CHAPTER 5
FUZZY MULTI-OBJECTIVE APPROACH TO COMPONENT SELECTION FOR COTS BASED MODULAR SOFTWARE SYSTEM INCORPORATING MANDATORY REDUNDANCY FOR CRITICAL MODULES

In business, industry and in our personal life, we use a number of systems that are engineered and technology driven. They are driven by information technology. Such systems perform important operations but all of them are not critical. Failure of such systems may not always have severe damaging economic consequences. At the most, failure may cause inconvenience to the user of the system. The system whose failure results in chaos, financial loss, severe damages are termed as critical [Jawadekar, (2004)]. These systems play a very pivotal role in business operations and have following essential quality features:

- Available for use all the time
- Dependable in performance
- Reliable in results and operations
- Safe to operate without damage
- Secured from unauthorized interventions, accidental or otherwise

The measure of availability, dependability, reliability, safety and security is its probability, indicating high-level assurance of quality. The probability is a measure of


system’s capability to be ‘up and running’ with efficiency and effectiveness. The cost of development of critical systems is very high, if one has to achieve a high probability score on these features. Let us present a couple of critical systems whose score on availability, dependability, reliability, safety and security is very high. In management science such critical systems are classified as mission critical, business critical and operation critical systems.

➢ **Mission Critical System (MCS)**

These systems are those that affect and have significant impact on the goal. The goals for MCS are long term, strategic and are set for achieving high-level performance. The failure of such a system will be damaging to business due to non achievement of mission critical goals. The software supporting mission critical systems is called a mission critical software system. Examples of mission critical software are

- Business forecasting, capacity planning and resource management
- Business modeling and simulation systems
- Enterprise software like supply chain management and product life cycle management

➢ **Business Critical Systems (BCS)**

These systems are those that affect short-term business goals and functional targets of the business. Failure of BCS continuously affects not only short-term business goals but have an impact on mission critical systems and their goals. Examples of BCS are

- Retail banking
- Production, planning and scheduling
- Customer Relations Management (CRM)
- Enterprise Resource Planning (ERP)
- Firewall and secured information access systems

Failure of BCS leads to customer dissatisfaction, financial loss, under utilization of resources and capacity, and poor returns on investment.

➢ **Operation Critical Systems (OCS)**

These systems are those that affect the day to day operations of the business. Their failure causes delay, disservice, temporary financial loss, extra cost to make up the shortfalls of targets and so on. The examples of OCS are
Transaction processing systems like billing, ordering, inventory processing, tracking and monitoring, ATM card processing, e-mail communication and messaging, data capturing and so on.

A reliable critical system is developed using two approaches: fault minimization and fault tolerance. Use of programming constructs such as goto statements, pointers, recursion, inheritance and floating point numbers are error prone and therefore should be avoided in the program. The program execution should be concluded only after verification and validation of input-process-output at each stage. The program architecture and code should be tested through formal and informal methods, assuring that SRS is completely dealt in the program. The system and program are termed fault tolerant, if they are capable of failure detection, damage limitation, failure recovery, and repairing by fault(s) eradication. *This chapter focuses on developing a critical system by using fault tolerant approach.*

Component Based Software Engineering (CBSE) is a method of software development where tested components are used to expedite software development and to ensure the quality of the software. CBSE approach involves identification of modules for all the functional requirements of the system. Various components are integrated to form a module. The components could be secured from COTS or internal component libraries. If the components are not available, then they can be built in-house. The in-house components can be developed from the scratch or by reusing and modifying the existing components. The choice of component is influenced by the design, architecture of the software, technology, platform chosen, limitations due to interface and security management needs. CBSE methods become successful over a period of time, when it is chosen as a software development strategy. The aim of this chapter is to develop optimization model for selection of components for the modules of the fault tolerant software system incorporating the issue of critical modules. If a module is identified as a critical module then it is mandatory to create redundancy in that module to prevent system from failure. Therefore, identification of critical modules is an important activity in designing a software system using CBSE approach.
When we develop the application with finite budgets, it is necessary to utilize the resources in an optimized way without compromising on quality and reliability. In order to achieve this we need to have optimization models which will minimize the cost and maximize the reliability and also ensures that the application runs without fail. Therefore there is a need for developing the system with a build in redundancy for the critical modules so that even if a particular critical module fails, the other one will takeover. We can consider to introduce redundancy for the critical modules and also at the same time choose the modules which meet the required reliability criteria and cheaper to acquire and use.

The objective of this chapter is to discuss the problems of optimal component selection in designing of a fault tolerant modular software system under consensus recovery block scheme incorporating mandatory redundancy for critical modules. This chapter is divided into three sections. In Section 5.1 fuzzy multi-objective optimization models are formulated for COTS selection for a software system under consensus recovery block scheme. The models are formulated with the objective of maximization of system reliability and minimization of cost under the constraints on mandatory redundancy for critical modules and component selection. Section 5.2 focuses on COTS selection with the objectives of minimization of the execution time of the overall system and maximization of software reliability with the constraints on mandatory redundancy for critical modules and component selection. In Section 5.3 fuzzy multi-objective optimization models are formulated for component selection for a software system under consensus recovery block scheme incorporating build-or-buy strategy. The models are formulated with the objective of maximization of system reliability and minimization of cost under the constraints on mandatory redundancy for critical modules, delivery time and component selection. A very common problem associated with the use of COTS component is that some of the alternatives available for one module may not be compatible with some alternatives of another module due to issues of implementation, interfacing, etc. Therefore, the issue of compatibility amongst the alternatives of the modules is also discussed in both the sections.

Following are the generalized notations and assumptions which are required for model formulation of the sections.
Chapter 5: Fuzzy Multi-objective Approach to Component Selection for COTS based Modular Software System Incorporating Mandatory Redundancy for Critical Modules

Notations

\( R \quad \text{System quality measure} \)
\( C \quad \text{Overall system cost} \)
\( T \quad \text{Overall deviational execution time} \)
\( f_l \quad \text{Frequency of use, of function } l \)
\( s_l \quad \text{Set of modules required for function } l \)
\( R_i \quad \text{Reliability of module } i \)
\( L \quad \text{Number of functions, the software is required to perform} \)
\( n \quad \text{Number of modules in the software} \)
\( m_i \quad \text{Number of alternatives available for module } i \)
\( V_{ij} \quad \text{Number of versions available for alternative } j \text{ of module } i \)
\( c_{ijk} \quad \text{Cost of version } k \text{ of alternative } j \text{ of module } i \)
\( t_1 \quad \text{Probability that next alternative is not invoked upon failure of the current alternative} \)
\( t_2 \quad \text{Probability that the correct result is judged wrong} \)
\( t_3 \quad \text{Probability that an incorrect result is accepted as correct} \)
\( Y_{ij} \quad \text{Event that correct result of alternative } j \text{ of module } i \text{ is accepted} \)
\( X_{ij} \quad \text{Event that output of alternative } j \text{ of module } i \text{ is rejected} \)
\( r_{ij} \quad \text{Reliability of alternative } j \text{ of module } i \)
\( r_{ijk} \quad \text{Reliability of version } k \text{ of alternative } j \text{ of module } i \)
\( t_i \quad \text{Average execution time of module } i \)
\( t_{ijk} \quad \text{Actual execution time of version } k \text{ of alternative } j \text{ of module } i \)
\( R_0 \quad \text{Minimum reliability level (aspiration) required to attain for system reliability} \)
\( T_0 \quad \text{Maximum deviational time (aspiration level) required for system execution} \)
\( C_0 \quad \text{Maximum budget (aspiration level) required for component selection in CBSE} \)
\( x_{ijk} \quad \begin{cases} 1, & \text{if version } k \text{ of alternative } j \text{ of module } i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \)
\( z_{ij} \quad \begin{cases} 1, & \text{if alternative } j \text{ is present in module } i \\ 0, & \text{otherwise} \end{cases} \)
Assumptions

The following assumptions are applicable to all the three sections.

1. There is a specified budget for the development of software system.
2. A software system consists of a finite number of modules.
3. A software system is required to perform a known number of functions. The program written for a function can call a series of modules \( \leq n \). A failure occurs if a module fails to carry out an intended operation.
4. Codes written for integration of modules do not contain any bug.
5. Several alternatives are available for each module. Fault tolerant architecture is desired in the modules (it has to be within the specified budget). Independently developed alternatives (primarily COTS components) are attached in the modules and work similar to the consensus recovery block scheme discussed in [Kumar, (1998); Berman and Kumar, (1999)].
6. The cost of an alternative is the buying price for the COTS product. Reliability for all the components are known and no separate testing is done.
7. Different versions with respect to cost and reliability of a module are available.
8. Other than available cost-reliability versions of an alternative, we assume the existence of a virtual versions, which has a negligible reliability of 0.001 and zero cost. These components are denoted by index one in the third subscript of \( x_{ijk}, c_{ijk} \) and \( r_{ijk} \). for example \( r_{j1} \) denotes the reliability of first version of alternative \( j \) for module \( i \), having the above property.

Structure Description of Component Based Software System

A critical software system can be developed using CBSE approach. In CBSE approach functional/ customer requirements are first defined. The number and nature of software modules are then determined. During this phase software modules (critical and non-critical) should be identified during the development process and each module must contain at least one alternative. Each alternative then has several
versions (referred as COTS components). Figure 1.1.13 [For details, refer Chapter 1 (Section 1.1.10)] shows how a fault tolerant software system is developed using a component based software engineering approach by assembling software components. This structure of software is applicable to Section 5.1 and Section 5.2. Section 5.3 is based on the structure of the software discussed in figure 1.1.12 [For details, refer Chapter 1 (Section 1.1.10)]. The objective in all the sections of this chapter is to develop optimization models for fault tolerant software systems by creating mandatory redundancy for critical modules.

5.1 FUZZY MULTI-OPTIMIZATION “COST-RELIABILITY” MODEL FOR COTS SELECTION WITH MANDATORY REDUNDANCY FOR CRITICAL MODULES

Component Based Software Engineering (CBSE) is concerned with designing selecting and composing components [Crnkovic and Larsson, (2002)]. As the popularity of this approach and hence number of commercially available software components (COTS) grows, selecting a set of components to satisfy a set of requirements while maximizing reliability and minimizing the cost is becoming more difficult. Fault tolerant software should be used in safety critical systems. Examples of such systems are aircraft that must be in operations throughout the duration of the flight, telecommunication systems, critical command and control systems, business such as banking, retail, manufacturing systems etc. Design diversity techniques certainly increase the overall reliability of the software system by adding redundant components in the system. Therefore, multi-optimization models for selection of COTS components are formulated in this section. The issue of creating mandatory redundancy for critical modules in designing a fault tolerant software system is discussed in this section.

5.1.1 Model Formulation

Let \( S \) be a software architecture made of \( n \) modules, with a maximum number of \( m_i \), alternatives available for each module and each COTS alternative has different
versions. The constraints discussed in Chapter 3 [For details, refer Chapter 3 (Section 3.1.1)] are also applicable to the model formulated in this section.

1. Reliability Equation of COTS Components
2. Version Selection for an Alternative
3. Redundancy Constraint

5.1.2 Objective Function

Proposed model is a fuzzy multi-objective optimization model which aims at maximizing system reliability by simultaneously minimizing the cost. The two objectives are exactly the same as discussed in Chapter 3 [For details, refer Chapter 3 (Section 3.1.2)].

5.1.3 Problem Description

Reliability maximization and cost minimization are the two important goals in designing a fault tolerant modular software system. The developer aims at providing a reliable software system at an economic price thereby satisfying the user’s requirements as well. It is necessary to carry a trade-off between cost and reliability. Hence we consider a component selection problem with dual objectives of maximizing overall system reliability by simultaneously minimizing the cost. Fuzzy cost and reliability functions, soft and ambiguous statements by the COTS vendors make it necessary to define the COTS selection problem under fuzzy environment. Different optimization models are developed to have better understanding of the problem. The problem considered here can now be stated and discussed in the following sections.

5.1.3.1 Fuzzy Multi-optimization “Cost-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

In the optimization model it is assumed that the alternatives of a module are in a Consensus Recovery Block (CRB). Detailed explanation of CRB scheme is given in Chapter 2 [For details, refer Chapter 1 (Section 1.1.7.3)]. This is a hybrid technique of
fault-tolerance and incorporates two schemes-recovery block and N-version programming. Two fuzzifier objective functions of the optimization problem for component selection can be formulated as shown below in problem (5.1.P1)

**Problem (5.1.P1)**

Maximize  \( R = \sum_{l=1}^{L} f_l \prod_{i \in q_l} R_i \)

Minimize  \( C = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{V} c_{ijk} x_{ijk} \)

**Subject to**

\( X \in S = \{ X_{ijk} \text{ is binary variable} \} \)

\[ R_i = 1 - \sum_{j=1}^{m} \left[ \frac{1}{(1-t_j)^{z_{ij}}} \prod_{k=1}^{V} (1-r_{ijk})^{z_{ijk}} \right] + \frac{m}{\prod_{j=1}^{m} (1-t_j)^{z_{ij}}} \prod_{k=1}^{V} \left[ \sum_{j=1}^{m} (1-t_j) \right] \]

\[ P(X_i) = (1-t_i) \left[ (1-r_{ij}) (1-t_j) + r_{ij} t_j \right] \]

\[ P(Y_{ij}) = r_{ij} (1-t_j) \]

\[ r_{ij} = \sum_{k=1}^{V} x_{ijk} r_{ijk} ; j = 1,2,......,m_i ; i = 1,2,......,n \]

\[ \sum_{k=1}^{V} x_{ijk} = 1 ; j = 1,2,......,m_i ; i = 1,2,......,n \]

\[ x_{ij1} + z_{ij} = 1 ; j = 1,2,......,m_i \]

\[ \sum_{j=1}^{m} z_{ij} \geq 1 ; i = 1,2,......,n \]

where X is a vector of component \( x_{ijk} \) and \( z_{ij} \); \( i = 1,......,n ; j = 1,......,m_j ; k = 1,......,V_{ij} \)

Here the first objective stated is the reliability objective and the second one is the cost objective. Constraint (5.1.1) estimates the reliability of module i for a system under consensus recovery block scheme. Constraint (5.1.2) is the probability of event that
output of alternative $j$ of module $i$ is rejected and Constraint (5.1.3) is the probability of event that correct result of alternative $j$ of module $i$ is accepted. Constraint (5.1.4) gives the reliability of alternative $j$ of module $i$. Constraint (5.1.5) ensures that exactly one version is chosen from each alternative of a module. It includes the possibility of choosing a dummy version. Constraints (5.1.6) and (5.1.7) guarantee that not all chosen alternatives of modules are dummies.

5.1.3.2 Fuzzy Multi-optimization “Cost-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

In Section 5.1.3.1 optimization model for optimal selection of components has been discussed in which Constraint (5.1.7) ensures that for a given functionality at least one effective alternative for each of the module is selected. In practice, the software development is a complex process and work with many budget constraints. After the requirements are received, the analysts will perform a thorough analysis and a blue print for the system development is prepared. In this process the entire requirements are categorized and grouped according to the domain functionality. The functionality, which needs to be executed for most of the requirements, is identified and grouped as separate components. There can be some technical components which can be reused in all modules. These technical components such as database handling are common for all modules and can be purchased from the COTS market e.g. Database tools. There can be some functional requirements which are common for all modules. For example in a banking application, the user authorization is required for performing various operations. Each operation or requirement is different but they will share one common thing that is user authorization. So the user authorization is a module which can be developed as a generic module and can be used. Generic modules are normally used by most of the modules in the system based on the need to that module. Similarly if user wants to pay a bill through the bank, after the user authorization the user will go to specific functionality and this is specifically built to handle that functionality. So this can become a separate module which will take care of a specific function.
In the above example a common module like user authorization is very critical and without this the user cannot proceed to do any operation using the application. Similarly bill-payment functionality is less critical because even if this is not working, the user still can use some other activities using the application. So it is necessary that modules are categorized and identified based on their reusability and criticality to run the application. In order to achieve this, a new constraint on criticality of modules can be used to achieve the effective redundancy for all critical modules and at least one effective alternative for non critical modules.

In this section a new model which extends the model discussed in the Section 5.1.3.1 is discussed and formulated.

Let \( I \in \{1, 2, \ldots, n\} \) is the set of all modules

\( I_c \) is set of critical modules and is a subset of \( I \)

The following constraint is needed to select the module which belongs to the critical set \( I_c \)

\[
\sum_{j=1}^{m} z_{ij} \geq 2 ; \quad i \in I_c
\]  

...(5.1.8)

For all other modules which are not critical at least one module is selected. The following constraint takes care of this condition.

\[
\sum_{j=1}^{m} z_{ij} \geq 1 ; \quad i \in \{I - I_c\}
\]  

...(5.1.9)

Equation (5.1.8) ensures that at least minimum of two alternatives for all modules which are critical are selected and equation (5.1.9) will ensure a minimum of the one module for all non critical modules is selected.

The minimum reliability requirement is given by

\[
R_i \geq R_i^0 , i \in I_c
\]  

...(5.1.10)
R_0^0$ is the minimum desirable reliability for all critical modules. For reliability requirement for all other modules is taken care of by the general system reliability requirement. Therefore, the optimization model for COTS selection with mandatory redundancy for critical modules can be formulated as

**Problem (5.1.P2)**

Maximize $R = \sum_{j=1}^{L} f_j \prod_{i \in I} R_i$

Minimize $C = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{V} c_{ijk} x_{ijk}$

Subject to

$X \in S$

$\sum_{j=1}^{m_i} z_{ij} \geq 2 \quad ; \quad i \in I_c$

$\sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \quad i \in \{ I - I_c \}$

$R_i \geq R_i^0 \quad ; \quad i \in I_c$

5.1.3.3 Fuzzy Multi-optimization “Cost-Reliability” Model with Constraints on Mandatory Redundancy and Compatibility

Technological advances have increased both the complexity of system functionalities implemented by software and the information exchanged among collaborative subsystems or software components [Mattiello et al. (2011)]. However, the growing complexity of functions performed by large scale software system has increased the number of accidents caused by software faults in critical systems. Moreover, due to the highly competitive market, commercially-off-the-shelf software products are usually developed by different suppliers. Often these COTS product of one module must be compatible with the COTS product of another module to provide collaborative service. Therefore, the issue of compatibility amongst the alternatives of
the modules must be taken care by the software development team during the design phase of software development. This aims at verifying that all the compatible components must be assembled to form a fault tolerant software system and hence prevents system from failure. Compatibility constraints can be formulated in a similar way as discussed in Chapter 3 [For details, refer Chapter 3 (Section 3.1.3.2)]

Therefore, problem (5.1.P2) can be transformed to another optimization problem using compatibility constraint as

**Problem (5.1.P3)**

\[
\begin{align*}
\text{Maximize} & \quad R = \sum_{i=1}^{I} \prod_{j} R_i \\
\text{Minimize} & \quad C = \sum_{i=1}^{n} \sum_{j=1}^{m_i} V_{ij} c_{jk} x_{jk} \\
\text{Subject to} & \quad X \in S \\
& \quad \sum_{j=1}^{m_i} z_{ij} \geq 2 ; \quad i \in I_c \\
& \quad \sum_{j=1}^{m_i} z_{ij} \geq 1 ; \quad i \in \{ I - I_c \} \\
& \quad R_i \geq R_i^0 ; \quad i \in I_c \\
& \quad x_{ij} - x_{w} \leq M \sum_{t} Y_{t} ; \quad q = 2, \ldots, V_{ij} ; \quad c = 2, \ldots, V_{ij} ; \quad s = 1, \ldots, m_i \\
& \quad \sum_{t} Y_{t} \leq z(v_{ji} - 2) 
\end{align*}
\]

Crisp optimization techniques cannot be applied directly to solve the problems [(5.1.P1), (5.1.P2) and (5.1.P3)] since these methods provide no well defined mechanism to handle the uncertainties quantitatively. Hence we use fuzzy optimization approach to solve the problem.
5.1.4 Solution Procedure

Following algorithm specifies the sequential steps required to solve the fuzzy mathematical programming problems discussed in Section 5.1.3.

5.1.4.1 Solution Procedure of Fuzzy Multi-optimization “Cost-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

Step 1: Reliability $R_0$ and cost $C_0$ are considered to be Triangular Fuzzy Numbers (TFN). They are represented as $A = (a_l, a_m, a_u)$. The crisp equivalents of the fuzzy parameters are computed using a defuzzification function (ranking of fuzzy numbers).

We use the defuzzification function of the type $F_2(A) = \frac{a_l + 2a_m + a_u}{4}$ to defuzzify the aspiration levels.

Step 2: By incorporating the objective function of the fuzzifier min (max) as a fuzzy constraint with a restriction (aspiration) level. The problem (5.1.P1) can be rewritten as

Problem (5.1.P4)

Find $X$

Subject to

$R(X) = \sum_{i=1}^{L} f_i \prod_{i=1}^{n} R_{i} \geq R_0$

$C(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{V_{mn}} c_{ijk} x_{ijk} \leq C_0$

$X \in S$

Step 3: Define appropriate membership functions for each fuzzy inequalities and constraint corresponding to the objective function. The membership function for the fuzzy less than or equal to and greater than or equal to type are given as

$$\mu_\theta(X) = \begin{cases} 1 & ; R(X) \geq R'_0 \\ \frac{R(X) - R'_0}{R_0 - R'_0} & ; R'_0 \leq R(X) < R_0 \\ 0 & ; R(X) < R'_0 \end{cases}$$
where $r_0$ is the aspiration level and $r_0^*$ is the tolerance levels to the fuzzy reliability objective function constraint.

$$
\mu_r(X) = \begin{cases} 
1 & ; C(X) \leq C_0 \\
\frac{C_0^* - C(X)}{C_0^* - C_0} & ; C_0 \leq C(X) < C_0^* \\
0 & ; C(X) > C_0^* 
\end{cases}
$$

where $C_0$ is the restriction and $C_0^*$ is the tolerance levels to the fuzzy budget constraint.

**Step 4:** Now we formulate the crisp optimization problem (5.1.P5) to identify the fuzzy decision and solve the fuzzy system of inequalities corresponding to previous problem:

**Problem (5.1.P5)**

Maximize $\alpha$

Subject to

$$
\mu_r(X) \geq \alpha \\
\mu_c(X) \geq \alpha \\
X \in S
$$

**Step 5:** The constraints corresponding to reliability, and cost in the problem (5.1.P5) are very important constraints which may be assigned different weights as per their relative importance, other constraints must be satisfied as given in $S$. By doing so we can rewrite the problem as

**Problem (5.1.P6)**

Maximize $\alpha$

Subject to

$$
\mu_r(X) \geq w_1 \alpha \\
\mu_c(X) \geq w_2 \alpha \\
X \in S
$$

**Step 6:** On substituting the values for $\mu_r(X)$ and $\mu_c(X)$ the problem can be converted to

**Problem (5.1.P7)**

Maximize $\alpha$
Subject to

\[
R(X) \geq R_0^r + (1-w_1\alpha)(R_0 - R_0^r)
\]
\[
C(X) \geq C_0^r - (1-\alpha)(C_0^r - C_0)
\]

\[X \in S\]
\[\alpha \in [0,1]\]
\[
\mu_r(X) = w_1\alpha
\]
\[
\mu_c(X) = w_2\alpha
\]
\[w_1, w_2 \geq 0\]
\[w_1 + w_2 = 1\]

Step 7: If a feasible solution is not obtained for the problem (5.1.P7) or (5.1.P6) then we can use fuzzy goal programming approach to obtain a compromised solution [Mohamed, (1997)].

5.1.4.2 Solution Procedure of Fuzzy Multi-optimization “Cost-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

Same algorithm (For details, refer Section 5.1.4.1) is used to solve fuzzy mathematical programming problems, is applicable to the optimization model formulated in Section (5.1.3.2) or problem (5.1.P2). Therefore, the problem can be restated as

Problem (5.1.P8)

Maximize \(\alpha\)

Subject to

\[
R(X) \geq R_0^r + (1-w_1\alpha)(R_0 - R_0^r)
\]
\[
C(X) \geq C_0^r - (1-\alpha)(C_0^r - C_0)
\]

\[X \in S\]

\[
\sum_{j=1}^{m} z_{ij} \geq 2 \quad ; \quad i \in I_c
\]

\[
\sum_{j=1}^{m} z_{ij} \geq 1 \quad ; \quad i \in \{I - I_c\}
\]
\[ R_i \geq R_i^0 \quad ; \ i \in I_C \]
\[ \alpha \in [0,1] \]
\[ \mu_x(X) = w_x\alpha \]
\[ \mu_t(X) = w_t\alpha \]
\[ w_1, \ w_2 \geq 0 \]
\[ w_1 + w_2 = 1 \]

### 5.1.4.3 Solution Procedure of Fuzzy Multi-optimization “Cost-Reliability” Model with Constraints on Mandatory Redundancy and Compatibility

Adding constraint of compatibility to problem (5.1.P8), it can be transformed to (5.1.P9) and can be re-written as

**Problem (5.1.P9)**

Maximize \( \alpha \)

Subject to

\[ R(X) \geq R^0 + (1 - w_x\alpha)(R^0 - R^0) \]
\[ C(X) \geq C^0 - (1 - w_t\alpha)(C^0 - C^0) \]
\[ X \in S \]
\[ \sum_{j=1}^{m_i} z_{ij} \geq 2 \quad ; \ i \in I_C \]
\[ \sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \ i \in \{1 - I_C\} \]
\[ R_i \geq R_i^0 \quad ; \ i \in I_C \]
\[ x_{pq} - x_{nc} \leq M \sum y_i \quad ; \ q = 2, \ldots, V_{gs}, \ c = 2, \ldots, V_{hn_i}, \ s = 1, \ldots, m_g \]
\[ \sum y_i \leq z(V_{hn_i} - 2) \]
\[ \alpha \in [0,1] \]
\[ \mu_x(X) = w_x\alpha \]
\[ \mu_t(X) = w_t\alpha \]
\[ \mu_c(X) = \alpha \]
\[ w_1, \ w_2 \geq 0 \]
\[ w_1 + w_2 = 1 \]
5.1.5 Numerical Illustration

A numerical example is illustrated to describe the proposed methodology of fuzzy multi-objective “cost-reliability” optimization model for COTS selection, with mandatory redundancy for critical modules. In the example the software system is decomposed into four modules \((m_1), (m_2), (m_3)\) and \((m_4)\). Three alternatives are available for module \((m_1)\), four alternatives for module \((m_2)\), two alternatives for module \((m_3)\) and four alternatives are available for module \((m_4)\). Each alternative has three versions \(v_1, v_2, v_3\). The cost reliability data for different versions are given in table 5.1.1. The cost of first version which is the virtual versions for all alternatives is zero and reliability is 0.001. This is done so, as if in the optimal solution for some module \(x_{yi} = 1\) that implies corresponding alternative is not to be attached in the module.

Let the software is required to perform four functions, so \(L = 4\). The set of modules required for the four functions are given by \(S_1 = \{1,2,3,4\}, S_2 = \{1,3,4\}, S_3 = \{2,4\}, S_4 = \{2,3,4\}\). The frequency of use is given by \(f_1 = 0.25, f_2 = 0.30, f_3 = 0.25\) and \(f_4 = 0.20\). It is also assumed that \(t_1 = 0.01, t_2 = 0.05\) and \(t_3 = 0.01\).

Table 5.1.1: Data Set of Cost and Reliability for Fuzzy Multi-Optimization “Cost-Reliability” Model with Mandatory Redundancy for Critical Modules

<table>
<thead>
<tr>
<th>Modules</th>
<th>Alternatives</th>
<th>Versions</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cost</td>
<td>Reliability</td>
<td>Cost</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.001</td>
<td>8.4</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0.001</td>
<td>8.5</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0.001</td>
<td>7.0</td>
<td>0.89</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0.001</td>
<td>3.3</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0.001</td>
<td>5.4</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0.001</td>
<td>10.2</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>0.001</td>
<td>14.7</td>
<td>0.97</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.001</td>
<td>9.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0.001</td>
<td>10.8</td>
<td>0.94</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0.001</td>
<td>8.6</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0.001</td>
<td>7.8</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0.001</td>
<td>8.4</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>0.001</td>
<td>14.5</td>
<td>0.97</td>
</tr>
</tbody>
</table>
5.1.5.1 Solution of Fuzzy Multi-optimization “Cost-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

Aspiration & Tolerance Level of Reliability and Cost

The TFN corresponding to the reliability and cost aspirations are tabulated in table 5.1.2. The values of these fuzzy parameters are assumed to be specified by the management based on past experiences and/or expert opinion. Defuzzified values of these parameters are computed using the defuzzification function

$$F_2(A) = \left( a_l + 2a_m + a_u \right) \frac{1}{4}$$

and are given in table 5.1.2 along with their tolerance level.

<table>
<thead>
<tr>
<th>Fuzzy Parameter (A)</th>
<th>$a_l$</th>
<th>$a_m$</th>
<th>$a_u$</th>
<th>Defuzzified Value ($F_2(A)$) (Aspiration level)</th>
<th>Tolerance Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.87</td>
<td>0.93</td>
<td>0.95</td>
<td>0.92</td>
<td>0.87</td>
</tr>
<tr>
<td>C</td>
<td>46</td>
<td>51</td>
<td>56</td>
<td>51</td>
<td>56</td>
</tr>
</tbody>
</table>

Membership Functions

Membership functions for reliability, and cost can be written as

$$\mu_R(X) = \begin{cases} 
1 & \text{; } R(X) \geq 0.92 \\
\frac{1}{0.92 - 0.87} & \text{; } 0.87 \leq R(X) < 0.92 \\
0 & \text{; } R(X) < 0.87 
\end{cases}$$

$$\mu_C(X) = \begin{cases} 
1 & \text{; } C(X) \leq 51 \\
\frac{56 - \sum_i \sum_j \sum_k c_{ijk} x_{ijk}}{56 - 51} & \text{; } 51 \leq C(X) < 56 \\
0 & \text{; } C(X) > 56 
\end{cases}$$
Assignment of Weights

The assignment of weights is based on the expert’s judgment for the reliability and deviational execution time, i.e. \( w = (w_1, w_2) = (0.6, 0.4) \).

Fuzzy Goal Programming Approach

On solving the problem, we found that the problem (5.1.P7) is not feasible. The management’s goal cannot be achieved for a feasible value of \( \alpha \in [0,1] \). Therefore, we have to use fuzzy goal programming technique to obtain a compromised solution. This approach is based on the goal programming technique for solving crisp goal programming problems [Mohamed, (1997)]. The maximum value of any membership function can be 1. Maximization of \( \alpha \in [0,1] \) is equivalent to making it as close to 1 as best as possible. This can be achieved by minimizing the negative deviational variables of goal programming (i.e. \( \eta \)) from 1. The fuzzy goal programming formulation for the problem (5.1.P7) introducing the negative and positive deviational variables \( \eta_j \) and \( \rho_j \) is given as

**Problem (5.1.P10)**

\[ \text{Minimize } u \]

Subject to

\[
\begin{align*}
\mu_R(X) + \eta_1 - \rho_1 &= 1 \\
\mu_C(X) + \eta - \rho &= 1 \\
u &\geq w_j * \eta_j \\
\eta_j * \rho_j &= 0 \quad ; \eta_j, \rho_j \geq 0 \quad ; j = 1, 2 \\
X &\in S \\
\alpha &\in [0,1] \\
w_1, w_2 &\geq 0 \\
w_1 + w_2 &= 1 \\
\alpha &= 1 - u
\end{align*}
\]

The optimal solution set so obtained for problem (5.1.P10) are optimal for (5.1.P1). Solving problem (5.1.P10) for optimization problem (5.1.P1) with the data set data set given in tables 5.1.1 and 5.1.2, following solution is obtained.
Table 5.1.3: Solution of Fuzzy Multi-optimization “Cost-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

<table>
<thead>
<tr>
<th>Modules</th>
<th>Selected COTS Components</th>
<th>System Reliability</th>
<th>System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x_{111} = x_{122} = x_{131} = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$x_{211} = x_{222} = x_{232} = x_{241} = 1$</td>
<td>0.89</td>
<td>51</td>
</tr>
<tr>
<td>3</td>
<td>$x_{313} = x_{321} = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$x_{411} = x_{423} = x_{433} = x_{441} = 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Modules ($m_1$) and ($m_3$) are critical modules. It can be clearly seen from the table that redundancy is not allowed in both the modules. Single component (i.e. $x_{122}$) is selected for module ($m_1$) and for module ($m_3$) only $x_{313}$ is selected. Two components ($x_{222}$ and $x_{232}$) and ($x_{423}$ and $x_{433}$) are selected for module ($m_2$) and ($m_4$) respectively. Therefore, it can be concluded that redundancy is allowed in these two modules. The diagrammatic representation of the solution can be shown in figure 5.1.1.

Figure 5.1.1: Diagrammatic Representation for Solution of Fuzzy Multi-optimization “Cost-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility
5.1.5.2 Solution Procedure of Fuzzy Multi-optimization “Cost-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

We are discussing the component selection problem for a critical software. Such software is built by first identifying the critical modules and then creating mandatory redundancy in them. This is done so because while execution if one alternative of a module fails then other one will take over. For such kind of system very high reliability is desired which in turn increase the overall cost of the system as we are now selecting at least two alternatives for the critical modules. Therefore, the aspiration and tolerance level for the optimization model discussed in this section are different from the model discussed in Section 5.1.5.1.

Aspiration & Tolerance Level of Reliability and Cost

The TFN corresponding to the reliability and cost aspirations are tabulated in table 5.1.4. The values of these fuzzy parameters are assumed to be specified by the management based on past experiences and/or expert opinion. Defuzzified values of these parameters are computed using the defuzzification function $F_2(A) = \frac{(a_l + 2a_m + a_u)}{4}$ and are given in table 5.1.4 along with their tolerance level.

<table>
<thead>
<tr>
<th>Fuzzy Parameter (A)</th>
<th>Defuzzified Value (F_2(A)) (Aspiration level)</th>
<th>Tolerance level</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.993</td>
<td>0.96</td>
</tr>
<tr>
<td>C</td>
<td>71</td>
<td>85</td>
</tr>
</tbody>
</table>

Membership Functions

Membership functions for reliability, and cost can be written as

$$
\mu_r(X) = \begin{cases} 
1 & ; R(X) \geq 0.993 \\
\frac{\sum_{i=1}^n \prod_{j=1}^m R_i - 0.96}{0.993 - 0.96} & ; 0.96 \leq R(X) < 0.993 \\
0 & ; R(X) < 0.96
\end{cases}
$$
Chapter 5: Fuzzy Multi-objective Approach to Component Selection for COTS based Modular Software System Incorporating Mandatory Redundancy for Critical Modules

Assignment of Weights

The assignment of weights is based on the expert’s judgment for the reliability and execution deviational time, i.e. \( w = (w_1, w_2) = (0.6, 0.4) \).

Fuzzy Goal Programming Approach

A feasible solution is not obtained for problem (5.1.P8). It is also solved using fuzzy goal programming approach and can be restated as

Problem (5.1.P11)

Minimize \( u \)

Subject to

\[ \mu_R(X) + \eta_1 - \rho_1 = 1 \]
\[ \mu_C(X) + \eta - \rho = 1 \]
\[ u \geq w_j * \eta_j \]
\[ \eta_j * \rho_j = 0 \quad ; \quad \eta_j, \rho_j \geq 0 \quad ; \quad j = 1, 2 \]
\[ X \in S \]
\[ \sum_{j=1}^{m_j} z_{ij} \geq 2 \quad ; \quad i \in I_c \]
\[ \sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \quad i \in \{ I - I_c \} \]
\[ R_i \geq R_i^0 \quad ; \quad i \in I_c \]
\[ \alpha \in [0, 1] \]
\[ w_1, w_2 \geq 0 \]
\[ w_1 + w_2 = 1 \]
\[ \alpha = 1 - u \]
Module \((m_1)\) and \((m_3)\) are identified as critical modules. So a constraint on criticality of module is added to the model of the previous solution [For details, refer Chapter 5 (Section 5.1.2.1)] to achieve redundancy for all critical modules, that is, for the first and third module we want at least two alternatives should be selected.

The optimal solution set so obtained for problem (5.1.P11) are optimal for (5.1.P2). Solving problem (5.1.P11) for optimization model discussed in section (5.1.3.2) and the above data set for cost and reliability, following solution is obtained.

Table 5.1.5: Solution of Fuzzy Multi-optimization “Cost-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

<table>
<thead>
<tr>
<th>Modules</th>
<th>Selected COTS Components</th>
<th>System Reliability</th>
<th>System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(x_{111} = x_{122} = x_{132} = 1)</td>
<td>0.993</td>
<td>72.60</td>
</tr>
<tr>
<td>2</td>
<td>(x_{211} = x_{222} = x_{231} = x_{243} = 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(x_{313} = x_{323} = 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(x_{411} = x_{423} = x_{433} = x_{441} = 1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The diagrammatic representation of the solution can be shown as

![Diagrammatic Representation of Solution](image-url)
By looking at the solution from table 5.1.5 as well as from the figure 5.1.2, we can see that redundancy is now allowed in the critical modules, viz., (m<sub>1</sub>) and (m<sub>3</sub>).

### 5.1.5.3 Solution of Fuzzy Multi-optimization “Cost-Reliability” Model with Mandatory Redundancy and Compatibility Constraints

The aspiration and tolerance level of reliability, membership functions and assignment of weights are exactly the same as discussed in the above section (For details, refer Section 5.1.5.2). The problem is solved by goal programming approach.

By adding the constraints on compatibility to problem (5.1.P11), a new problem can be written as

**Problem (5.1.P12)**

*Minimize*  

Subject to

\[
\mu_r(X) + \eta_1 - \rho_1 = 1
\]

\[
\mu_c(X) + \eta - \rho = 1
\]

\[u \geq w_j \eta_j\]

\[\eta_j \rho_j = 0 \quad ; \quad \eta_j, \rho_j \geq 0 \quad ; \quad j = 1, 2\]

\[X \in S\]

\[\sum_{j=1}^{m_j} z_{ij} \geq 2 \quad ; \quad i \in I_c\]

\[\sum_{j=1}^{m_j} z_{ij} \geq 1 \quad ; \quad i \in \{ I - I_c \}\]

\[R_i \geq R_i^0 \quad ; \quad i \in I_c\]

\[x_{gq} - x_{mv} \leq M \sum y_i \quad ; \quad q = 2, \ldots, V_{gq} \quad ; \quad c = 2, \ldots, V_{hv} \quad ; \quad s = 1, \ldots, m_g\]

\[\sum y_i = z(V_{hv} - 2)\]

\[\alpha \in [0, 1]\]

\[w_1, w_2 \geq 0\]

\[w_1 + w_2 = 1\]

\[\alpha = 1 - u\]
We assume that the third alternative \( x_{132} \) of first module is compatible with the second \( x_{223} \) and fourth \( x_{242} \) alternatives of the second module. Solving problem (5.1.P12) for Problem (5.1.P3) [For details, refer Section (5.1.3.3)], the following result is obtained.

**Table 5.1.6: Solution of Fuzzy Multi-optimization “Cost-Reliability” Model with Mandatory Redundancy and Compatibility Constraints**

<table>
<thead>
<tr>
<th>Modules</th>
<th>Selected COTS Components</th>
<th>System Reliability</th>
<th>System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x_{111} = x_{122} = x_{132} = 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( x_{211} = x_{221} = x_{232} = x_{242} = 1 )</td>
<td>0.987</td>
<td>83.1</td>
</tr>
<tr>
<td>3</td>
<td>( x_{313} = x_{323} = 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( x_{413} = x_{423} = x_{431} = x_{443} = 1 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is observed that due to the compatibility condition, the fourth alternative of the second module \( x_{242} \) is chosen as it is compatible with the third alternative of the first module \( x_{132} \).

**Figure 5.1.3: Diagrammatic Representation of Solution for Fuzzy Multi-optimization “Cost-Reliability” Model with Mandatory Redundancy and Compatibility Constraints**
5.2 FUZZY MULTI-OPTIMIZATION “EXECUTION TIME-RELIABILITY” MODEL FOR COTS SELECTION WITH MANDATORY REDUNDANCY FOR CRITICAL MODULES

The execution times for all the functions called in the system are not necessarily identical. The execution time may vary because of the variations in the number of alternatives present in each module that are being called by the functions and also the nature of task to be perform by the components of the modules. On invocation of the function, the module is called. All the alternatives of that module get executed simultaneously. Execution times of all the alternatives are different. Alternatives whose execution complete early, will have to wait for their corresponding alternatives to complete their job. Therefore, our objective here is to minimize the waiting time.

To resolve this problem, the deviational time has been introduced. An average time for the execution of an alternative has been assumed, and to solve the problem the deviation from the average time has been calculated. The deviation from the average time is used to address two issues [For details, refer Chapter 2 (figure 2.2.1)]. Firstly, to find out those alternatives whose execution completes early before the assumed average time. And secondly, to find those alternatives whose execution takes longer time to complete. In other words, the execution completes after the assumed average time. Therefore, in both the cases there is a deviation from average time. So the optimization model in this section aims at optimal COTS selection for a software system by minimizing the deviational time by simultaneously maximizing the system reliability.

5.2.1 Additional Assumptions

In addition to the assumptions discussed in the introduction of this chapter, following assumptions are also applicable to the model formulation in this section.

9. Execution times of all the alternatives are different. Those alternatives whose execution complete early, will have to wait for their corresponding alternatives to finish up their job.

10. Deviational time is the difference between the average time and the actual execution time of the software.
11. Different versions with respect to time, reliability and cost of modules are available.
12. The execution time of the virtual version is zero.

5.2.2 Model Formulation

Let $S$ be a software architecture made of $n$ modules, with a maximum number of $m_i$ alternatives available for each module and each COTS alternative has different versions. The constraints which are applicable to the model are given below [For details, refer Chapter 3 (Section 3.2.2)]

1. Reliability Equation of COTS Components
2. Version Selection for an Alternative
3. Redundancy Constraint
4. Budget Constraint

5.2.3 Objective Functions

The proposed model is a fuzzy multi-objective optimization model which aims at maximizing system reliability by simultaneously minimizing the cost. The two objectives are exactly the same as discussed in Chapter 3 [For details, refer Chapter 3 (Section 3.2.3)].

5.2.4 Problem Description

The important goals for a software developer for developing a fault tolerant system are maximizing system reliability and minimizing execution time of the overall system under budgetary constraint. The estimate of execution time, reliability and cost of COTS products cannot be done precisely because of ambiguous and uncertain factors. The problem with reliability maximization and execution deviational time minimization objectives subject to budget constraint can be considered as a multiple objective problem of reliability and execution deviational time while solving with the fuzzy optimization. The two objectives can be assigned different weights according to their importance level and the problem can be solved with the weighted min-max approach. Three different optimization problems are formulated in this section to have
better understanding of the problem. Defining a fuzzy multi-objective model with execution deviational time and reliability objectives with budget as one of the constraints can be stated as follows.

5.2.4.1 Fuzzy Multi-optimization “Execution Time-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

The optimization model for COTS selection with multiple objectives of reliability maximization and deviational execution time minimization under fuzzy environment can be written as

**Problem (5.2.P1)**

Maximize \( \tilde{R} = \sum_{i=1}^{L} \prod_{j \in y} R_i \)

Minimize \( \tilde{T} = \sum_{j = 1}^{i} \sum_{x_{ijk}} \sum_{s} \sum_{v_{ijk}} \in_{ijk} x_{ijk} \)

Subject to

\( X \in S = \{ x_{ijk} \text{ is binary variable} / \)

\[ \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{v_{ij}} c_{ijk} x_{ijk} \leq C_0 \]  \( \ldots(5.2.1) \)

\[ R_i = 1 + \left[ \sum_{j=1}^{n} \frac{1}{m} \left[ \prod_{k=1}^{m} (1 - \eta_{ik})^{x_{ijk}} \right] \prod_{j=1}^{m} (1 - \eta_{ij})^{x_{ijk}} \right] + \left[ \sum_{j=1}^{n} \left[ \prod_{k=1}^{m} \tilde{P}(X_{ik})^{x_{ijk}} \right] \tilde{P}(y_{ij})^{x_{ijk}} - 1 \right], i = 1, 2, \ldots, n \]  \( \ldots(5.2.2) \)

\[ P \left( Y_{ij} \right) = (1 - t_1) \left[ (1 - \eta_{ij})(1 - t_3) + \eta_{ij} t_2 \right] \]  \( \ldots(5.2.3) \)

\[ P \left( Y_{ij} \right) = r_y (1 - t_z) \]  \( \ldots(5.2.4) \)

\[ r_{ij} = \sum_{k=1}^{v_{ij}} x_{ijk} r_{jk} : j = 1, 2, \ldots, m_i; i = 1, 2, \ldots, n \]

\[ \sum_{j=1}^{m} x_{ijk} = 1 : j = 1, 2, \ldots, m_i; i = 1, 2, \ldots, n \]

\[ x_{ij1} + z_{ij} = 1 : j = 1, 2, \ldots, m_i \]

\[ \sum_{j=1}^{m} z_{ij} \geq 1 : i = 1, 2, \ldots, n \]  \( \ldots(5.2.8) \)
where $X$ is a vector of component $x_{ijk}$ and $z_{ij}$ ; $i = 1, \ldots, n$ ; $j = 1, \ldots, m_i$ ; $k = 1, \ldots, V_{ij}$

First objective function maximizes the system quality (in terms of reliability) through a weighted function of module reliabilities and second objective minimizes the execution deviational time. Constraint (5.2.1) is a budget constraint. Constraint (5.2.2) estimates the reliability of module $i$. Constraint (5.2.3) is the probability of event that output of alternative $j$ of module $i$ is rejected and Constraint (5.2.4) is the probability of event that correct result of alternative $j$ of module $i$ is accepted. Constraint (5.2.5) gives the reliability of alternative $j$ of module $i$. Constraint (5.2.6) ensures that exactly one version is chosen from each alternative of a module. It includes the possibility of choosing a dummy version. Equations (5.2.7) and (5.2.8) guarantee that not all chosen alternatives of modules are dummies.

### 5.2.4.2 Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

While designing a software, it is necessary that modules are categorized and identified based on their reusability and criticality to run the application. In order to achieve this, a new constraint on criticality of modules can be used to achieve the effective redundancy for all critical modules and at least one effective alternative for non-critical modules. The section is discussed in Section 5.1.3.2.

**Problem (5.2.P2)**

Maximize $\mathcal{R} = \sum_{l=1}^{L} f_l \prod_{i \in S} R_i$

Minimize $\mathcal{M} = \sum_{l=1}^{L} f_l \sum_{i \in S} \sum_{j=1}^{m_i} \sum_{k=1}^{V_{ij}} \in_{ijk} x_{ijk}$

**Subject to**

$X \in S$

$\sum_{j=1}^{m_i} z_{ij} \geq 2$ ; $i \in I_c$  \hspace{1cm} …(5.2.9)

$\sum_{j=1}^{m_i} z_{ij} \geq 1$ ; $i \in \{ I \setminus I_c \}$  \hspace{1cm} …(5.2.10)
5.2.4.3 Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and Compatibility

Multi-objective optimization model with Constraints on Mandatory Redundancy and Compatibility can be written as

**Problem (5.2.P3)**

\[
\begin{align*}
\text{Maximize} & \quad R = \sum_{i=1}^{L} f_i \prod_{s \in S} R_s \\
\text{Minimize} & \quad T = \sum_{i=1}^{L} f_i \sum_{i \in I_i} \sum_{j=1}^{m_i} \sum_{s=1}^{s_i} \in_{o} X_{i,j,s} \\
\text{Subject to} & \quad X \in S \\
& \quad \sum_{j=1}^{m_i} z_{ij} \geq 2 \quad ; \quad i \in I_c \\
& \quad \sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \quad i \in \{I - I_c\} \\
& \quad R_i \geq R_i^0 \quad ; \quad i \in I_c \\
& \quad x_{gs} - x_{huc} \leq M \sum_{s=1}^{m_{gs}} y_s \quad ; \quad q = 2, \ldots, V_{gs} \quad ; \quad c = 2, \ldots, V_{huc} \quad ; \quad s = 1, \ldots, m_g \\
& \quad \sum_{i=1}^{L} f_i \prod_{s \in S} R_s \\
\end{align*}
\]

Crisp optimization techniques cannot be applied directly to solve the problems [(5.2.P1), (5.2.P2) and (5.2.P3)] since these methods provide no well defined mechanism to handle the uncertainties quantitatively. Hence we use fuzzy optimization approach to solve the problem.

**5.2.5 Solution Procedure**

Following algorithm specifies the sequential steps required to solve the fuzzy mathematical programming problems discussed in Section 5.2.4.
5.2.5.1 Solution Procedure of Fuzzy Multi-optimization “Execution Time-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

**Step 1:** Reliability $R_o$, deviational execution time $T_o$, and cost $C_o$ are considered to be Triangular Fuzzy Numbers (TFN). They are represented as $A = (a_l, a_m, a_u)$. The crisp equivalents of the fuzzy parameters are computed using a defuzzification function (ranking of fuzzy numbers). We use the defuzzification function of the type $F_2 (A) = \frac{(a_l + 2a_m + a_u)}{4}$ to defuzzify the aspiration levels.

**Step 2:** By incorporating the objective function of the fuzzifier min (max) as a fuzzy constraint with a restriction (aspiration) level. The problem (5.2.P1) can be rewritten as

**Problem (5.2.P4)**

*Find $X$*

**Subject to**

$$R(X) = \sum_{j=1}^{f} \prod_{i \in j} R_i \geq R_o$$

$$T(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{V_{ik}} e_{ijk} x_{ijk} \leq T_o$$

$$C(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{V_{ik}} c_{ijk} x_{ijk} \leq C_0$$

$X \in S$

**Step 3:** Define appropriate membership functions for each fuzzy inequalities and constraint corresponding to the objective function. The membership function for the fuzzy less than or equal to and greater than or equal to type are given as

$$\mu_\beta(X) = \begin{cases} 1 & ; R(X) \geq R_0 \\ \frac{R(X) - R_0^*}{R_0 - R_0^*} & ; R_0^* \leq R(X) < R_0 \\ 0 & ; R(X) < R_0^* \end{cases}$$

where $R_o$ is the aspiration level and $R_o^*$ is the tolerance levels to the fuzzy reliability objective function constraint..
Chapter 5: Fuzzy Multi-objective Approach to Component Selection for COTS based Modular Software System Incorporating Mandatory Redundancy for Critical Modules

2012

Shivani, Ph.D. Thesis                           Department of Operational Research, University of Delhi

\[
\mu_t(X) = \begin{cases} 
1 & ; \ T(X) \leq T_0 \\
\frac{T'_0 - T(X)}{T'_0 - T_0} & ; \ T_0 \leq T(X) < T'_0 \\
0 & ; \ T(X) > T'_0 
\end{cases}
\]

where \( T_0 \) is the restriction and \( T'_0 \) is the tolerance levels to the fuzzy absolute deviational execution time objective function constraint

\[
\mu_c(X) = \begin{cases} 
1 & ; \ C(X) \leq C_0 \\
\frac{C'_0 - C(X)}{C'_0 - C_0} & ; \ C_0 \leq C(X) < C'_0 \\
0 & ; \ C(X) > C'_0 
\end{cases}
\]

where \( C_0 \) is the restriction and \( C'_0 \) is the tolerance levels to the fuzzy budget constraint.

**Step 4:** Now we formulate the crisp optimization problem (5.2.P5) to identify the fuzzy decision and solve the fuzzy system of inequalities corresponding to previous problem:

**Problem (5.2.P5)**

Maximize \( \alpha \)

Subject to

\[
\mu_r(X) \geq \alpha \\
\mu_e(X) \geq \alpha \\
\mu_c(X) \geq \alpha
\]

\( X \in S \)

**Step 5:** The constraints corresponding to reliability and execution deviational time in the problem (5.2.P5) are very important constraints which may be assigned different weights as per their relative importance. Other constraints must be satisfied as given in \( S \), so the problem becomes
Problem (5.2.P6)

Maximize $\alpha$

Subject to

\[ \mu_{R}(X) \geq w_{1}\alpha \]
\[ \mu_{T}(X) \geq w_{2}\alpha \]
\[ \mu_{C}(X) \geq \alpha \]
\[ X \in S \]

Step 6: On substituting the values for $\mu_{R}(X), \mu_{T}(X)$ and $\mu_{C}(X)$ the problem becomes

Problem (5.2.P7)

Maximize $\alpha$

Subject to

\[ R(X) \geq R_{c} + (1-w_{1}\alpha)(R_{c} - R_{o}) \]
\[ T(X) \geq T_{c} - (1-w_{2}\alpha)(T_{c} - T_{o}) \]
\[ C(X) \geq C_{c} - (1-\alpha)(C_{c} - C_{o}) \]
\[ X \in S \]
\[ \alpha \in [0,1] \]
\[ \mu_{R}(X) = w_{1}\alpha \]
\[ \mu_{T}(X) = w_{2}\alpha \]
\[ \mu_{C}(X) = \alpha \]
\[ w_{1}, w_{2} \geq 0 \]
\[ w_{1} + w_{2} = 1 \]

Step 7: If a feasible solution is not obtained for the problem (5.2.P7) or (5.2.P6) then we can use fuzzy goal programming approach to obtain a compromised solution [Mohamed, (1997)].

5.2.5.2 Solution Procedure of Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

Same algorithm which is used [For details, refer (Section 5.2.5.1)] to solve fuzzy mathematical programming problems is applicable to the optimization model discussed in section (5.2.4.2) or problem (5.2.P2). Therefore, the problem can be re-stated as
Problem (5.2.P8)

Maximize $\alpha$

Subject to

\[
R(X) \geq R_0^* + (1-w_1\alpha)(R_0^* - R_0^*)
\]

\[
T(X) \geq T_0^* - (1-w_2\alpha)(T_0^* - T_0^*)
\]

\[
C(X) \geq C_0^* - (1-\alpha)(C_0^* - C_0^*)
\]

$X \in S$

\[
\sum_{j=1}^{m_i} z_{ij} \geq 2 ; i \in I_c
\]

\[
\sum_{j=1}^{m_i} z_{ij} \geq 1 ; i \in \{ I - I_c \}
\]

\[
R_i \geq R_i^0 ; i \in I_c
\]

$\alpha \in [0,1]$

$\mu_k(X) = w_1\alpha$

$\mu_j(X) = w_2\alpha$

$\mu_c(X) = \alpha$

$w_1, w_2 \geq 0$

$w_1 + w_2 = 1$

5.2.5.3 Solution Procedure of Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and Compatibility

Adding constraint of compatibility to problem (5.1.P8) it can be transformed to (5.1.P9) and can be stated as

Problem (5.2.P9)

Maximize $\alpha$

Subject to

\[
R(X) \geq R_0^* + (1-w_1\alpha)(R_0^* - R_0^*)
\]

\[
T(X) \geq T_0^* - (1-w_2\alpha)(T_0^* - T_0^*)
\]

\[
C(X) \geq C_0^* - (1-\alpha)(C_0^* - C_0^*)
\]
Optimal Component Selection for Fault Tolerant Software Design under Consensus
Recovery Block Scheme

\[ X \in S \]

\[ \sum_{j=1}^{m_i} z_{ij} \geq 2 \quad ; \quad i \in I_c \]

\[ \sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \quad i \in \{ I - I_c \} \]

\[ R_i \geq R_i^0 \quad ; \quad i \in I_c \]

\[ X_{gs} - x_{hs,c} \leq M \sum_{q=2}^{2} y_{q} \quad ; \quad q = 2 \ldots \ldots V_{gs} \quad ; \quad c = 2 \ldots \ldots V_{hs} \quad ; \quad s = 1 \ldots \ldots m_g \]

\[ \sum y_i \leq z(V_{hs,c} - 2) \]

\[ \alpha \in [0,1] \]

\[ \mu_s(X) = w_1 \alpha \]

\[ \mu_r(X) = w_2 \alpha \]

\[ \mu_c(X) = \alpha \]

\[ w_1, w_2 \geq 0 \]

\[ w_1 + w_2 = 1 \]

5.2.6 Numerical Illustration

A numerical example is illustrated to describe the proposed methodology of fuzzy multi-objective execution-time reliability model for COTS selection with mandatory redundancy for critical modules. In the example the software system is decomposed into seven modules \((m_1) - (m_7)\). Three alternatives are available for module \((m_1)\), four alternatives for module \((m_2)\), three alternatives for module \((m_3)\), two alternatives for module \((m_4)\), four alternatives for module \((m_5)\), three alternatives for module \((m_6)\) and lastly, two alternatives are available for module \((m_7)\). Each alternative has three versions \((v_1, v_2, v_3)\). The cost, reliability and execution time data for different versions are given in table 5.2.1. The cost and execution time of first version which is the virtual version for all alternatives is zero and reliability is 0.001. This is done so, as if in the optimal solution for some module \(x_{v1} = 1\) that implies corresponding alternative is not to be attached in the module.
Let the software is required to perform four functions, so $L = 4$. The set of modules required for the four functions are given by $S_1 = \{1,2,3,4,7\}$, $S_2 = \{3,4,5,6\}$, $S_3 = \{1,3,4,5\}$ and $S_4 = \{3,4,7\}$. The frequency of use is given by $f_1 = 0.3$, $f_2 = 0.25$, $f_3 = 0.25$ and $f_4 = 0.2$. It is also assumed that $t_1 = 0.01$, $t_2 = 0.05$ and $t_3 = 0.01$.

**Table 5.2.1: Data Set of Cost, Reliability and Execution Time for Fuzzy Multi-optimization “Execution Time-Reliability” Model with Mandatory Redundancy for Critical Modules**

| Modules | Alternatives | Versions |  |  |  |
|---------|--------------|----------|  |  |  |
|         |              | 1        | 2        | 3        |
|         |              | Cost     | Reliability | Execution Time | Cost   | Reliability | Execution Time | Cost     | Reliability | Execution Time |
| 1       | 1            | 0        | 0.001    | 0.0       | 10.4   | 0.90        | 0.15         | 11.0     | 0.88        | 0.19         |
| 2       | 0            | 0.001    | 0.0       | 8.7       | 0.88    | 0.22        | 9.0          | 0.87     | 0.24        |
| 3       | 0            | 0.001    | 0.0       | 9.2       | 0.89    | 0.07        | 9.0          | 0.87     | 0.13        |
| 4       | 0            | 0.001    | 0.0       | 13.0      | 0.92    | 0.23        | 12.7         | 0.91     | 0.32        |
| 2       | 1            | 0        | 0.001    | 0.0       | 13.6    | 0.94        | 0.19        | 14.2     | 0.93        | 0.10        |
| 3       | 0            | 0.001    | 0.0       | 15.2      | 0.98    | 0.50        | 15.0         | 0.94     | 0.42        |
| 4       | 0            | 0.001    | 0.0       | 12.5      | 0.91    | 0.57        | 12.4         | 0.90     | 0.54        |
| 3       | 1            | 0        | 0.001    | 0.0       | 5.5     | 0.85        | 0.17         | 7.2      | 0.90        | 0.20        |
| 2       | 0            | 0.001    | 0.0       | 6.6       | 0.87    | 0.25        | 6.8          | 0.87     | 0.32        |
| 4       | 0            | 0.001    | 0.0       | 5.8       | 0.86    | 0.32        | 7.8          | 0.88     | 0.15        |
| 5       | 1            | 0        | 0.001    | 0.0       | 14.2    | 0.97        | 0.42         | 14.0     | 0.95        | 0.38        |
| 2       | 0            | 0.001    | 0.0       | 15.7      | 0.99    | 0.55        | 15.4         | 0.98     | 0.47        |
| 6       | 1            | 0        | 0.001    | 0.0       | 8.5     | 0.88        | 0.14         | 9.2      | 0.89        | 0.10        |
| 5       | 2            | 0        | 0.001    | 0.0       | 9.2     | 0.90        | 0.20         | 8.5      | 0.88        | 0.32        |
| 2       | 0            | 0.001    | 0.0       | 6.7       | 0.87    | 0.44        | 5.8          | 0.85     | 0.24        |
| 4       | 0            | 0.001    | 0.0       | 10.0      | 0.89    | 0.22        | 10.0         | 0.91     | 0.12        |
| 6       | 1            | 0        | 0.001    | 0.0       | 10.9    | 0.91        | 0.10         | 11.2     | 0.91        | 0.27        |
| 5       | 2            | 0        | 0.001    | 0.0       | 11.7    | 0.92        | 0.15         | 11.6     | 0.90        | 0.25        |
| 3       | 0            | 0.001    | 0.0       | 12.4      | 0.96    | 0.27        | 12.0         | 0.92     | 0.28        |
| 6       | 1            | 0        | 0.001    | 0.0       | 8.8     | 0.87        | 0.14         | 9.5      | 0.87        | 0.16        |
| 7       | 2            | 0        | 0.001    | 0.0       | 8.7     | 0.86        | 0.27         | 8.1      | 0.86        | 0.10        |
5.2.6.1 Solution of Fuzzy Multi-optimization “Execution Time-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

Aspiration & Tolerance Level of Reliability and Cost

The TFN corresponding to the reliability, deviational execution time and cost aspirations are tabulated in table 5.2.2. The values of these fuzzy parameters are assumed to be specified by the management based on past experiences and/or expert opinion. Defuzzified values of these parameters are computed using the defuzzification function

\[ F_2(A) = \frac{a_l + 2a_m + a_u}{4} \]

and are given in table 5.2.2 along with their tolerance level.

<table>
<thead>
<tr>
<th>Fuzzy Parameter (A)</th>
<th>( a_l )</th>
<th>( a_m )</th>
<th>( a_u )</th>
<th>Defuzzified Value (( F_2(A) )) (Aspiration level)</th>
<th>Tolerance Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.87</td>
<td>0.93</td>
<td>0.95</td>
<td>0.92</td>
<td>0.87</td>
</tr>
<tr>
<td>T</td>
<td>1.1</td>
<td>1.20</td>
<td>2.50</td>
<td>1.50</td>
<td>2.50</td>
</tr>
<tr>
<td>C</td>
<td>95</td>
<td>100</td>
<td>105</td>
<td>100</td>
<td>108</td>
</tr>
</tbody>
</table>

Membership Functions

Membership functions for reliability, execution deviational time and cost can be written as

\[
\begin{align*}
\frac{1}{\sum_{i=1}^l f_i \prod_{j=1}^{n_i} R_j - 0.92} & ; \quad R(X) \geq 0.92 \\
\frac{0.92 - 0.87}{0.92 - 0.87} & ; \quad 0.87 \leq R(X) < 0.92 \\
0 & ; \quad R(X) < 0.87 \\
\frac{1}{2.5 - \sum_{i=1}^l f_i \sum_{j=1}^{n_i} \sum_{k=1}^{n_{ik}} \epsilon_{iak} X_{ik}} & ; \quad T(X) \leq 1.50 \\
\frac{2.5 - 1.50}{2.50 - 1.50} & ; \quad 1.50 \leq T(X) < 2.50 \\
0 & ; \quad T(X) > 2.50
\end{align*}
\]
Assignment of Weights

The assignment of weights is based on the expert’s judgment for the reliability and execution deviational time, i.e. \( w = (w_1, w_2) = \left(0.6, 0.4\right) \).

Fuzzy Goal Programming Approach

A feasible solution is not obtained for problem (5.1.P7), therefore it is solved using fuzzy goal programming approach and can be restated as

**Problem (5.2.P10)**

**Minimize** \( u \)

**Subject to**

\[
\begin{align*}
\mu_R(X) + \eta_1 - \rho_1 &= 1 \\
\mu_T(X) + \eta_2 - \rho_2 &= 1 \\
\mu_C(X) + \eta - \rho &= 1 \\
u &\geq w_j \cdot \eta_j \\
\eta_j \cdot \rho_j &= 0 \quad ; \eta_j, \rho_j \geq 0 \quad ; j = 1, 2 \\
X &\in S \\
\alpha &\in [0, 1] \\
w_1, w_2 &\geq 0 \\
w_1 + w_2 &= 1 \\
\alpha &= 1 - u
\end{align*}
\]

The optimal solution set so obtained for problem (5.2.P10) are optimal for (5.2.P1). Solving problem (5.1.P10) for optimization problem (5.1.P1) with the data set given in table 5.2.1 and table 5.2.2, following solution is obtained
NOTE: Modules (m_2), (m_4) and (m_7) are considered to be critical modules. Redundancy constraint is not added in the optimization model.

Table 5.2.3: Solution of Fuzzy Multi-optimization “Execution Time-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility

<table>
<thead>
<tr>
<th>Modules</th>
<th>Selected COTS Components</th>
<th>Deviational Time</th>
<th>System Reliability</th>
<th>System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x_{111} = x_{123} = x_{131} = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>x_{211} = x_{223} = x_{231} = x_{241} = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>x_{311} = x_{322} = x_{332} = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>x_{411} = x_{422} = 1</td>
<td>1.69</td>
<td>0.90</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>x_{511} = x_{523} = x_{531} = x_{542} = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>x_{611} = x_{623} = x_{632} = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>x_{711} = x_{722} = 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure: 5.2.1: Diagrammatic Representation of Solution for Fuzzy Multi-optimization “Execution Time-Reliability” Model without Constraints on Mandatory Redundancy and Compatibility
By table 5.2.3 and figure 5.2.1 we can easily conclude that redundancy is not allowed in module \((m_1), (m_2), (m_4)\) and \((m_7)\). As already discussed that modules \((m_2), (m_4)\) and \((m_7)\) are critical modules, mandatory therefore redundancy must be allowed in these modules which is discussed in the next section.

### 5.2.6.2 Solution of Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

**Aspiration & Tolerance Level of Reliability and Cost**

The triangular fuzzy reliability and deviational time values are computed using fuzzy values of these parameters and then defuzzified using Heilpern’s defuzzifier. Using the defuzzification function \(F_2(A) = \left(\frac{a_l + 2a_m + a_u}{4}\right)\) we defuzzify aspiration levels given in table 5.2.4.

**Table 5.2.4: Defuzzified Values of Parameters (TFN)**

<table>
<thead>
<tr>
<th>Fuzzy Parameter (A)</th>
<th>(a_l)</th>
<th>(a_m)</th>
<th>(a_u)</th>
<th>Defuzzified Value ((F_2(A))) (Aspiration level)</th>
<th>Tolerance Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.98</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
<td>0.97</td>
</tr>
<tr>
<td>T</td>
<td>2</td>
<td>2.50</td>
<td>3</td>
<td>2.50</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>138</td>
<td>140</td>
<td>150</td>
<td>142</td>
<td>150</td>
</tr>
</tbody>
</table>

**Membership Functions**

Membership functions for reliability, execution deviational time and cost can be written as

\[
\mu_R(X) = \begin{cases} 
1 & ; R(X) \geq 0.99 \\
\frac{\sum \prod R_i - 0.96}{0.99 - 0.96} & ; 0.96 \leq R(X) < 0.99 \\
0 & ; R(X) < 0.96 
\end{cases}
\]
Optimal Component Selection for Fault Tolerant Software Design under Consensus Recovery Block Scheme

\[\mu_R(x) = \begin{cases} 1 & ; T(X) \leq 2.50 \\ \frac{3 - \sum_{l=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} \varepsilon_{jk} x_{jk}}{3 - 2.50} & ; 2.50 \leq T(X) < 3 \\ 0 & ; T(X) > 3 \end{cases}\]

\[\mu_T(X) = \begin{cases} 1 & ; C(X) \leq 140 \\ \frac{150 - \sum_{l=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} c_{jk} x_{jk}}{150 - 140} & ; 140 \leq C(X) < 150 \\ 0 & ; C(X) > 150 \end{cases}\]

Assignment of Weights

The assignment of weights is based on the expert’s judgment for the reliability and execution deviational time, i.e. \( w = (w_1, w_2) = (0.6, 0.4) \).

Fuzzy Goal Programming Approach

A feasible solution is not obtained for problem (5.1.P8) therefore it is solved using fuzzy goal programming approach and can be restated as

Problem (5.2.P11)

Minimize \( u \)

Subject to

\[\mu_R(X) + \eta_1 - \rho_1 = 1\]
\[\mu_T(X) + \eta_2 - \rho_2 = 1\]
\[\mu_C(X) + \eta - \rho = 1\]
\[u \geq \eta_j \ast \eta_j\]
\[\eta_j \ast \rho_j = 0 ; \eta_j, \rho_j \geq 0 ; J = 1, 2\]

\( X \in S \)
\[ \sum_{j=1}^{m_i} z_{ij} \geq 2 \quad ; i \in I_c \]
\[ \sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; i \in \{I - I_c\} \]
\[ R_i \geq R_i^0 \quad ; i \in I_c \]
\[ \alpha \in [0,1] \]
\[ w_1, w_2 \geq 0 \]
\[ w_1 + w_2 = 1 \]
\[ \alpha = 1 - u \]

NOTE: A constraint on criticality of module is added to achieve the redundancy for all the critical modules, that is, for the \( m_2 \), \( m_4 \) and \( m_7 \) module we want at least two alternatives to get selected.

Table 5.2.5: Solution of Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

<table>
<thead>
<tr>
<th>Modules</th>
<th>Selected COTS Components</th>
<th>Deviational Time</th>
<th>System Reliability</th>
<th>System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x_{111} = x_{123} = x_{311} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( x_{211} = x_{223} = x_{231} = x_{243} = 1 )</td>
<td>2.74</td>
<td>0.98</td>
<td>145</td>
</tr>
<tr>
<td>3</td>
<td>( x_{313} = x_{323} = x_{332} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( x_{412} = x_{422} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( x_{513} = x_{523} = x_{532} = x_{541} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>( x_{611} = x_{622} = x_{633} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( x_{712} = x_{722} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 5.2.2: Diagrammatic Representation for Solution of Fuzzy Multi-optimization “Execution Time- Reliability” Model with Constraints on Mandatory Redundancy and without Compatibility

After adding constraint on mandatory redundancy for critical modules the number of selected components has increased and results in increase in the cost and deviational time. Also, reliability of the system gets improved.

5.2.6.3 Solution of Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and Compatibility

Aspiration and tolerance level of reliability, membership functions and assignment of weights are exactly the same as discussed in the above section (For details, refer (Section 5.2.5.2)]. The problem is solved by goal programming approach.

By adding the constraints on compatibility to problem (5.2.P11) a new problem can be written as
Problem (5.2.P12)

Minimize $u$

Subject to

\[
\mu_R(X) + \eta_1 - \rho_1 = 1 \\
\mu_T(X) + \eta_2 - \rho_2 = 1 \\
\mu_C(X) + \eta - \rho = 1 \\
\]

\[
u \geq w_j * \eta_j \\
\eta_j * \rho_j = 0 \quad ; \quad \eta_j, \rho_j \geq 0 \quad ; \quad j=1,2 \\
X \in S \\
\sum_{j=1}^{m_i} z_{ij} \geq 2 \quad ; \quad i \in I_c \\
\sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \quad i \in \{I-I_c\} \\
R_i \geq R_i^0 \quad ; \quad i \in I_c \\
x_{gq} - x_{lw} \leq M \sum y_i \quad ; \quad q = 2,\ldots,V_{gs}, \quad c = 2,\ldots,V_{hs} \quad ; \quad s = 1,\ldots,m_s \\
\sum y_i \leq z(V_{hs} - 2) \\
\alpha \in [0,1] \\
w_1, w_2 \geq 0 \\
w_1 + w_2 = 1 \\
\alpha = 1 - u
\]

For compatibility we use the solution of previous optimization model [For details, refer (Section 5.2.6.2)]. We assume that the second alternative of fifth ($x_{523}$) module is compatible with the second ($x_{623}$) and third ($x_{632}$) alternatives of the sixth module.
Table 5.2.6: Solution of Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and Compatibility

<table>
<thead>
<tr>
<th>Modules</th>
<th>Selected COTS Components</th>
<th>Deviation Time</th>
<th>System Reliability</th>
<th>System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x_{111} = x_{122} = x_{133} = 1 )</td>
<td>2.22</td>
<td>0.98</td>
<td>147.2</td>
</tr>
<tr>
<td>2</td>
<td>( x_{211} = x_{222} = x_{233} = x_{244} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( x_{311} = x_{322} = x_{333} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( x_{411} = x_{422} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( x_{511} = x_{522} = x_{533} = x_{544} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>( x_{611} = x_{622} = x_{633} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( x_{711} = x_{722} = 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2.3: Diagrammatic Representation for Solution of Fuzzy Multi-optimization “Execution Time-Reliability” Model with Constraints on Mandatory Redundancy and Compatibility
It is observed that owing to the compatibility condition, the second alternative of the sixth module \((x_{623})\) is chosen as it is compatible with the second alternative of the fifth \((x_{523})\) module.

5.3 FUZZY MULTI-OPTIMIZATION “BUILD-OR-BUY” MODEL FOR COMPONENT SELECTION WITH MANDATORY REDUNDANCY FOR CRITICAL MODULES

In component-based system development, it is common to identify software modules first. Once they are identified, we need to select appropriate software components for each module. These components can either bought as commercial off-the-shelf components and probably adapted to work in the software system or can be developed in-house. This is called “build-or-buy” decision. This section discusses a framework that helps a developer to decide whether to buy or to build software components while designing a fault-tolerant modular software system. This section proposes optimization models for optimal component selection for a fault-tolerant modular software system under the consensus recovery block scheme. It is necessary to identify critical modules in the design of a fault-tolerant modular software system and also to develop a system with a built in redundancy for critical modules. Therefore, the optimization model is developed for optimal component selection with the dual objective of reliability maximization and cost minimization of the overall system under the constraints on the delivery time and criticality of modules. In practice, it is not possible for a management to obtain precise value of reliability, cost, delivery time, etc., therefore optimization models are formulated as fuzzy multi-objective optimization models. A case study of developing a manufacturing system for medium-size enterprise is used to illustrate the proposed methodology.

5.3.1 Assumptions

Assumptions (1 to 5) discussed in the introduction of this chapter are applicable to this model also. In addition to these five assumptions the following assumptions hold well for optimization models using build-or-buy approach.
6. The cost of an alternative is the development cost, if developed in house; otherwise it is the buying price of the COTS product. Reliability for all the COTS components are known and no separate testing is done.

7. Different COTS alternatives with respect to cost, reliability and delivery time of a module are available.

8. Different in-house alternatives with respect to unitary development cost, estimated development time, average time and testability of module are available.

9. Cost and reliability of an in-house component can be specified by using basic parameters of the development process, e.g. a component cost may depend on a measure of developer skills, or the component reliability depends on the amount of testing.

5.3.2 Model Formulation

Let \( S \) be a software architecture made of \( n \) modules, with a maximum number of \( m_i \) alternatives available for each module. All the constraints of fuzzy multi-optimization build-or-buy model are applicable to this model. [For details, refer Chapter 4 (Section 4.1.3)]. An additional constraint on mandatory redundancy is applicable to this model, which is discussed below.

5.3.2.1 Constraint on Mandatory Redundancy for critical modules

Let \( I \in \{1, 2, \ldots, n\} \) be the set of all modules

\( I_c \) a set of critical modules and a subset of \( I \)

The following constraint is needed to select the module which belongs to the critical set \( I_c \)

\[
\sum_{j=1}^{m_i} z_{ij} \geq 2 \quad ; \quad i \in I_c
\]
Above equation ensures at least a minimum of two alternatives for all modules which are critical are selected.

For all other modules which are not critical, at least one module is selected. The following constraint takes care of this condition.

\[ \sum_{j=1}^{m_i} y_{ij} \geq 1 ; \quad i \in \{ I - I_c \} \]

The minimum reliability requirement is taken care of by the following constraint.

\[ R_i \geq R^0_i ; \quad i \in I_c \]

\( R^0_i \) is the minimum desirable reliability for all critical modules. Reliability requirement for all other modules is taken care of by the general system reliability requirement.

### 5.3.3 Objective Function

The motivation of using CBSE arises from two concepts:

1. Reusable code and architecture and
2. Design that can save the cost of development and that can ensure quality of the developed product.

Therefore the proposed model aims at maximizing system reliability and minimizing the overall cost. The two objectives are formulated in a similar way as discussed in Chapter 4 [For details, refer Chapter 4 (Section 4.1.4)]

### 5.3.4 Problem Description

Multi-optimization “build-or-buy” models are developed for optimal component selection with the dual objective of reliability maximization and cost minimization of the overall system under the constraints on the delivery time and criticality of modules. In practice, it is not possible for a management to obtain precise value of reliability, cost, delivery time, etc., therefore optimization models are formulated as
fuzzy multi-objective optimization models. Different optimization models are developed to have better understanding of the problem. The problem considered here can now be stated and discussed in the following three sections.

5.3.4.1 Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection without Constraints on Mandatory Redundancy and Compatibility

Multi-optimization “build-or-buy” model under fuzzy environment is formulated in Problem (5.3.P1). In the optimization model it is assumed that the alternatives of a module are in a Consensus Recovery Block [For details, refer Chapter 1, Section (1.1.7.3)].

Problem (5.3.P1)

Maximize \( R = \sum_{i=1}^{I} f_i \prod_{j \in I} R_j \)

Minimize \( C = \sum_{i=1}^{n} \left( \sum_{j=1}^{m} c_{ij} \left( t_{ij} + \tau_{ij} N_{ij}^{\text{tot}} \right) y_{ij} + \sum_{j=1}^{m} C_{ij} x_{ij} \right) \)

Subject to

\[ X \in S = \{ x_{ij} \text{ and } y_{ij} \text{ are binary variable/} \]

\[ R = 1 + \sum_{k=1}^{m} \frac{1}{m} \left[ \prod_{t_i} \left( 1 - r_{ik} \right)^{\pi_{ik}} \right] \left[ 1 - \left( 1 - r_{ik} \right)^{\pi_{ik}} \right] + \sum_{k=1}^{m} \left[ \prod_{t_i} \left( 1 - r_{ik} \right)^{\pi_{ik}} \right] \left( \prod_{j=1}^{m} \left[ \prod_{t_i} \left( 1 - r_{ij} \right)^{\pi_{ij}} \right] \right) \right] \]

\[ \bigg( \sum_{k=1}^{m} \left[ \prod_{t_i} \left( 1 - r_{ik} \right)^{\pi_{ik}} \right] \bigg) - 1 \]

\[ i = 1, 2, ..., n \]

\[ (5.3.1) \]

\[ P(X_{ij}) = (1 - t_{1}) \left[ \left( 1 - r_{ij} \right) \left( 1 - t_{3} \right) + r_{ij} t_{2} \right] \]

\[ (5.3.2) \]

\[ P(y_{ij}) = r_{ij} \left( 1 - t_{2} \right) \]

\[ (5.3.3) \]

\[ N_{ij}^{\text{arc}} = \left( 1 - \pi_{ij} \right) N_{ij}^{\text{tot}} ; \quad i = 1, 2, ..., n \quad ; \quad j = 1, 2, ..., m_{i} \]

\[ (5.3.4) \]

\[ \rho_{ij} = \frac{1 - \pi_{ij}}{\left( 1 - \pi_{ij} \right) + \pi_{ij} \left( 1 - \pi_{ij} \right)^{N_{ij}^{\text{arc}}}} ; \quad i = 1, 2, ..., n \text{ and } j = 1, 2, ..., m_{i} \]

\[ (5.3.5) \]
Chapter 5: Fuzzy Multi-objective Approach to Component Selection for COTS based Modular Software System Incorporating Mandatory Redundancy for Critical Modules

\[ r_{ij} = \rho_y y_{ij} + s_j x_{ij} \quad ; \quad i = 1, 2, \ldots, n \quad ; \quad j = 1, 2, \ldots, m_i \]  
\[ \ldots (5.3.6) \]

\[ y_{ij} + x_{ij} \leq 1 \quad ; \quad i = 1, 2, \ldots, n \quad ; \quad j = 1, 2, \ldots, m_i \]  
\[ \ldots (5.3.7) \]

\[ y_{ij} + x_{ij} = z_{ij} \quad ; \quad i = 1, 2, \ldots, n \quad ; \quad j = 1, 2, \ldots, m_i \]  
\[ \ldots (5.3.8) \]

\[ \sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \quad i = 1, 2, \ldots, n \]  
\[ \ldots (5.3.9) \]

\[ y_{ij} \left( t_{ij} + r_{ij} N_{ij} \right) + d_{ij} x_{ij} \leq T_u \]  
\[ \ldots (5.3.10) \]

where \( X \) is a vector of components \( (x_{ij}, y_{ij}, z_{ij}) \quad ; \quad i = 1, \ldots, n \quad ; \quad j = 1, \ldots, m_i) \)

Here the first objective stated is the reliability objective and the second one is the cost objective. Constraint (5.3.1) estimates the reliability of module \( i \) for a system under consensus recovery block scheme. Constraint (5.3.2) is the probability of event that output of alternative \( j \) of module \( i \) is rejected and Constraint (5.3.3) is the probability of event that correct result of alternative \( j \) of module \( i \) is accepted. Constraint (5.3.4) ensures that \( N^{\text{free}}_{ij} \) failure free test cases have performed. Constraint (5.3.5) is the probability that in-house developed alternative is failure free during a single run given that \( N^{\text{free}}_{ij} \) test cases have successfully performed. Constraint (5.3.6) is the reliability of alternative \( j \) of module \( i \). Constraint (5.3.7) ensures that if an alternative is bought then there is no in-house development and vice-versa. Constraint (5.3.8) and (5.3.9) guarantees that redundancy is allowed in the modules. Constraint (5.3.10) is the delivery time constraint.

5.3.4.2 Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Constraints on Mandatory Redundancy and without Compatibility

It is discussed in the introduction that all the critical modules are identified during the initial phase of software development. One way to prevent the system from failure is to incorporate mandatory redundancy in critical modules. Therefore adding the constraint of mandatory redundancy to problem (5.3.P1), the problem can be written as
Problem (5.3.P2)

\[ \text{Maximize } R = \sum_{i=1}^{I} f_i \prod_{j \in X_i} R_j \]

\[ \text{Minimize } C = \sum_{j=1}^{n} \left( \sum_{i=1}^{m} c_{ij} \left( t_{ij} + \tau_y N_{ij}^{tot} \right) y_{ij} + \sum_{j=1}^{m} C_{ij} x_{ij} \right) \]

Subject to

\[ X \in S \]

\[ \sum_{j=1}^{m} z_{ij} \geq 2 \quad ; i \in I_c \quad \text{...}(5.3.11) \]

\[ \sum_{j=1}^{m} z_{ij} \geq 1 \quad ; i \notin \{ I - I_c \} \quad \text{...}(5.3.12) \]

\[ R_i \geq R_i^0 \quad ; i \in I_c \quad \text{...}(5.3.13) \]

5.3.4.3 Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Constraints on Mandatory Redundancy and Compatibility

The optimization model discussed in this section considers the issue of compatibility amongst the alternatives of the modules. [For details, refer Chapter 4 (Section 4.1.5.2)]

Problem (5.3.P3)

\[ \text{Maximize } R = \sum_{i=1}^{I} f_i \prod_{j \in X_i} R_j \]

\[ \text{Minimize } C = \sum_{j=1}^{n} \left( \sum_{i=1}^{m} c_{ij} \left( t_{ij} + \tau_y N_{ij}^{tot} \right) y_{ij} + \sum_{j=1}^{m} C_{ij} x_{ij} \right) \]

Subject to

\[ X \in S \]

\[ \sum_{j=1}^{m} z_{ij} \geq 2 \quad ; i \in I_c \]
Chapter 5: Fuzzy Multi-objective Approach to Component Selection for COTS based Modular Software System Incorporating Mandatory Redundancy for Critical Modules

\[ \sum_{j=1}^{m_i} z_{ij} \geq 1 \quad ; \quad i \in \{ I-I_c \} \]

\[ R_i \geq R^0_i \quad ; \quad i \in I_c \]

\[ x_{gs} - x_{hu_i} \leq My_i \quad \text{...(5.3.14)} \]

\[ \sum_{i=1}^{z} y_i \leq z-1 \quad \text{...(5.3.15)} \]

Problems [(5.3.P1), (5.3.P2) and (5.3.P3)] cannot be solved using crisp optimization techniques as these methods do not provide well defined mechanism to handle ambiguities. Therefore, we use fuzzy optimization approach to solve these problems.

5.3.5 Solution Procedure

The problem is solved exactly in the same way as solved for optimization models discussed in Chapter 4 [For details, refer Chapter 4 (Section 4.1.6)] using ‘Fuzzy Optimization Algorithm’.

5.3.6 Case Study

In this section, a case study of CBSE is presented to illustrate the proposed methodology of optimizing the selection of software components for a fault-tolerant modular software system, using build-or-buy strategy. The components are selected in such a way that the total reliability of the system is maximized and the overall cost is minimized, under the constraint of criticality of modules and also the delivery time constraint. A software system supplier planned to develop a software system for a medium-size manufacturing company which provides the assembly manufacturing of Light Fittings and Accessories. Functional requirements of the system were identified, namely, maintaining and planning inventory, estimating stock levels, making sales quotation, preparing sales order and procurement plan, business rules, financial reports and schedules, Customer Relationship Management (CRM) process, employee master, salary plan and appraisals and so on. The software system development team of the company has defined eight software modules, Production Planning (m_1), Sales Planning (m_2), and Procurement Planning (m_3), Back Office (m_4), Financial Management (m_5), Product Design/R&D (m_6), CRM (m_7), HRM/Payroll (m_8). The functional requirements of the modules are mentioned in table 5.3.1.
Table 5.3.1: Functional Requirements of Modules

<table>
<thead>
<tr>
<th>Module No.</th>
<th>Module Specification</th>
<th>Functional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>m₁</td>
<td>Production Planning</td>
<td>Maintain inventory safety stock and MIN MAX levels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trigger alerts at pre-set MIN levels on critical items</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automatic requisition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Raw material reservations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Material forecast report</td>
</tr>
<tr>
<td>m₂</td>
<td>Sales Planning</td>
<td>Estimate current stock levels before preparing sales quotation to customer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consult production capacity plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Book sales quote and convert to order</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Handover to production department with job card and delivery notifications</td>
</tr>
<tr>
<td>m₃</td>
<td>Procurement Planning</td>
<td>Purchase requisition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Request for Information (RFI) letter to suppliers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quote comparison reports</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conversion of quote to purchase orders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Purchase order analysis and follow ups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Procurement plan based on production and sales demand and forecast</td>
</tr>
<tr>
<td>m₄</td>
<td>Back Office</td>
<td>Inventory Control and Movements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E-commerce</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automatic Updates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Security and Administration</td>
</tr>
<tr>
<td>m₅</td>
<td>Financial Management</td>
<td>Business rules/ protocols</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Financial reports and schedules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Balance sheet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Profit /Loss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Cash flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Ration analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. Fixed asset register</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. Receivables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. Payables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. Bank ledgers</td>
</tr>
</tbody>
</table>
A total of 25 COTS components available in the market were considered. Information on cost, reliability and delivery time of each COTS component is given by the vendor. The cost, reliability and development time of building these components were also estimated as the software supplier can also build these components in-house. The decision is to choose the right components for each software module, so as to get a reliable software system at a minimum cost with the desired delivery time and with the constraint on criticality of modules.

The system consists of eight modules. Each module provides different functional requirements. A system is to be developed by integrating components/alternatives which can be either COTS or the in-house-built components. The objective of this study is to select optimal set of alternatives for each module so as to get a highly reliable
manufacturing software system. For each module, various alternatives are available and various COTS versions are available for each alternative of a module. Furthermore, an in-house alternative for each module can be built. The data-sets for COTS and in-house developed components are given in table 5.3.2 and table 5.3.3 respectively.

Let the software is required to perform four functions, so $L = 3$. The set of modules required for the four functions are given by $S_1 = \{1,2,3,4,5,6,7,8\}$, $S_2 = \{1,2,4,5,7,8\}$, $S_3 = \{1,2,5,7,8\}$ and $S_4 = \{1,2,5,6,8\}$. The frequency of use is given by $f_1 = 0.40$, $f_2 = 0.30$, $f_3 = 0.15$ and $f_4 = 0.15$. It is also assumed that $t_1 = 0.01$, $t_2 = 0.05$ and $t_3 = 0.01$.

**Table 5.3.2: Data set of COTS components for Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Mandatory Redundancy for Critical Modules**

<table>
<thead>
<tr>
<th>Module No.</th>
<th>Module Specification</th>
<th>COTS Alternatives</th>
<th>Cost</th>
<th>Reliability</th>
<th>Delivery Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_1</td>
<td>Production Planning</td>
<td>$x_{11}$</td>
<td>8</td>
<td>0.87</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{12}$</td>
<td>10</td>
<td>0.89</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{13}$</td>
<td>7</td>
<td>0.86</td>
<td>4</td>
</tr>
<tr>
<td>m_2</td>
<td>Sales Planning</td>
<td>$x_{21}$</td>
<td>25</td>
<td>0.95</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{22}$</td>
<td>18</td>
<td>0.91</td>
<td>2</td>
</tr>
<tr>
<td>m_3</td>
<td>Procurement Planning</td>
<td>$x_{31}$</td>
<td>16</td>
<td>0.92</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{32}$</td>
<td>15</td>
<td>0.88</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{33}$</td>
<td>18</td>
<td>0.95</td>
<td>4</td>
</tr>
<tr>
<td>m_4</td>
<td>Back Office</td>
<td>$x_{41}$</td>
<td>15</td>
<td>0.90</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{42}$</td>
<td>17</td>
<td>0.94</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{43}$</td>
<td>21</td>
<td>0.96</td>
<td>3</td>
</tr>
<tr>
<td>m_5</td>
<td>Financial Management</td>
<td>$x_{51}$</td>
<td>10</td>
<td>0.86</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{52}$</td>
<td>12</td>
<td>0.88</td>
<td>3</td>
</tr>
<tr>
<td>m_6</td>
<td>Product Design/ R&amp;D</td>
<td>$x_{61}$</td>
<td>27</td>
<td>0.95</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{62}$</td>
<td>22</td>
<td>0.90</td>
<td>3</td>
</tr>
<tr>
<td>m_7</td>
<td>CRM</td>
<td>$x_{71}$</td>
<td>15</td>
<td>0.92</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{72}$</td>
<td>22</td>
<td>0.96</td>
<td>1</td>
</tr>
<tr>
<td>m_8</td>
<td>HRM/ Payroll</td>
<td>$x_{81}$</td>
<td>19</td>
<td>0.95</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{82}$</td>
<td>22</td>
<td>0.96</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_{83}$</td>
<td>17</td>
<td>0.93</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 5.3.2 shows the cost, reliability, and delivery time for the COTS components. The second column of Table 5.3.2 lists the eight modules of the software system. The third column provides various alternatives for each module. The fourth, fifth and sixth columns provide the parameters of cost (in Kilo Euros), reliability and delivery time (in weeks) for each alternative, respectively.

Table 5.3.3: Data set of In-house Components for Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Mandatory Redundancy for Critical Modules

<table>
<thead>
<tr>
<th>Module No.</th>
<th>Module Specification</th>
<th>COTS Alternatives</th>
<th>( t_{ij} )</th>
<th>( r_{ij} )</th>
<th>( c_{ij} )</th>
<th>( \pi_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_1</td>
<td>Production Planning</td>
<td>( y_{11} )</td>
<td>4</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{12} )</td>
<td>5</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{13} )</td>
<td>4</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td>m_2</td>
<td>Sales Planning</td>
<td>( y_{21} )</td>
<td>6</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{22} )</td>
<td>7</td>
<td>0.05</td>
<td>3</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{23} )</td>
<td>8</td>
<td>0.05</td>
<td>3</td>
<td>0.002</td>
</tr>
<tr>
<td>m_3</td>
<td>Procurement Planning</td>
<td>( y_{31} )</td>
<td>4</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{32} )</td>
<td>5</td>
<td>0.05</td>
<td>6</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{33} )</td>
<td>6</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{34} )</td>
<td>6</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td>m_4</td>
<td>Back Office</td>
<td>( y_{41} )</td>
<td>6</td>
<td>0.05</td>
<td>3</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{42} )</td>
<td>5</td>
<td>0.05</td>
<td>2</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{43} )</td>
<td>4</td>
<td>0.05</td>
<td>2</td>
<td>0.002</td>
</tr>
<tr>
<td>m_5</td>
<td>Financial Management</td>
<td>( y_{51} )</td>
<td>5</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{52} )</td>
<td>6</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{53} )</td>
<td>4</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td>m_6</td>
<td>Product Design/ R&amp;D</td>
<td>( y_{61} )</td>
<td>4</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{62} )</td>
<td>5</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td>m_7</td>
<td>CRM</td>
<td>( y_{71} )</td>
<td>4</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{72} )</td>
<td>3</td>
<td>0.05</td>
<td>3</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{73} )</td>
<td>5</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td>m_8</td>
<td>HRM/ Payroll</td>
<td>( y_{81} )</td>
<td>4</td>
<td>0.05</td>
<td>4</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{82} )</td>
<td>3</td>
<td>0.05</td>
<td>3</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y_{83} )</td>
<td>5</td>
<td>0.05</td>
<td>5</td>
<td>0.002</td>
</tr>
</tbody>
</table>
Table 5.3.3 shows the parameters that we have collected for in-house development of components. For each component, the average development time $t_{ij}$ (in weeks) is given in the fourth column and the average time required to perform a single test $\tau_{ij}$ (in days) is given in the fifth column, the unitary development cost $c_{ij}$ (KE per week) is given in the sixth column, finally the component testability $\pi_{ij}$ is given in the last column.

**Aspiration & Tolerance Level of Reliability and Cost**

The TFN corresponding to the reliability and cost aspirations are tabulated in table 5.3.4. Defuzzified values of these parameters are computed using the defuzzification function $F_2(A) = \frac{(a_l + 2a_m + a_u)}{4}$ and are given in table 5.3.4 along with their tolerance level.

**Table 5.3.4: Defuzzified Values of Parameters (TFN)**

<table>
<thead>
<tr>
<th>Delivery Time</th>
<th>TFN $(a_l, a_m, a_u)$</th>
<th>Defuzzified Value $(F_2(A))$</th>
<th>Tolerance Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>$R = (0.84, 0.90, 0.92)$, $C = (540, 550, 580)$</td>
<td>$R_0 = 0.89$ $C_0 = 555$</td>
<td>$R_0^* = 0.82$ $C_0^* = 580$</td>
</tr>
<tr>
<td>10</td>
<td>$R = (0.95, 0.97, 0.99)$, $C = (520, 540, 560)$</td>
<td>$R_0 = 0.97$ $C_0 = 540$</td>
<td>$R_0^* = 0.92$ $C_0^* = 570$</td>
</tr>
<tr>
<td>14</td>
<td>$R = (0.95, 0.97, 0.99)$, $C = (540, 560, 580)$</td>
<td>$R_0 = 0.97$ $C_0 = 560$</td>
<td>$R_0^* = 0.92$ $C_0^* = 580$</td>
</tr>
</tbody>
</table>

**Membership Functions**

Membership functions can be written by adding the values of tolerance and aspiration levels given in table 5.3.4. Membership functions for different delivery time will be different. The aspiration and tolerance level of reliability and cost changes when we change the delivery time.

**Assignment of Weights**

The assignment of weights is based on the expert’s judgment for the reliability and execution deviational time, i.e. $w = (w_1, w_2) = (0.6, 0.4)$.
5.3.6.1 Solution of Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection without Constraints on Mandatory Redundancy and Compatibility

The model is solved using a software package LINGO (Version 11). The solution to model gives the optimal components selection for the software system along with the corresponding cost and reliability of the overall system under fuzzy environment. It is decided by the management that the back office (m₄), financial management (m₅) and HRM/Payroll (m₈) are considered to be the critical modules. There must be redundancy attached to these modules. Solution to the model for different delivery times are given below.

CASE 1: Delivery time is 7 weeks. Constraint on criticality of modules is not added

Figure 5.3.1: Solution of Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection without Constraints on Mandatory Redundancy and Compatibility (Case 1)

Since the constraint on criticality of modules in not added, therefore, redundancy is not allowed in modules (m₁), (m₅) and (m₈). However, modules (m₅) and (m₈) are critical modules. System reliability is 0.87 and the overall cost is 564.
5.3.6.2 Solution of Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Constraints on Mandatory Redundancy and without Compatibility

CASE 2: Delivery time is 7 weeks. Constraint on criticality of modules is added

In Case 2, the constraint on module criticality is added, it can be seen from the solution above that all critical modules have at least two components selected. For back office (m₄) module three in-house built components were selected, two COTS products for financial management (m₅) module and one in-house and one COTS product for HRM/Payroll (m₈) module. Redundancy improves system reliability and requires additional resources. Therefore, system reliability is 0.94 and the overall cost is 582.

CASE 3: Delivery time is 10 weeks
Chapter 5: Fuzzy Multi-objective Approach to Component Selection for COTS based Modular Software System Incorporating Mandatory Redundancy for Critical Modules

At delivery time 10 weeks, we have added a constraint on criticality. So from the solution it can be seen that all critical modules, namely, Back office \( (m_4) \), Financial Management \( (m_5) \), and HRM/Payroll \( (m_8) \) have redundant components. System reliability is 0.96 and the overall cost is 543.

CASE 4: Delivery time is 14 weeks

![Figure 5.3.4: Solution of Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Constraints on Mandatory Redundancy and without Compatibility (Case 4)](image)

At delivery time 14 weeks, it can be seen that redundancy is allowed in all the eight modules. System reliability is 0.972 and the overall cost is 572.

In Case 1, system reliability is 0.87 and overall system cost is 564. But this solution is not accepted because redundancy is not allowed in critical modules. Therefore, in Case 2, a constraint on criticality of module is added, which in turn give a system reliability of 0.94 and cost 582.

In Case 3, when delivery time increases from 7 to 10 weeks, there is a significant reduction in the overall cost and also there is an increase in reliability. More of in-house built components are selected which reduces the system cost and redundancy is also allowed in all critical modules.

In Case 4, redundancy is allowed in all eight modules, there is no significant increase in the reliability but the cost has increased significantly.

Therefore as part of decision making manager will choose case 2 as the best decision because we are getting a higher reliability at a lowest cost.
5.3.6.3 Solution of Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Constraints on Mandatory Redundancy and Compatibility

CASE 5: Delivery time is 10 weeks

We assume that the third alternative of the fifth module is compatible with first and third alternatives of third module.

![Diagram of Solution of Fuzzy Multi-optimization “Build-or-Buy” Model for Component Selection with Constraints on Mandatory Redundancy and Compatibility (Case 5)](image)

It is observed that due to the compatibility condition, the third alternative of third module is chosen as it is compatible with the third alternative of fifth module. The overall system cost is 557 and system reliability is 0.92.