The type of mathematical programming which have been used for application purpose while designing optimization models is given as follows:

Type 1: Unconstrained Single-objective Optimization Problem
An optimization or a mathematical programming problem can be stated as follows:

\[ \begin{align*}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
\end{align*} \]

To find \( X \) which minimizes \( f(x) \)

where \( X \) is an \( n \)-dimensional vector called the decision vector, \( f(x) \) is called the objective function. This type of problem is called as unconstrained optimization problem. The field of unconstrained optimization is quite a large and prominent one, for which a lot of algorithms and software are available.

Type 2: Constrained Single-objective Optimization problem
Optimization problem can be defined with constraints as well.

\[ \begin{align*}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
\end{align*} \]

To find \( X \) which minimizes \( f(x) \)

subject to

\[ 
    g_i(X) \leq M \quad i = 1, 2, \ldots, m \\
    l_j(X) = P \quad j = 1, 2, \ldots, p
\]

where \( g_i(X) \) and/or \( l_j(X) \) are known as inequality and equality constraints with restriction level \( M \) and \( P \) respectively. the number of variables is \( n \) and the number of constraints is \( m \) and/or \( p \) need not be related in any way. Such problems are called constrained optimization problems.
Type 3: Unconstrained Multi-objective Optimization Problem

Multi-objective optimization problem with total number of $m$ conflicting objectives can be formulated as:

$$\text{Minimize } y = f(x) = (f_1(x), f_2(x), \ldots, f_m(x))$$

subject to

$$x = (x_1, x_2, \ldots, x_n) \in X$$
$$y = (y_1, y_2, \ldots, y_m) \in Y$$

where $x, X, y$ and $Y$ are called decision vector, decision space, objective vector, and objective space, respectively. $f_i(x)$ is the $i^{th}$ objective function of the problem. This type of problem is called as unconstrained multi-objective optimization problem.

Type 4: Constrained Multi-objective Optimization Problem

Multi-objective optimization can be defined with multiple constraints as well:

$$\text{Minimize } y = f(x) = (f_1(x), f_2(x), \ldots, f_m(x))$$

subject to

$$g_j(x) \leq P_j \quad j = 1, 2, \ldots, p$$
$$l_k(x) = Q_k \quad k = 1, 2, \ldots, q$$

$$x = (x_1, x_2, \ldots, x_n) \in X$$
$$y = (y_1, y_2, \ldots, y_m) \in Y$$

where $g_j(x)$ is the $j^{th}$ inequality constraints and/or $l_k(x)$ is the $k^{th}$ equality constraints with restriction level $P_j$ and $Q_k$ respectively. The number of inequality constraints and equality constraints are $j$ and $k$ respectively and the number of constraints is $j$ and/or $k$ need not be related in any way. This type of problem is called a constrained multi-objective optimization problem.
Appendix B

Preliminary Concepts of Fuzzy Set Theory

**Fuzzy set:** Let $X$ be the universe whose generic element is denoted by $x$. A fuzzy set $A$ in $X$ is a function $A: X \rightarrow [0,1]$. Fuzzy set $A$ is characterized by its membership function $\mu_A: X \rightarrow [0,1]$ which, associates with each $x$ in $X$, a real number $\mu_A(x)$ in $[0,1]$ representing the grade of $x$ in $A$.

**Support of a fuzzy set:** The support of a fuzzy set $A$ in $X$, denoted by $S(A)$, is the crisp set given by $S(A) = \{x \in X : \mu_A(x) > 0\}$.

**Normal Fuzzy Set:** The height $h(A)$ of a fuzzy set $A$ is defined as

$$h(A) = \sup_{x \in X} \mu_A(x) > 0$$

if $h(A) = 1$, then the fuzzy set $A$ is called a normal fuzzy set, otherwise subnormal which can be normalized as $\frac{\mu_A(x)}{h(A)}, x \in X$.

**Standard Union:** The standard union of two fuzzy sets $A$ and $B$ is a fuzzy set $C$ whose membership function is given by $\mu_C(x) = \max(\mu_A(x), \mu_B(x))$ for all $x \in X$. This we express as $C = A \cup B$.

**Standard Intersection:** The standard intersection of two fuzzy sets $A$ and $B$ is a fuzzy set $D$ whose membership function is given by $\mu_D(x) = \min(\mu_A(x), \mu_B(x))$ for all $x \in X$. This we express as $C = A \cap B$.

**$\alpha$-cut:** The $\alpha$-cut of the fuzzy set $A$ in $X$ is the crisp set $A_\alpha$ given by $A_\alpha = \{x \in X : \mu_A(x) > \alpha\}$ where $\alpha \in (0,1]$.

**Convex fuzzy set:** A fuzzy set $A$ in $\mathbb{R}^n$ is said to be a convex fuzzy set if its $\alpha$-cuts $A_\alpha$ are (crisp) convex sets for all $\alpha \in (0,1]$. 

**Theorem 1:** A fuzzy set $A$ in $\mathbb{R}^n$ is said to be a convex fuzzy iff for all $x_1, x_2 \in \mathbb{R}^n$ and $0 \leq \lambda \leq 1$

\[
\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2))
\]

**Zadeh’s Extension Principle:** Let $f : X \rightarrow Y$ be a crisp function and $F(X)$ ($F(Y)$) be the set of all fuzzy sets of $X$ ($Y$). The function $f : X \rightarrow Y$ induces two functions $f : F(X) \rightarrow F(Y)$ and $f^{-1} : F(X) \rightarrow F(Y)$. The extension principle gives formulas to compute the membership function of fuzzy sets $f(A)$ in $Y$ ($f^{-1}(B)$ in $X$) in terms of membership functions of fuzzy sets $A$ in $X$ ($B$ in $Y$). The principle states that

1. $\mu_{f(A)}(y) = \sup_{A \in F(X)} (\mu_A(x))$, $\forall A \in F(X)$

2. $\mu_{f^{-1}(B)}(x) = \mu_B(x)$, $\forall B \in F(Y)$

If the function $f$ maps a $n$-tuple in $X$ to a point in $Y$ and $f : X \rightarrow Y$ given by $y = f(x_1, x_2, \ldots, x_n)$. Let $A_1, A_2, \ldots, A_n$ be $n$ fuzzy sets in $X_1, X_2, \ldots, X_n$ respectively. The extension principle of Zadeh allows to extend the crisp function $y = f(x_1, x_2, \ldots, x_n)$ to act on $n$ fuzzy subsets of $X$, namely $A_1, A_2, \ldots, A_n$ such that $B = f(\{A_1, A_2, \ldots, A_n\})$

The fuzzy set $B$ is defined as

\[
B = \{(y, \mu_B(y)) : y = f(x_1, x_2, \ldots, x_n), (x_1, x_2, \ldots, x_n) \in X_1 \ast \ast X_n\}
\]

and

\[
\mu_B(y) = \sup_{x \in X, y \in f(x)} \min(\mu_{A_1}(x_1), \ldots, \mu_{A_n}(x_n))
\]

**Fuzzy number:** A fuzzy set $A$ in $\mathbb{R}$ is called a fuzzy number if it satisfies the following conditions

(i) $A$ is normal

(ii) $A$ is convex

(iii) $\mu_A$ is upper semi continuous

(iv) Support of $A$ is bounded
Theorem 2: Let $A$ be a fuzzy set in $\mathbb{R}$. Then $A$ is a fuzzy number if and only if there exists a closed interval (which may be singleton) $[a, b] \neq \emptyset$ such that

$$\mu_A(x) = \begin{cases} 1, & x \in [a, b] \\ l(x), & x \in (-\infty, a] \\ r(x), & x \in [b, \infty] \end{cases}$$

where

(i) $l:(-\infty, a)\rightarrow[0,1]$ is non-decreasing, continuous from the right and $l(x)=0$ for $x \in (-\infty, w_1), w_1 < a$

(ii) $r:(b, \infty)\rightarrow[0,1]$ is non-increasing, continuous from the left and $r(x)=0$ for $x \in (w_2, \infty), w_2 > b$ and $\mu_A(x)$ is called ‘Membership Function’ of fuzzy set $A$ on $\mathbb{R}$.

An element mapping to the value 0 means that the member is not included in the given set, 1 describes a fully included member. Values strictly between 0 and 1 characterize the fuzzy members. Figure B.1 illustrate a fuzzy set graphically.

**Triangular fuzzy number (TFN):** A fuzzy number $A$ denoted by the triplet $A = (a_l, a, a_u)$ having the shape of a triangle is called a TFN. The $\alpha$-cut of a TFN is the closed interval $A_{\alpha} = [a_{\alpha}^L, a_{\alpha}^R] = [(a-a_l)\alpha + a_l, (a-a_u)\alpha + a_u], \alpha \in (0,1]$ and its membership function $\mu_\alpha$ is given by
\[
\mu_A(x) = \begin{cases} 
0, & x < a_l, x > a_u, \\
(x - a_l) / (a - a_l), & a_l \leq x \leq a, \\
(a_u - x) / (a_u - a), & a < x \leq a_u
\end{cases}
\]

**Ranking of fuzzy numbers**

Ranking of fuzzy numbers is an important issue in the study of fuzzy set theory and is useful in various applications. Fuzzy mathematical programming is one of the applications. There are numerous methods proposed in literature for ranking the fuzzy numbers such as ranking function (index) approach, k-preference index approach and possibility theory approach, useful in particular context but not in general. We use the Ranking function (index) approach for ranking the fuzzy numbers for our problem.

**Ranking function (index) approach**

Let \( N(\mathbb{R}) \) be the set of all fuzzy numbers in \( \mathbb{R} \) and \( A, B, \in N(\mathbb{R}) \). Define a function \( F : N(\mathbb{R}) \rightarrow \mathbb{R} \), called a ranking function or ranking index, where \( F(A) \leq F(B) \) is equivalent to \( A(\leq) B \). Following indices are proposed by Yager [15].

(i) \( F_1(A) = \left( \int_{a_l}^{a_u} x \mu_A(x) dx \right) / \left( \int_{a_l}^{a_u} \mu_A(x) dx \right) \), Where \( a_l \) and \( a_u \) are the lower and upper limits of the support of \( A \). The value \( F_1(A) \) is the centroid of the fuzzy number \( A \in N(\mathbb{R}) \). For example, If \( A = (a_l, a, a_u) \) is a triangular fuzzy number (TFN) where \( a_l \) and \( a_u \) are the lower and upper limits of the support of \( A \) and \( a \) is the model value then \( F_1(A) = (a_l + a + a_u) / 3 \).

(ii) \( F_2(A) = \left( \int_0^{a_{\max}} m[a_{\alpha}^L, a_{\alpha}^R] d\alpha \right) \), Where \( a_{\max} \) is the height of \( A \), \( A_\alpha = [a_{\alpha}^L, a_{\alpha}^R] \) is a \( \alpha \)-cut, \( \alpha \in (0, 1) \), and \( m[a_{\alpha}^L, a_{\alpha}^R] \) is the mean value of elements of the \( \alpha \)-cut. For example, \( A = (a_l, a, a_u) \) is a TFN, \( a_{\max} = 1 \) and \( A_\alpha = [a_{\alpha}^L, a_{\alpha}^R] = [(a - a_l)\alpha + a_l, (a - a_u)\alpha + a_u] \) then \( m[a_{\alpha}^L, a_{\alpha}^R] = ((a - a_l)\alpha + a_l + (a - a_u)\alpha + a_u) / 2 \) and \( F_2(A) = (a_l + 2a + a_u) / 4 \).
Fuzzy Mathematical Programming Problem

For the mathematical programming problems in the crisp scenario, the aim is to maximize or minimize the objective function under certain set of constraints. But in many practical situations, the decision maker may not be in a position to specify the objective and/or constraint functions precisely but rather can specify them in a “fuzzy sense”. In such situations, it is desirable to use some fuzzy programming type of modeling so as to provide more flexibility to the decision maker. Since the fuzziness may appear in a mathematical programming problem in many ways (e.g. the inequalities may be fuzzy, the goals may be fuzzy or the problem parameters \( c, A, b \) may be in terms of fuzzy numbers), the definition of fuzzy linear programming problem is not unique. This thesis aims to study optimization models which have fuzzy inequalities and objective function(s).

Mathematical Programming with fuzzy inequalities and objective function

The general model of a mathematical programming problem with fuzzy objective and fuzzy constraints is formulated as:

\[(Problem \ P1)\]

\[
\begin{align*}
\text{Maximize} & \quad C^T x \\
\text{subject to} & \quad Ax \lesssim b \\
& \quad x \geq 0
\end{align*}
\]

The symbol \( \lesssim \) is called “fuzzy greater (less) than or equal to and have linguistic interpretation “essentially greater (less) than or equal to”. Crisp optimization techniques can’t be applied directly to solve the problem since these methods provide no well-defined mechanism to handle the uncertainties quantitatively. Hence we use a fuzzy optimization approach to solve the problem. The two objectives can be assigned different weights according the relative importance and the problem can be solved with the weighted min-max approach.
In the next section using the concepts of fuzzy set theory we discuss optimization technique to solve the problem (P1).

The Solution

Following algorithm specifies the sequential steps to solve the fuzzy mathematical programming problems. Figure 3.1.1 illustrates the solution methodology in the form of a flowchart.

Algorithm 1

1. Compute the crisp equivalent of the fuzzy parameters using a defuzzification function (ranking of fuzzy numbers). Same defuzzification function is to be used for each of the parameters. Here we use the defuzzification function of type
   \[ F_2(A) = \left(a_1 + 2a_2 + a_3\right)/4. \]

2. Incorporate the objective function of the fuzzifier min (max) as a fuzzy constraint with a restriction (aspiration) level. The inequalities are defined softly if the requirement (resource) constants are defined imprecisely.

3. Define appropriate membership functions for each fuzzy inequalities as well as 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
   \[
   \mu(T) = \begin{cases} 
   1 & : G(T) \leq G_0 \\
   \frac{G^* - G(T)}{G^* - G_0} & : G_0 < G(T) \leq G^* \\
   0 & : G(T) > G^* 
   \end{cases}
   \]
   \[
   \mu(T) = \begin{cases} 
   1 & : Q(T) \geq Q_0 \\
   \frac{Q - Q^*}{Q_0 - Q^*} & : Q^* \leq Q(T) < Q_0 \\
   0 & : Q(T) < Q^* 
   \end{cases}
   \]
   respectively, where \( G_0 \) and \( Q_0 \) are the restriction and aspiration levels respectively and \( G^* \) and \( Q^* \) are the corresponding the tolerance levels. The membership functions can be a linear or piecewise linear function that is concave or quasiconcave.

4. Employ extension principle [Bector and Chandra, (2005)] to identify the fuzzy decision, which results in a crisp mathematical programming problem given by
(Problem P2)

Maximize $\alpha$

Subject to

$$\mu_i(T) \geq \alpha, \quad i=1,2,...,n; \quad \alpha \geq 0, \alpha \leq 1, \quad T \geq 0$$

and can be solved by the standard crisp mathematical programming algorithms.

5. While solving the problem following steps 1-4, objective of the problem is also treated as a constraint. In the release time decision problem under consideration each constraint corresponds to one major factor affecting the release time. Hence we can consider each constraint to be an objective for the decision maker and the problem can be looked as a fuzzy multiple objective mathematical programming problem. Further each objective can have different level of importance and can be assigned weight to measure the relative importance. The resulting problem can be solved by the weighted min max approach. The crisp formulation of the weighted problem is given as

(Problem P3)

Maximize $\alpha$

Subject to

$$\mu_i(T) = w_i \alpha, \quad i=1,2,...,n$$

$$\alpha \geq 0, \alpha \leq 1, \quad T \geq 0,$$

$$\sum_{i=1}^{n} w_i = 1$$

where $n$ is the number of constraints in P2 and $\alpha$ represents the degree up to which the aspiration of the decision maker is met. The problem P3 can be solved using standard mathematical programming approach.
If a feasible solution is not obtainable for the problem \((P^*)\) or \((P^{**})\) then we can use fuzzy goal programming approach to obtain a compromised solution [Mohamed, (1997)]. The method is discussed in detail in the numerical example.
Theorem 1.2.1: For a recovery block scheme with $n$ independent versions, the list ordered from smallest to the largest based on failure probabilities is at least as reliable as any other list of the $n$ versions.

Proof: Let $R_n^1$ and $R_n^2$ be the reliability of recovery block respectively for list 1 and list 2 where

List 1 1,2,3,.....$j$, $j+1$,$.....,n$

List 2 1,2,3,.....$j-1$, $j+1$, $j$, $j+2$,$.....,n$

This is sufficient to prove that if $p_1 \leq p_2 \leq p_3 \leq .......p_n$ then $R_n^1 \geq R_n^2$

From equation (12.3.15) $R_n^1$ and $R_n^2$ are given by

$$R_n^1 = P(Y_1) + \sum_{i=1}^{j-2} \left[ \prod_{k=1}^{i} P(X_k) \right] P(Y_{i+1}) + \sum_{i=1}^{j-1} \left[ \prod_{k=1}^{i} P(X_k) \right] P(Y_{i+1}) + \prod_{k=1}^{n} \left[ \prod_{i=j+1}^{i} P(X_i) \right] P(Y_{i+1})$$

...(1.2.1.1)

$$R_n^2 = P(Y_1) + \sum_{i=1}^{j-2} \left[ \prod_{k=1}^{i} P(X_k) \right] P(Y_{i+1}) + \sum_{i=1}^{j-1} \left[ \prod_{k=1}^{i} P(X_k) \right] P(Y_{i+1}) + \prod_{k=1}^{n} \left[ \prod_{i=j+1}^{i} P(X_i) \right] P(Y_{i+1})$$

...(1.2.1.2)

$$R_n^1 - R_n^2 \geq 0 \iff P(Y_j) + P(X_j)P(Y_{j+1}) - P(Y_{j+1}) - P(X_{j+1})P(Y_j) \geq 0$$

...(1.2.1.3)

Substituting the values of $P(Y_j)$, $P(Y_{j+1})$, $P(X_j)$ & $P(X_{j+1})$ we get

$$(1-t_2)(p_{j+1} - p_j) - (1-t_1)(1-t_2)(1-t_3)(p_{j+1} - p_j) \geq 0$$

...(1.2.1.4)

above inequality is true only if $p_{j+1} - p_j \geq 0$, hence $R_n^1 \geq R_n^2$. 


**Theorem 1.2.2** For a consensus recovery block scheme with \( n \) versions, the list of versions ordered from smallest to largest based on failure probability is more reliable than any other list of versions.

**Proof.** Let \( RC_1^n \) and \( RC_2^n \) be the reliability of a consensus recovery block scheme for lists 1 and lists 2 given by:

List 1 \( 1, 2, 3, \ldots, j, j+1, \ldots, n \)

List 2 \( 1, 2, 3, \ldots, j-1, j+1, j, j+2, \ldots, n \)

It is sufficient to prove that, if \( p_1 \leq p_2 \leq \ldots \leq p_n \), then \( RC_1^n \geq RC_2^n \)

From Eq. (1.2.3.8) (refer section (1.2.3), Chapter 2)

\[
RC_1^n = 1 + P(D_1^n) \left[ R_n^1 - 1 \right]
\]

\[
RC_2^n = 1 + P(D_2^n) \left[ R_n^2 - 1 \right]
\]

where \( P(D_i^n) \) and \( R_i^n \) are respectively the probability of failure for an N-version programming and the reliability of a recovery block, given by list \( i (i=1, 2) \)

Consider

\[
RC_1^n - RC_2^n = 1 + P(D_1^n) \left[ R_n^1 - 1 \right] - 1 - P(D_2^n) \left[ R_n^2 - 1 \right]
\]

\[
= P(D_1^n) \left[ R_n^1 - 1 \right] - P(D_2^n) \left[ R_n^2 - 1 \right]
\]

But \( P(D_1^n) = P(D_2^n) \), thus \( RC_1^n \geq RC_2^n \) iff \( R_1^n \geq R_2^n \).

It has already been proved in (Theorem 1.2.1) that \( R_1^n \geq R_2^n \), thus \( RC_1^n \geq RC_2^n \).