here is the assumption of triangular possibility distributions to represent imprecise coefficients in the objective functions and constraints. In real-world COTS selection problem, the decision maker may generate suitable possibility distributions based on subjective judgments and/or historical data. Thus, it would be interesting to apply trapezoidal, bell-shaped, exponential, hyperbolic or other possibility distribution patterns for representing imprecise numbers in solving COTS selection model using the proposed approach. However, it may be noted that more general forms of possibility distributions increases the complexity of solving the fuzzy optimization problem. The use of AHP in this paper depends on the number of alternatives and the number of criteria used for evaluation. Although, excellent software systems are available for implementing the AHP technique, yet, one may explore the possibility of using some other techniques for pairwise comparisons which are computationally less time consuming and more reliable. The bi-objective fuzzy COTS selection model studied in this work concerns the assumption that both the quality and cost objective functions have same relative importance and that the operation of the COTS product is statistically independent. Thus, the proposed model can be extended further to make it better suited to practical applications by considering different levels of importance for the objective functions. Also, the single application development task considered in this paper can be extended to multiple application tasks incorporating the reusability of COTS products.

Acknowledgment

We are thankful to the Editor-in-Chief, Associate Editor, and the anonymous referees for their valuable comments and suggestions to improve presentation of the paper. The first author acknowledges the research grant received under a scheme for strengthening R & D Doctoral Research Programme of University of Delhi, Delhi, India.

References


Pankaj Gupta received his Ph.D. degree from Department of Mathematics, University of Delhi, Delhi, India in 2000. Currently he is associated with the Department of Operational Research, University of Delhi, Delhi, India. His research interests include optimization: theory and applications, fuzzy optimization and portfolio optimization.


Hoang Pham received the M.S. degree in Statistics from the University of Illinois at Urbana Champaign; and the M.S., and Ph.D. degrees in Industrial Engineering from State University of New York at Buffalo. He is a professor and chairman of the Department of Industrial and Systems Engineering at Rutgers University. He is the author or coauthor of five books: Software Reliability (Springer, 2000), System Software Reliability (2006), Reliability and Optimal Maintenance (2006), Maintenance for Industrial Systems (2010), and Software Reliability Assessment with OR Applications (2011). He is the editor of eight books including the Handbook of Reliability Engineering (2003), and Handbook of Engineering Statistics (2006), among many others. He is also the editor of Springer Series in Reliability Engineering, and has published over 120 journal articles. He is the editor-in-chief of the International Journal of Reliability, Quality and Safety Engineering; and is an editorial board member of several journals. He is an associate editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. His recent awards include 2009 IEEE Reliability Society Engineer of the Year Award. He is a fellow of the IEEE, and IIE.

Mukesh Kumar Mehlawat received his Ph.D. degree from Department of Operational Research, University of Delhi, Delhi, India in 2011. Currently he is associated with the Department of Operational Research, University of Delhi, Delhi, India. His research interests include: optimization theory and applications, fuzzy optimization and financial optimization. He has published many research papers in reputed international journals including Fuzzy Sets and Systems, Information Sciences, Expert Systems with Applications, Journal of Global Optimization, Optimization Letters, TOP, Knowledge-Based Systems, International Journal of Fuzzy Systems, Insurance: Mathematics and Economics, IEEE Transactions on Fuzzy Systems, International Journal of Information Technology and Decision Making. He has received many teaching and research awards.

Shilpi Verma is working as an Assistant Professor in Northern India Engineering College, New Delhi, India. Presently, she is pursuing her Ph.D. degree from Department of Operational Research, University of Delhi, Delhi, India. Her research interests include fuzzy optimization and COTS selection. She has published few research papers in reputed international journals.