CHAPTER 3

TOOLS, TECHNIQUES AND METHODOLOGY

In this chapter tools and techniques are explained which have been used in coming chapters of the dissertation. In this thesis Interpretive Structural Modelling (ISM) is used to identify interrelationship among ASC enablers. The concept of fuzzy logic approach is used to evaluate the agility of supply chain, which is followed by identifying the barriers to agility. Finally Analytical Hierarchy Process (AHP) and Goal Programming (GP) are used to maximize the agility of supply chain. All these four methodologies are briefly explained in this Chapter. Methodologies like AHP, ISM and MICMAC are overused in the literature. The reasons behind using these methodologies are as follows:

- Though these methodologies are overused in the literature but still good choice among researchers due to its intuitiveness, easy applicability, well validated consistency, broad set of application and so on. Number of recent research papers that we can see in the literature using these methodologies are Singhal et al. (2018), Xiao Lin (2018) Avinash et al (2018), Misra et al. (2017), Acharya et al. (2017), Anand and Bansal (2017).
- There are requirements of the problems to use these methodologies which are explained as follows:
    - The major research problems of this thesis are to develop conceptual model for measuring agility (Chapter 5) and to Develop Goal Programming (GP) model (Chapter 7) to maximize the agility. For doing so seven agility enablers have considered. It is desirable to study seven agility enablers to know their interrelationship and to understand their relative importance. Interpretive Structural Modelling and MICMAC analysis are among the best methods to analyse and understand the interrelationship and relative importance of these seven enablers.

- For developing Goal Programming (GP) model, the local and global weights of agility enablers are required; no other methodology can provide better solutions than AHP.

- The focus was on solving the problem not the methodology. Methodologies are the medium to solve the problems. The problems stated in those Chapters (Chapter 4 and Chapter 6) could be solved using those established methodologies, that is why those are used.

3.1 Interpretive Structural Modelling (ISM)

The ISM was first proposed by J. N. Warfield (1976) to analyze the complex socio economic systems (Dewangan et al., 2015). Since then ISM has often been used to help understand complex situations and to enable a strategy for solving problems (Debnath and Shankar, 2012). ISM is an interactive learning process in which a set of dissimilar and directly related elements are structured into a comprehensive systematic model (Govindan et al., 2013). The model so formed portrays the structure of a complex issue, a system of a field of study, in a carefully designed pattern using graphics as well as words (Pandey and Garg, 2009; Agarwal et al., 2007). ISM is interpretive as based on expert's judgment and decision whether and how the system's elements are linked (Digalwar and Giridhar, 2015). Therefore, ISM develops insights into collective understandings of these relationships. The various steps, which are relevant to the development of ISM model, are as follows (Figure 3.1):

3.1.1 Identification of variables relevant to the problem:

First step of the ISM methodology is the identification and finalizing of variables which are relevant to the problem or issues. Identification of variables could be done by literature survey. Literature survey means rigorous review of the literature related to the problem or issues. For literature survey one has to use database of reputed publishers. Identification of variables related to problem is followed by finalization of the variables. Finalization of variables could be done by any group problem-solving technique like Brain Storming, Nominal Group Technique, Delphi Method etc. The problem-solving technique involves thoughts, discussions, actions, and decisions of the experts.

3.1.2 Developing a contextual relationship among variables

From the elements identified in previous section, a contextual relationship is developed among each element with respect to which pairs of elements would be examined. Contextual relationship means how the variables will be related to each other. The contextual relation must be cogently stated as a possible statement of relationship among the elements (Pfohl et al., 2011). Relations may be of several types like comparative, influence, neutral or temporal relations (Warfield, 1994). ISM methodology suggests the use of expert opinions in developing the contextual relationship among the enablers (Pandey and Garg, 20009). The contextual relations among variables are based only on the knowledge and experience of the experts (Mahajan et al., 2014). Therefore, to avoid the subjective biasness of respondent, experts should have sufficient practical experience and knowledge in supply chain domain. Generally four symbols V, A, X and O are used for denoting the directional relationships between all the variables. The rules for using these four symbols to express the relationship between two enablers (say *i* and *j*) are as follows:

- V: element *i* will ameliorate element *j* (relation from *i* to *j* but not in both direction)
- A: element *j* will ameliorate element *i* (relation from *j* to *i* but not in both direction)
- X: element *i* and *j* will ameliorate each other (relation from *i* to *j* and *j* to *i*)
- O: element *i* and *j* are unrelated (no relation).

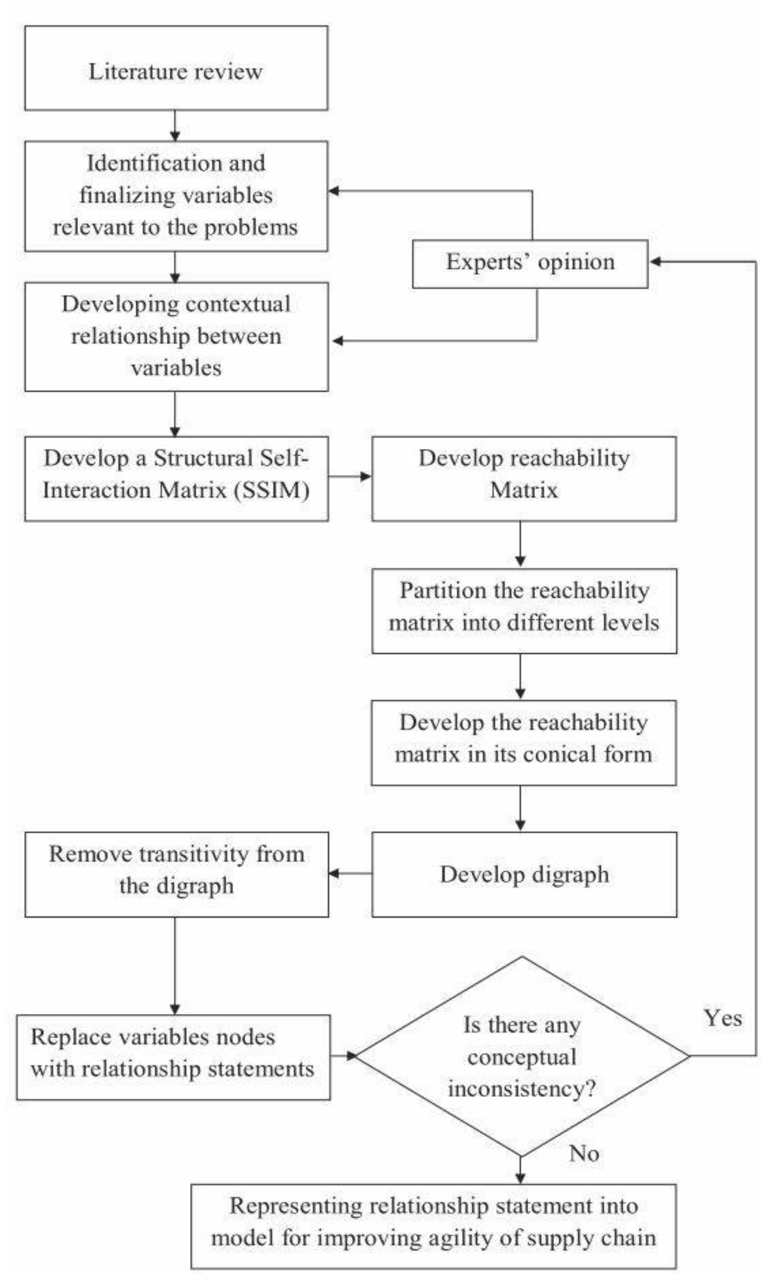

Figure 3.1: Generalized flow diagram of ISM

After resolving the contextual relationship among variables, a Structural Self-Interaction Matrix (SSIM) is prepared in next section.

3.1.3 Developing a Structural Self-Interaction Matrix (SSIM)

Using symbols and rules for directional relationships between all the variables explained in previous section a Structural Self-Interaction Matrix (SSIM) of elements is developed in this section. SSIM of elements indicates pair-wise relationship between elements of the system under consideration.

3.1.4 Developing a reachability matrix

The SSIM is converted into a binary matrix, called the initial reachability matrix. The initial reachability matrix is developed by substituting V, A, X and O by 1 and 0 as per the case. The fundamental rules for the binary conversion are given in Table 3.1 and explained as follows:

- if the (i, j) entry in the SSIM is V, then the (i, j) entry in the reachability matrix becomes 1 and the (j, i) entry becomes 0
- if the (i, j) entry in the SSIM is A, then the (i, j) entry in the matrix becomes 0 and the (j, i) entry becomes 1
- if the (i, j) entry in the SSIM is X, then the (i, j) entry in the matrix becomes 1 and the (j, i) entry also becomes 1
- if the (i, j) entry in the SSIM is O, then the (i, j) entry in the matrix becomes 0 and the (j, i) entry also becomes 0.

Table 3.1: Rules for binary format conversion

Entry in SSIM	**V**	**A**	**X**	**O**
Entry in reachability matrix (i, j)	1	0	1	0
Entry in reachability matrix (j, i)	0	1	1	0

The initial reachability matrix is checked for transitivity. Transitivity is a basic assumption in ISM that leads to the final reachability matrix. To verify the initial reachability matrix for embedded transitivity, a method is used given by Malone (1975), which is again explained by Ojha et al. (2014). The flow diagram of this method is shown in Figure 3.2 and steps of this method are as follows:

- multiply the initial reachability matrix with itself
- replace all values ≥1 with 1
- if the obtained output matrix is the same as the initial matrix, reachability has been achieved. If not, repeat the process on the new output matrix
- if the output matrix, so obtained, is the same as the previous matrix, transitivity has been achieved
- if not, continue the process till the penultimate and final output matrices are identical

After incorporating transitivity final reachability matrix is achieved.

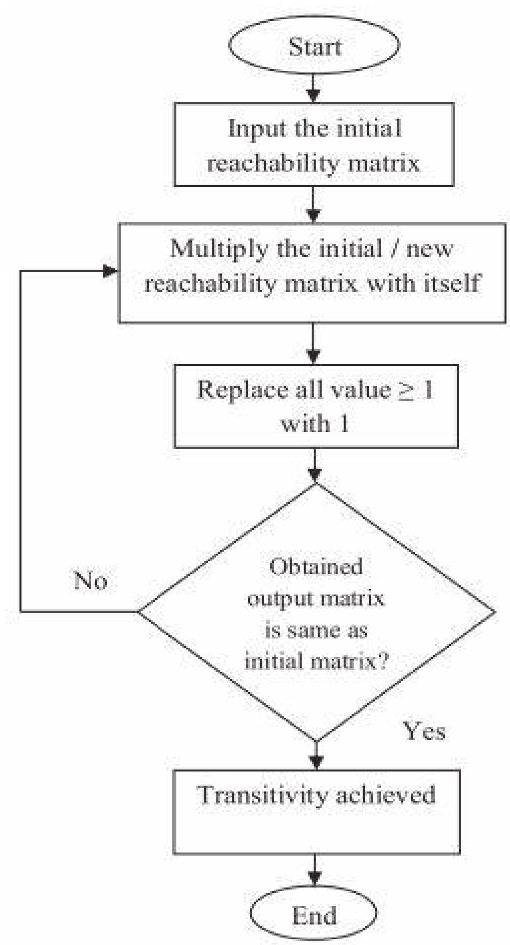

Figure 3.2: Flow chart for incorporating the transitivity

3.1.5 Partitioned reachability matrix into different levels

The final reachability matrix obtained in previous section is partitioned into different levels. In order to partition reachability matrix into different levels it is required to identify the reachability set and antecedent set for enabler. The reachability set and antecedent set for each enabler were derived from the final reachability matrix. The reachability set for a particular enabler consists of the element itself and the other elements which it may impact, whereas the antecedent set consists of the element itself and the other elements which may impact it (Jharkharia and Shankar, 2004). Subsequently the intersection set of reachability set and antecedent set is derived for all the enablers. The enablers, for which the reachability set and the intersection set are the same, are assigned as level I and occupy the top level in the ISM hierarchy. The top-level element in the hierarchy would not help achieve any other element above its own level. Once the top-level element is identified, it is discarded from the other elements and first iteration of level partition is completed. Then, the same process is repeated to find out the elements in the next level. This process is continued until the level of each element is found. These identified levels help in building the digraph and final model.

3.1.6 Developing conical matrix

In this step, conical matrix is developed by clustering enablers in the same level across rows and columns of the final reachability matrix (Agarwal et al., 2007). Developing conical matrix is the step just before building the digraph (Poduval et al., 2015). In this matrix, most zero (0) enablers are written down in the upper diagonal half of the matrix and most unitary (1) enablers are written down in the lower half of the matrix. The conical matrix is similar to the reachability matrix with the exception that the variables in the conical matrix are written on the x- and y-axes based on their levels starting from level.

3.1.7 Formation of ISM-based model

Based on the relationships given above in the reachability matrix, directed graph is drawn by means of vertices or nodes and lines of edges. If there is a relationship between the enablers *i* and *j*, this is shown by an arrow which point from (*i* to *j*); or (*j* to *i*); or be bi-directional (Hasan et al., 2009). A digraph is depicted after removing the transitivity between enablers. In the Digraph, the top level comprises of variables in Level-1 of the level partitions, followed by second level of Level-2 variables and so on. The direction of arrows shows the relationship type. The resultant digraph is finally converted in to ISM-based model by replacing nodes of the factors with statements. The developed ISM model is reviewed to check for conceptual inconsistency and necessary modifications are made.

3.1.8 MICMAC analysis

The MICMAC analysis stands for Matrice d'Impacts Croises Multipication Applique' an Classment (cross-impact matrix multiplication applied to classification), (Yadav and Barve, 2015). The MICMAC principle is used to analyze the driving power and dependence of variables (Mandal and Deshmukh, 1994). Based on the driving power and the dependence, variables can be classified into four categories namely autonomous, dependent, linkage, and independent. The first cluster consists of the autonomous variables that have weak driving power and weak dependence. Second cluster consists of the dependent variables having weak driving power but strong dependence. Third cluster has the linkage variables having both strong driving power as well as dependence. Fourth cluster includes the independent enablers having strong driving power but weak dependence. Further variables can be plotted on driver-dependency diagram based on the four categories. In driver-dependency plot dependency is taken as x-axis and driving power is taken as y-axis (Figure 3.3). Four quadrants are obtained by drawing average driver line and average dependence line (Soti et al., 2010). The basic rules for plotting the variables are tabulated in Table 3.2

Table 3.2: Rules for plotting the variables

S.N.	Categories types	Driving power	Dependence	Position in plot
1.	Autonomous variables	Weak	weak	1st quadrant
2.	Dependent variables	Weak	strong	2nd quadrant
3.	Linkage variables	Strong	strong	3rd quadrant
4.	Independent variables	Strong	weak	4th quadrant

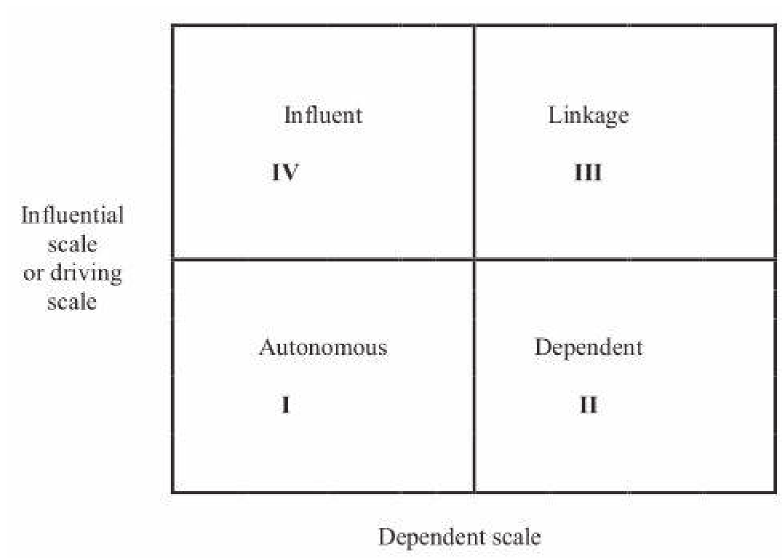

Figure 3.3: Generalized driving power and dependence plot

3.2 Fuzzy Theory and Decision Making

For the purpose of agility assessment in supply chain, the basic properties of fuzzy set theory and decision making are explained in this section. It was Lotfi A. Zadeh who proposed the fuzzy set theory in 1965, after that Bellman and Zadeh (1970) presented the concepts of decision-making in such a fuzzy environment. Since then, a lot of theoretical developments have taken place in this field. Decisions which are largely dependent on estimation of the experts may be imprecise and ambiguous. To deal with vagueness in human thought, fuzzy theory can be used. Fuzzy set theory is an excellent mathematical tool to handle the uncertainty arising due to vagueness. Fuzzy theory provides formalized tools to deal with the imprecision inherent to decision making problems. Fuzzy logic is currently being used extensively in many industrial applications such as water treatment, travel time reduction, subway systems, washing machines, vacuum cleaners, rice cookers and aircraft flight control (Yen and Langari, 1999).

3.2.1 Fuzzy versus Crisp

Consider the query, 'is 'X' honest?' The answer to this is a definite *Yes* or *No*. If '*yes*' is accorded a value of *1* and '*no*' is accorded a value of *0*, this statement results in a *0*/*1* type situation. Such a logic which demands binary types of handling is termed as a *crisp* in the fuzzy logic theory. Answer of the above query can also be given considering a variety of answers spanning a range, such as 'extremely honest', 'very honest', 'honest at times' and 'extremely dishonest'. If, for instance, 'extremely honest' were to be accorded a value of *1*, at the high end of the spectrum of values, 'extremely dishonest' a value of *0* at the low end of the spectrum, then 'very honest' and 'honest at times' could be assigned values of 0.85 and 0.40 respectively (Rajasekaran and Pai, 2012). Such a situation is called as fuzzy. Figure 3.4 shows the above explanation in the form of schematic diagram.

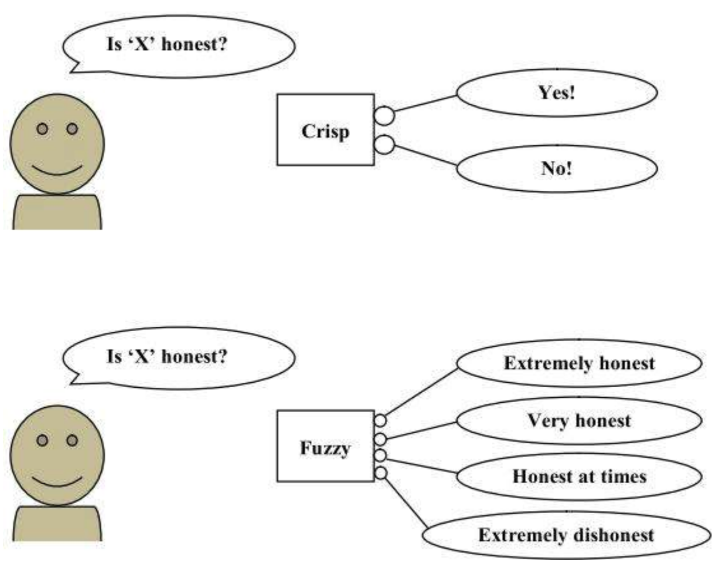

Figure 3.4: Fuzzy versus crisp

3.2.2 Definition of Fuzzy Sets

A fuzzy set (class) ***A*** in ***X*** is characterized by a membership (characteristic) function $f_A(x)$ which associates with each point in ***X*** a real number in the interval **[0,1]** (Zadeh**, 1965). The function** values of $f_A(x)$ represents the grade of membership of ***x*** in **A**. Thus, as the value of $f_A(x)$ approaches to unity, the grade of membership of *x* in ***A*** **tends** the higher.

3.2.3 Definition of Fuzzy Numbers

A fuzzy subset *A* is called a fuzzy number if *A* is convex and there exists exactly one real number '*a*' with $f_A(a)$ = 1. There are many forms of fuzzy numbers (such as triangular, trapezoidal, curved etc.) to represent imprecise information. For the purpose of calculation of agility in chapter 5, triangular fuzzy numbers are applied. Therefore, only concept of triangular fuzzy number is explained.

3.2.4 The concept of Generalized Triangular Fuzzy Number

Suppose, a positive triangular fuzzy number is '*A*' and that can be defined as (*a, b, c*) shown in Figure 3.4. The membership function $f_A(x)$ of triangular fuzzy number, which is associated with a real number in the interval [0, 1] can be defined as:

$$f_A(x) = \begin{cases} (x-a)/(b-a), & a \le x \le b, \\ (x-c)/(c-b), & b \le x \le c, \\ 0, & otherwise, \end{cases} \quad (3.1)$$

The triangular fuzzy number is parameterized by the triplet *A* = (*a*, *b*, *c*). The parameter '*b*' gives the maximal grade of $f_A(x)$ i.e. $f_A(b)$ =1, which is the most probable value of the evaluation data. The parameters '*a*' and '*c*' are the lower and upper bounds of the available area for the evaluation data.

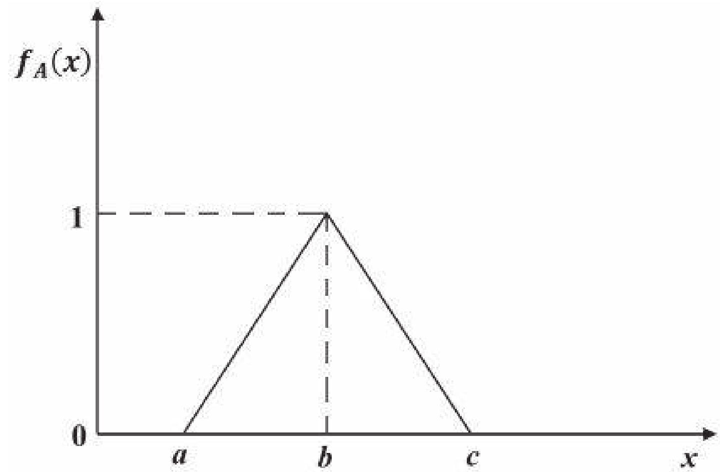

Figure 3.5: A triangular fuzzy number 'A'

3.2.5 Linguistic Variables

As discussed earlier, there may be ambiguity and impreciseness in human thoughts. To deal with ambiguity and impreciseness of human thoughts linguistic terms can be used. A linguistic variable is the variable whose values are words or sentences in natural or artificial language (Tseng and Lin, 2011). For example 'good' and 'high' are linguistic variables. Linguistic expressions are so vague and converting them into numerical value may prove to be difficult (Vinodh et al., 2013). The field of artificial intelligence offers a solution to face these challenges by offering 'fuzzy logic' methodology. Furthermore, by the approximate reasoning of fuzzy sets theory, the linguistic value can be represented by a fuzzy number. Therefore, in order to calculate agility in supply chain the linguistic terms and the corresponding fuzzy numbers are shown in Table 3.3.

3.2.6 Fuzzy number arithmetic operations

Suppose that A_1 and A_2 are two generalized triangular fuzzy numbers, where $A_1 = (a_1, b_1, c_1)$ and $A_2 = (a_2, b_2, c_2)$, then the triangular fuzzy-number addition, subtraction and multiplication operations of A_1 and A_2 are shown as follows:

Table 3.3: Linguistic terms and fuzzy numbers

Performance-rating (R)			Importance-weighting (W)		
Linguistic variable	**Notation**	**Fuzzy number**	**Linguistic variable**	**Notation**	**Fuzzy number**
Worst	W	(0, 0.5, 1.5)	Very Low	VL	(0, 0.05, 0.15)
Very Poor	VP	(1, 2, 3)	Low	L	(0.1, 0.2, 0.3)
Poor	P	(2, 3.5, 5)	Fairly Low	FL	(0.2, 0.35, 0.5)
Fair	F	(3, 5, 7)	Medium	M	(0.3, 0.5, 0.7)
Good	G	(5, 6.5, 8)	Fairly High	FH	(0.5, 0.65, 0.8)
Very Good	VG	(7, 8, 9)	High	H	(0.7, 0.8, 0.9)
Excellent	E	(8.5, 9.5, 10)	Very High	VH	(0.85, 0.95, 1.0)

3.2.6.1 Fuzzy number addition

$A_1 \oplus A_2 = (a_1, b_1, c_1) \oplus (a_2, b_2, c_2)$

$= (a_1 + a_2, b_1 + b_2, c_1 + c_2)$, Where $\oplus$ addition operator

3.2.6.2 Fuzzy number subtraction

$A_1 \ominus A_2 = (a_1, b_1, c_1) \ominus (a_2, b_2, c_2)$

$= (a_1 - a_2, b_1 - b_2, c_1 - c_2)$, Where $\ominus$ subtraction operator

3.2.6.3 Fuzzy number multiplication

$A_1 \otimes A_2 = (a_1, b_1, c_1) \otimes (a_2, b_2, c_2)$

$= (a_1 * a_2, b_1 * b_2, c_1 * c_2)$, Where $\otimes$ multiplication operator

3.2.7 Euclidean distance method

The Euclidean distance method consists of calculating the Euclidean distance from the given specific fuzzy number to each of the fuzzy numbers representing the natural-language expressions set. Consider a specific fuzzy number represented by ‘fuzzy agility index' (FAI) and consider fuzzy number representing the natural-language expression set is denoted by 'Agility Level' (AL) then Euclidean distance *D* between FAI and AL was calculated by following formula (Vinodh and Devadasan, 2011). Fuzzy number representing the natural-language expression set is tabulated in Table 3.4 (Lin et al., 2006b).

$$D(\mathrm{FAI},\mathrm{AL}) = \left\{\sum_{x\in p}[f_{FAI}(x) - f_{AL}(x)]^2\right\}^{1/2} \tag{3.2}$$

Table 3.4: Natural-language expression set for labelling the agility level

Symbol	**Linguistic terms**	**Fuzzy number**
EA	Extremely Agile	(7, 8.5, 10)
VA	Very Agile	(5.5,7,8.5)
A	Agile	(3.5,5,6.5)
FA	Fairly Agile	(1.5,3,4.5)
SA	Slowly Becoming Agile	(0,1.5,3)

3.2.8 Ranking of Generalized Triangular Fuzzy Number

Ranking of the fuzzy number is also called defuzzification. In many situations, it is easier to take a crisp decision if the output is represented as a single scalar quantity. Therefore, ranking of the fuzzy numbers is important. There are many methods available in the literature to rank the fuzzy numbers such as centroid method, centre of sum method and mean of maxima method. In present dissertation centroid method is used to rank the fuzzy numbers. The reason behind this is that centroid method is simple and easy to implement (Vinodh et al., 2013). Using centroid method ranking score can be calculated by equation (3.3) (Vinodh and Vimal, 2012). Ranking of fuzzy number is used to identify the Fuzzy

Performance Importance Index (FPII) which is further used to identify the principal obstacles of agility in supply chain.

$$\text{Ranking score} = \frac{(a + 4b + c)}{6} \qquad (3.3)$$

Where

$a =$ Lower number of triangular fuzzy number

$b =$ Middle number of triangular fuzzy number

$c =$ Upper number of triangular fuzzy number

3.3 Multi Criteria Decision Making (MCDM)

Multiple Criteria Decision Making (MCDM) method is generally used to make decision in the presence of multiple, generally conflicting criteria. Decision making is a difficult task when multiple and conflicting criteria are supposed to be considered. For example, a decision maker wants to buy a car which has maximum fuel efficiency, while at lowest cost. In this situation MCDM is most suitable tool to make correct decision among the available choices.

3.3.1 Analytical Hierarchy Process (AHP) as MCDM

There are many methods available in literature to solve multiple and conflicting criteria decision making problem. However, Analytic Hierarchy Process (AHP) is the most popular and widely accepted decision making technique among all of them. AHP is developed by Thomas L. Saaty in 1980. From then it is thoroughly tested by thousands of organizations around the world (Opydo, 2013). AHP is still a good choice as MCDM tools even though it is 35 years old decision making method. It is due to reason of its intuitiveness and easy applicability, well validated consistency, broad set of application and so on. AHP has been used in almost all the applications related with decision-making (Vaidya and Kumar, 2006).

3.3.2 Steps involved in AHP

The steps involved in decision making by using the AHP method are as follows:

Step 1: First step of the AHP methodology is to determine the objective of the problem, selection criteria, and alternatives.

Step 2: After determining the objective, criteria and alternative, structure the problem into a hierarchy. Hierarchy of the problem consists of goal at the highest level, the criteria and sub-criteria at lower levels and the alternatives at the lowest level.

Step 3: Next step is the preparation of questionnaire and collection of data for pair-wise comparisons of criteria with respect to goal or objective and pair-wise comparisons of alternatives with respect to each of the criterion.

Step 4: Determining priority weights for each criterion and checking the consistency of judgements by calculating consistency ratio for the comparison of criteria.

Step 5: Calculating priority weights of each alternative with respect to each criterion and calculate the consistency ratio for the each pair-wise comparison of alternatives. Priority weight of each alternative with respect to each criterion is called local weight of the alternatives.

Step 6: Once the local weights of each alternative are obtained, they are aggregated to find global weight of the each alternative by multiplying the priority weight of decision alternatives to priority weight of selection criteria and summing over all criteria.

Step 7: Finally prioritization of variables will be obtained. Alternative with highest priority weights considerably more desirable followed by next higher priority weight and so on.

3.3.3 Basic principles of AHP

Saaty (1980) stated that there are three basic principles in the AHP method, which are as follows:

i. Hierarchy construction

After the problem has been defined, decomposition is necessary to be done, which is dividing a problem into some smaller parts. The division process will result in some levels of a problem. This is why this process of analysis is named hierarchy construction. The AHP model, structured in a hierarchy of three basic levels is shown in Figure 3.6. The top level of the hierarchy contains the goal of the problem, followed by the selection criteria at second level and finally, the third level lists the alternatives.

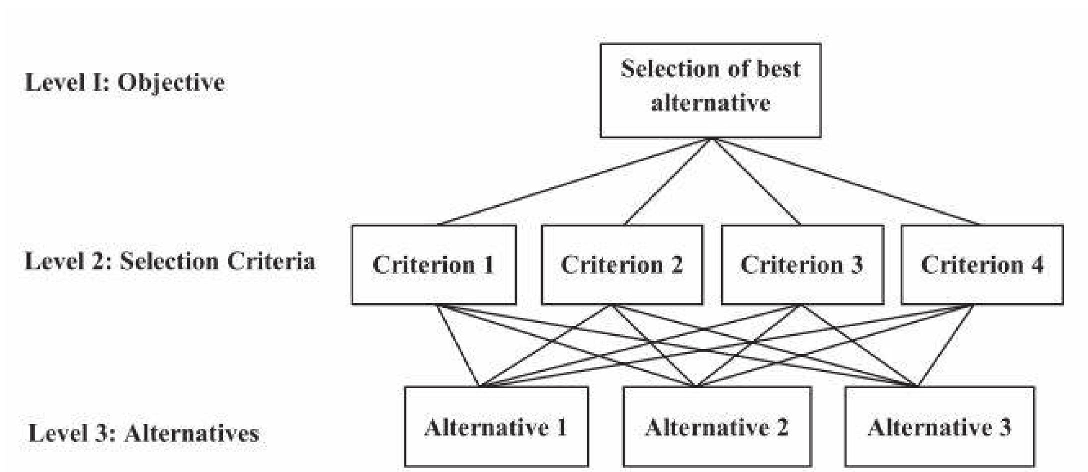

Figure 3.6: A simple hierarchy construction of AHP

ii. Comparative Judgment

Once the model is built, the decision-maker evaluates the elements by making pair-wise comparisons. A pair-wise comparison is the process of comparing the relative importance of criteria with respect to goal as well as relative importance of alternatives with respect to each of the criterion. The pair-wise comparison is established using a nine-point Saaty scale which is shown in Table 3.5. This scale indicates how many times more important or dominant one element is over another element with respect to the criterion or property with respect to which they are compared.

Table 3.5: Thomas Saaty's nine-point scale for making the judgement

Intensity of importance	**Definition**	**Explanation**
1	Equal Importance	Two activities contribute equally to the objective
3	Moderate importance of one over another	Experience and judgement slightly favour one activity over another
5	Strong importance of one over another	Experience and judgement strongly favour one activity over another
7	Very strong importance of one over another	An activity is favoured very strongly over another; its dominance demonstrated in practice
9	Extreme importance of one over another	The evidence favouring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate value between two adjacent judgments	When compromise is needed between the any two of the value listed above
Reciprocals of above	If activity i has one of the above non-zero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i	A reasonable assumption

iii. Priority analysis

When all the comparisons are completed, decision maker has to calculate the priority weights of each pair-wise comparison matrix. To calculate the priority weight the pair-wise comparison matrix is normalized by dividing the elements of each column by the sum of the corresponding column. Then, the average of each row will give the corresponding priority vector or priority weight.

3.3.4 Test of Consistency

The next stage is to calculate a Consistency Ratio (C_R) to measure how consistent the judgements have been relative to large samples of purely random judgements. The consistency ratio is an approximate mathematical indicator which provides consistency of pair-wise comparisons (Canada and Sullivan, 1989). Consistency ratio for the comparison matrix should be within a 0.10, which is the empirical upper limit suggested by Saaty. If the consistency ratio is greater than 0.10 the judgements are untrustworthy and the pair wise analysis must be repeated for consistency. In general, the lower the consistency ratio gives the more accurate priority weights. Mathematically consistency ratio can be expressed by the ratio of the consistency index to the random index, which is shown by equation (3.4).

$$C_R = \frac{CI}{RI} \tag{3.4}$$

Where C_R = Consistency Ratio, CI = Consistency Index and RI = random index

Consistency Index for a matrix size '*n*' is given by the following formula.

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{3.5}$$

Where λ_{max} = Maximum eigenvalue and n = size of the matrix

Random Index (RI) can be obtained from simulation runs and depends upon the order of matrix. Table 3.6 shows the average values of RI for the matrices of order 1-10 (Saaty, 1980).

Table 3.6: Random index (RI) based on matrix order (*n*)

Size of matrix (*n*)	1	2	3	4	5	6	7	8	9	10
Random Index (*RI*)	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Maximum eigenvalue (λ_{max}) is calculated in order to obtain C_R. To calculate λ_{max} first multiply the matrix of pair-wise comparisons (call it matrix A) by priority weight (call it matrix B) to get new matrix C (equation 3.6). In next step divide each element in vector [C] by its corresponding element in vector [B] to find a new vector [D]. Now, average the elements in vector [D], which can be called as maximum eigenvalue and denoted by λ_{max}. Applying these expressions, consistency ratio of the each of the comparison matrices can be calculated.

$$[A] * [B] = [C] \tag{3.6}$$

$$[D] = \frac{[C]}{[B]} \tag{3.7}$$

3.4 Goal Programming (GP)

Goal Programming (GP) involves solving problems containing a collection of goals rather than single specific objective function. In GP, the objective function attempts to minimize the unwanted deviations of all the goal constraints. The basic approach of goal programming is to establish a specific numeric goal for each of the objectives, formulate an objective function for each objective, and then seek a solution that minimizes the (weighted) sum of deviations of these objective functions from their respective goals. According to Hillier and Lieberman (2005), there are three possible types of goals:

1. A lower, one-sided goal sets a lower limit that we do not want to fall under (but exceeding the limit is fine).
2. An upper, one-sided goal sets an upper limit that we do not want to exceed (but falling under the limit is fine).
3. A two-sided goal sets a specific target that we do not want to miss on either side.

3.4.1 Terminology and Concepts

Prior to further discussion of GP, it is desirable to describe the basic terminology and concepts that develop this technique:

3.4.1.1 Decision maker(s): The decision maker(s) refer to the person(s), organization(s), or stakeholder(s) to whom the decision problem under consideration belongs.

3.4.1.2 Decision variable: A decision variable is defined as a factor over which the decision maker has control.

3.4.1.3 Deviation variable: The difference between what is accomplished and the targeted goal is referred to as the deviation variable. Deviation can be expressed in terms of underachievement (negative deviation) and/ or overachievement (positive deviation). d_k^- represents negative deviation variables; representing the part that decision-making value is not reached the target whereas d_k^+ represents positive deviation variables; representing the part that decision value exceeds the target value.

In practice, when the target is determined, there are three possibilities of deviations:

i. Decision value exceeds the target value, expressed as $d_k^+ \geq 0$, $d_k^- = 0$.
ii. Decision value does not reach the target value, expressed as $d_k^+ = 0$, $d_i^- \geq 0$.
iii. Decision value equals to the target value, expressed as $d_k^+ = 0$, $d_k^- = 0$.

3.4.1.4 Objective function: In order to get the satisfactory solution which satisfies the system constraints and objective constraints, from the perspective of decision makers, determining their advantages and disadvantages should be based on the calculated value of the target deviation. Mathematical expression for the objective function of the goal programming model is:

$$\min z = f(d_k^+, d_k^-) \qquad (3.8)$$

3.4.1.5 Constraint: A constraint is a restriction upon the decision variables that must be satisfied in order for the solution to be implementable in practice. This is distinct from the concept of a goal whose non-achievement does not automatically make the solution non-implementable. A constraint is normally a function of several decision variables and can be equality or an inequality.

3.4.2 General Structure of the Goal Programming Methodology

GP minimizes the achievement function $f(d_k^-, d_k^+)$ one term at a time in the order of any stated preemptive priorities (Canada and Sullivan, 1989). The function f_k is a linear function of deviational variables. For example $f = d_k^-$, $f = d_k^+$, $f = (d_k^- + d_k^+)$. The model can be described in general notation as follows:

Minimize

$$Z = f(d_k^-, d_k^+) \tag{3.9}$$

Subject to

$$\sum_{i=1}^{n} c_{ki} x_i + d_k^- - d_k^+ = b_k \qquad for\ k = 1, 2, \dots, m \tag{3.10}$$

$$x_i, d_k^-, d_k^+ \geq 0 \tag{3.11}$$

Where

x_i = decision variable *i*.

d_k^- = negative deviation (underachievement) for constraint *k*.

d_k^+ = positive deviation (overachievement) for constraint *k*.

b_k = targeted goal or 'right hand side' value for constraint *k*.

c_{ki} = technological coefficient associated with variable *i* of constraint *k*.

m = total number of constraints.

For each of the objectives, a target value or goal would be given (b_k), which is needed to be achieved. Finally, the undesired deviations d = (d_k^+, d_k^-) from the given set of targets (b_k) are minimized by using an achievement function (Z). In effect, a deviational variable represents the deviation between the targeted level and the actual attainment of the goal.

Hence, the deviation variable d is replaced by two variables: $d = (d_k^- - d_k^+)$ where $d_k^-, d_k^+ \geq$. The preceding ensures that the deviational variables never take on negative values. The constraint ensures that one of the deviation variables will always be zero. Finally, the unwanted deviational variables need to be brought together in the form of an achievement function whose purpose is to minimize them and thus ensure that a solution that is "as close as possible" to the set of desired goals is found. This solution is called a compromised (harmonized) solution rather than optimal and that is why it is called a satisfying technique (Orumie and Ebong, 2014).

3.4.3 Categorization of the GP model

Goal programming problems can be categorized according to the type of mathematical programming model (linear programming, integer programming, nonlinear programming, etc.) that it fits except for having multiple goals instead of a single objective (Hillier and Lieberman, 2005). In this dissertation linear goal programming is used. Another categorization is according to how the goals compare in importance which are explained as follows:

3.4.3.1 Lexicographic (Pre-emptive) Goal Programming Model

In some situations, decision maker is not able to determine precisely the relative importance of the goals; in this condition pre-emptive goal programming has to be used. In pre-emptive goal programming, the decision maker must rank each goal from the most important (first goal) to least important (last goal). Pre-emptive goal programming procedure starts by concentrating on meeting the most important goal as closely as possible, before proceeding to the next higher goal. *i.e.* the objective functions are prioritized such that attainment of first goal is far more important than attainment of second goal, which is far more important than attainment of third goal on so on. It means that lower order goals are only achieved as long as they do not degrade the solution attained by higher priority goal.

3.4.3.2 Weighted Goal Programming Model

If the decision maker is more interested in direct comparisons of the objectives then Weighted Goal Programming (WGP) should be used. In weighted goal programming, weights are attached to each of the objectives to measure the relative importance of deviations from their target. The relative weights may be any real number, where greater the weight, greater the assigned importance to minimize the respective deviation variable to which the relative weight is attached.

3.4.4 Advantages of Goal Programming Model

Specifically, there are some advantages of GP model, which are (Canada and Sullivan, 1989):

- The model development and implementation are relatively simple, flexible, efficient and straightforward.
- The model and its assumptions seem consistent with typical real-world problems.
- It has the capability of handling decision problems with a single goal having multiple sub-goals. A single goal can be achieved by collectively accomplishing a set of sub-goals.
- The objective function of a goal programming model may contain non-homogeneous units of measurement.