ISSN No:-2456-2165

# An Integrated Method for the Selection of Software Requirements Using Fuzzy TOPSIS and Genetic Algorithm

Mohd. Nazim, Tanveer Hassan,
Chaudhary Wali Mohammad
Department of Applied Sciences and Humanities,
Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi, India

Mohd. Sadiq
Computer Engineering Section,
UPFET, Jamia Millia Islamia, New Delhi, India

**Abstract:- Software requirements selection is one of the key activities of the software development process. In this activity the requirements are selected based on their ranking order. Various methods have been developed for selecting the requirements using fuzzy logic, rough set theory, and Metaheuristic algorithms, etc. One of the limitations of the fuzzy based methods is that the membership functions of fuzzy numbers are manually decided by the decision makers. In these methods less attention is given to the automated generation of fuzzy membership function. To address this issue, this paper presents a method for the selection of software requirements in which genetic algorithm has been used for automatically generating the fuzzy numbers. In the proposed method, a random population of size 15 is initialized then the process of reproduction is started by using the selection, crossovers, and mutation operators of genetic algorithm for 23 generations. The best chromosome with the fitness value of 0.883333333 is selected from the last generation as an input in fuzzy TOPSIS method. The applicability of the proposed method is discussed by the requirements of an institute examination system.**

***Keywords:-*** *Functional requirements; Fuzzy TOPSIS; Genetic algorithm; Fitness value; Crossover; Mutation; Population; Institute Examination System.*

## I. INTRODUCTION

Software requirements (SRs) are the expectations of the stakeholders which are identified by various requirements elicitation techniques like traditional methods, group elicitation, goal-oriented method, etc. The software requirements are classified into functional requirements (FRs) and non-functional requirements (NFRs) [1]. The FRs describes the functionality of software. For example, download the hall ticket of a student from an examination system of an institute. The NFRs describe the non-behavioural aspect of a system, for example, the system should be less expensive and secure [2, 3]. SRs selection is a multicriteria decision making method in which various stakeholders participate during the selection process [4]. It has been observed that a system may have thousands of requirements after the completion of requirements elicitation process. It is not feasible to implement all the requirements because of the budget, time, and other constraints of an organization [5- 8].

Various methods have been developed for selecting the requirements using soft computing techniques using fuzzy logic, rough set theory, etc. In fuzzy based methods, triangular fuzzy numbers (TFNs) are mostly used during the computational process because of its simplicity in understanding and representation. The rough-set theory based approach was developed by Sadiq and Devi [10] for the selection of requirements. In [10] the authors divided the SRs selection methods into manual and automated methods. In our previous work [11], we compared two MCDM methods, i.e., fuzzy “analytic hierarchy process” (AHP) and fuzzy “technique for order of preference by similarity to ideal solutions” (TOPSIS) methods for the selection of SRs of both small and large dataset. One of the limitations of these methods was that the membership functions are selected manually during the computationally process. To address this issue, in this paper we used the genetic algorithm during the computational process of requirements selection of software. The contributions of our work are as follows:

- A genetic algorithm based fuzzy TOPSIS method has been developed for the selection of SRs. In the proposed method genetic algorithm performed automated generation of the fuzzy membership function set. In this set the linguistic variables are used by the decision makers during the evaluation of requirements based on different criteria
- The applicability of the proposed method is discussed by using the requirements of an institute examination system (IES)
- The proposed method is compared with fuzzy TOPSIS method based on the ranking order of the requirements of an IES

This paper is organized as follows: The related work in the field of the SRs selection is presented in Section II. Section III provides an overview of the GA algorithm. The proposed method for the SRs selection using genetic algorithm based fuzzy TOPSIS method in Section IV. The experimental results are shown in Section V, and Section VI provides the conclusion and the future work.

## II. RELATED WORK

A large number of methods have been developed in the area of SRs selection using different concepts like fuzzy logic, crisp logic, rough set theory, Metaheuristic algorithms, etc. For example, Ijaz et al. [12] proposed a value-based fuzzy approach for the prioritization of FRs and NFRs. Singh et al. [13] proposed an MCDM technique in which logarithmic fuzzy trapezoidal AHP (LFTAHP) has been employed for SRs prioritization. Mougouei and Powers [15] proposed an expert system based on fuzzy graph and integer programming named "Dependency-Aware Requirements Selection (DARS)". In another study, Mougouei and Powers [16] proposed a "Prioritization And Partial Selection (PAPS)" method for reducing the number of security requirements that are ignored during the development process. Hassan and Ramadan [17] proposed a fuzzy-based hybrid combination of quality function deployment, Cumulative Voting and AHP to enhance the SRs prioritization and selection process. Singh et al. [18] proposed an "ANN fuzzy AHP model" for the prioritization of requirements. Mougouei et al. [19] proposed a fuzzy-based optimization technique for the SRs selection. A "fuzzy inference system" based framework was developed by Alrashoud and Abhari [20] to perform the selection of SRs for the software next release problem. An "adaptive fuzzy hierarchical cumulative voting (HCV)" was developed by Jawale and Bhole [21] for the prioritization of requirements. Afrin and Sadiq [22] proposed an integrated approach by using fuzzy AHP and TOPSIS methods for SRs selection. Ahmad et al. [23] proposed a "fuzzy based MoSCoW" method for computing the ranking order of requirements. Gerogiannis and Tzikas [24] used the "Fuzzy Linguistic 2-Tuples (F2TL)" method for the SRs prioritization based on the evaluations of stakeholders. Sadiq et al. [25] developed a method for eliciting the security requirements in which fuzzy logic was used for selecting those set of requirements which need more security during the development process. Achimugu et al. [26] proposed an "adaptive fuzzy decision matrix model (AFDMM)" for the prioritization of SRs based on the weights given by different stakeholders. Sharif et al. [27] developed a method in which HCV was used under fuzzy environment. In this method, the fuzzy expert system was also used for analysing the requirements. Momeni et al. [28] proposed an approach for the prioritization of software quality requirements using Neuro-fuzzy system. The SRs selection has also been used as one of the steps of various methods like "attributed goal oriented requirements analysis" (AGORA), "fuzzy attributed goal oriented software requirements analysis" (FAGOSRA), etc. For example, Sadiq and Jain [29] developed a method to strengthen the goal oriented requirements engineering in which fuzzy logic was used for selecting and prioritizing the requirements of an IES. The fuzzy TOPSIS method was used in [14] for the selection of the requirements of an information system. In this method the membership function for the fuzzy numbers was selected manually. There was no automated method for the generation of fuzzy numbers. In this paper an attempt has been made for selecting the requirements of an IES using fuzz TOPSIS in which genetic algorithm has been applied for generating the fuzzy numbers that will be used as input during the decision making process.

## III. GENETIC ALGORITHM

Genetic algorithm (GA) is one of the key components of soft computing which is based on Darwin's evolution theory, i.e., survival of the fittest species [30]. GA has been used to solve various issues in different fields like operations management, multimedia, wireless networking, etc. [31]. The GA is inspired by three main concepts of biology, i.e., selection, reproduction, and mutation. It is commonly used as a search method to find out the optimal solution of a problem from a set of solutions [31]. This set of solutions is termed as population. To get the optimal solution, the different possible solutions, i.e., chromosomes are used. The chromosomes are used to form a generation, i.e. initial population, and the next generations are formed by selecting the fittest chromosomes on the basis of their fitness. The GA's success depends on the computation of fitness function, i.e., a chromosome with higher fitness has a higher chance of evolving into the next generation. The size of the population is the main parameter because it directly affects the capacity to find an optimal solution in the search space.

There are four basic operators of a GA namely encoding techniques, selection/reproduction, crossover, and mutation. Encoding is used to convert given information into a certain bit string. Binary, octal, hexadecimal, permutation, value-based, and tree are all well-known encoding techniques. Binary encoding is the most prevalent encoding technique in which each chromosome is denoted by a string of 0s and 1s [31]. Selection/Reproduction operator is a significant step in GA which is used to produce the present population's individual to the next generation in accordance to their fitness. The roulette wheel, stochastic universal sampling, rank, Boltzmann, and tournament are some of the most popular selection techniques. Crossover operator is used to generate a new individual by combining the genetic value of two or more than two chromosomes together on the basis of some criteria. The single point, two-point, k-point, shuffle, etc. are some of the most popular crossover operators. Mutation operator is used to maintain the genetic diversity from population to population. Some of the most popular mutation operators are simple inversion, scramble mutation, and displacement. In our work, we have used binary encoding and single point crossover during the software requirements selection process.

## IV. PROPOSED METHOD

This section presents a method for computing the ranking order of SRs so that it can be selected during the development process. The block diagram of the proposed method is given in Fig. 1. Genetic algorithm has been used with fuzzy TOPSIS method in which optimal value of fuzzy membership function is generated. The following are the major steps in the proposed method:

- **Step 1:**Identification of stakeholders and their requirements
- **Step 2:**Determine the various linguistic variables for the rating purpose

- **Step 3:**Decision makers provide their ratings for SRs
- **Step 4:**Specify a population size (N)
- **Step 5:**Set a target chromosome by using the corresponding values of LVs generated through some intuitionist approach
- **Step 6:**Encode the target chromosome into binary form
- **Step 7:**Generate an initial population (P)
- **Step 8:**Check, is the target achieved?, (i) If yes, go to step-15, If no, go to step-9,
- **Step 9:**Compute the fitness value of each chromosome of the population,
- **Step 10:**Select two parent chromosomes based on their fitness values.

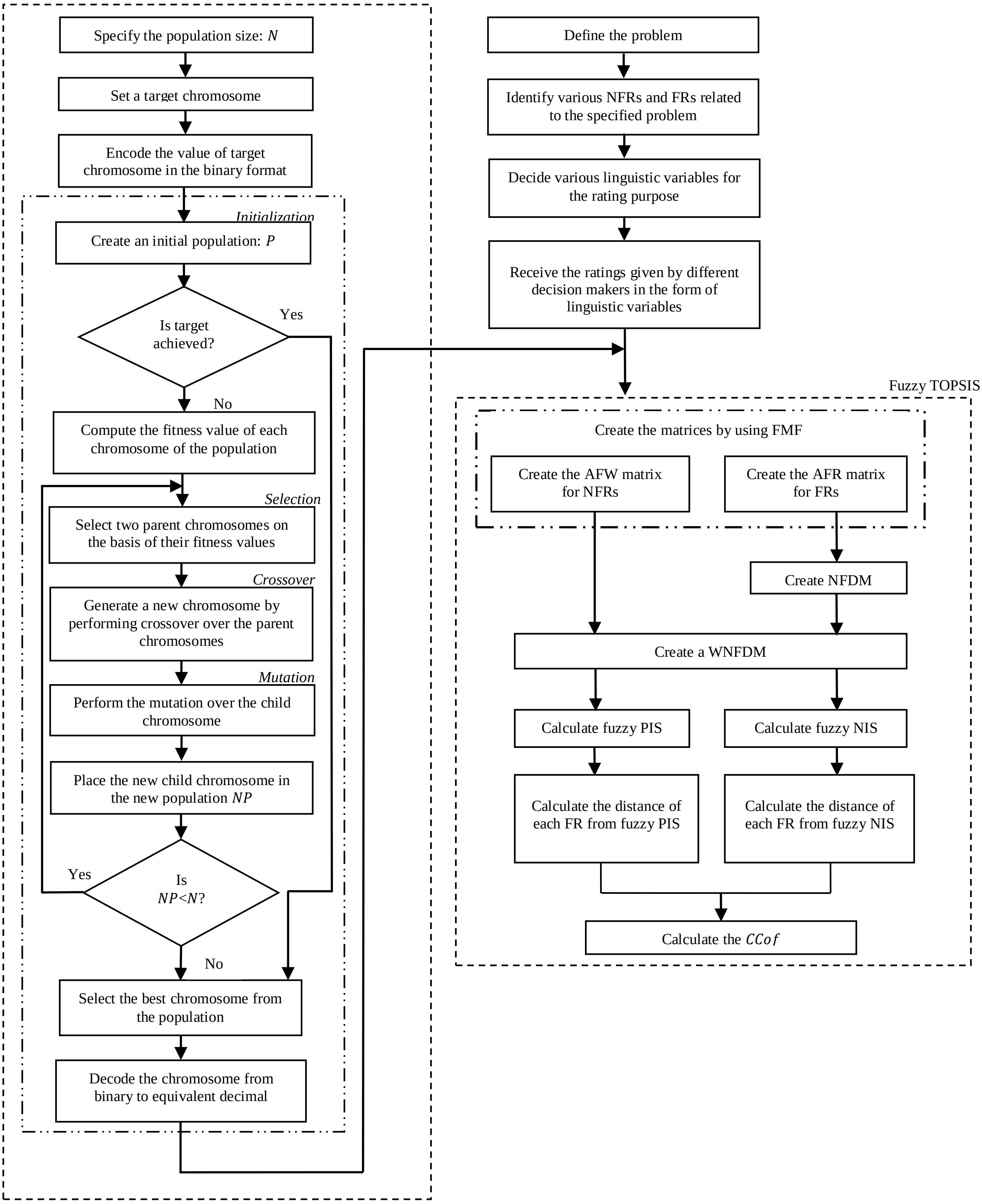
Specify the population size: $N$
Set a target chromosome
Encode the value of target chromosome in the binary format
Initialization
Create an initial population: $P$
Is target achieved?
Yes
No
Compute the fitness value of each chromosome of the population
Selection
Select two parent chromosomes on the basis of their fitness values
Crossover
Generate a new chromosome by performing crossover over the parent chromosomes
Mutation
Perform the mutation over the child chromosome
Place the new child chromosome in the new population $NP$
Yes
Is $NP<N$?
No
Select the best chromosome from the population
Decode the chromosome from binary to equivalent decimal
Define the problem
Identify various NFRs and FRs related to the specified problem
Decide various linguistic variables for the rating purpose
Receive the ratings given by different decision makers in the form of linguistic variables
Fuzzy TOPSIS
Create the matrices by using FMF
Create the AFW matrix for NFRs
Create the AFR matrix for FRs
Create NFDM
Create a WNFDM
Calculate fuzzy PIS
Calculate fuzzy NIS
Calculate the distance of each FR from fuzzy PIS
Calculate the distance of each FR from fuzzy NIS
Calculate the $CCof$

Fig. 1: Block diagram of the proposed method

- **Step 11:**Crossover these parent chromosomes based on their PC to generate a new child chromosome
- **Step 12:**Perform the mutation over the child chromosome based on PM,
- **Step 13:**Place this new child chromosome in the new population (NP),
- **Step 14:**Check, is the population limit reached?, If NO, go to step-10, If yes, go to step-8,
- **Step 15:**Select the best chromosome from the NP to be used as the FMF and decode it from binary to decimal values,
- **Step 16:**Calculate the aggregated fuzzy weight (AFW) for the NFRs and aggregated fuzzy rating (AFR) for the FRs by using the fuzzy membership function (FMF)
- **Step 17:**Calculate the normalized fuzzy decision matrix (NFDM) and then calculate the weighted NFDM (WNFDM) by using NFDM and AFW
- **Step 18:**Calculate the fuzzy positive ideal solution (PIS) and fuzzy negative ideal solution (NIS) by using WNFDM,
- **Step 19:**Calculate the distance of each FR from fuzzy PIS and fuzzy NIS,
- **Step 20:**Calculate the closeness coefficient by using the fuzzy PIS and fuzzy NIS.

- **Step 1:Identification of stakeholders and their requirements**
  Stakeholders are the main source of SRs and it should be identified prior to the beginning of the SRs elicitation process [6].Stakeholders are people, groups, or individuals who have an impact on or are affected by a company directly or indirectly [6] Stakeholders can be categorized according to their project participation, like the project's financers, decision makers, developers, users, etc. The main objective of this step is to recognize the different stakeholders for the identification of various types of SRs.

- **Step 2: Determine the various linguistic variables**
  In our work, following linguistic variables have been used for capturing the opinions of the stakeholders: "Very High (VH)", "High (H)", "Medium (M)", "Low (L)", and "Very Low (VL)".

- **Step 3:Decision makers provide their ratings for SRs**
  A decision-maker or an expert is one of the key stakeholders involved in the development of software product. A decision-maker provides his preferences or judgments over the importance of SRs based on some defined criteria in the form of some linguistic variables like "Very High", "High", "Medium", "Low", and "Very Low", etc. The aim of this step is to provide the ratings of the SRs by decision makers.

- **Step 4: Specify a population size ($\boldsymbol{N}$)**
  The effectiveness of a GA is strongly dependent on the proper selection of the population size, selection method, kind of crossover, crossover probability, mutation probability, etc. [31]. Among these, the population size is the most important factor because it impacts the GA's performance. The initial performance of a GA is increased if the size of the population is small. The large population enhances the performance for the long term [32]. All through, the size of the population should be constantduring the search process [32].

- **Step 5: Set a target chromosome by using the corresponding values of LVs generated through some intuitionist approach**
  A target chromosome is formed by using the corresponding values for the defined linguistic variables. These values are generated through some intuitionist approach.

- **Step 6: Encode the target chromosome into binary form**
  The aim of this step is to encode the target chromosome into the binary format. The binary encoding is the simplest and the most frequently used encoding scheme that converts the target chromosome into a string of binary digits, i.e., 0s and 1s [32].

- **Step 7: Generate an initial population ($\boldsymbol{P}$)**
  The generation of an initial population is the first and very important step in a GA process. The selection of the optimum solutions and reducing the computing time for such solutions are possible if the initial population is sufficiently varied [31]. There are several techniques for creating initial populations. The random generation is the default and commonly used technique [32]. The random generation of the initial population is based on the probability of the genes (i.e., a string of 1s and 0s) of the chromosome. The value of a gene may be 1 or 0 on the basis of its probability. If the value of a gene is 1 with its probability $pr$, then the probability of gene with value 0 will be 1-pr [32].

- **Step 8: Is the target chromosome achieved?**
  The aim of this step is to check whether the target chromosome has been achieved or not. If the target chromosome has been achieved then go to step 15, otherwise go to step 9.

- **Step 9: Compute the fitness value of each chromosome of the population**
- The calculation of the fitness value of each individual (or chromosome) of a population is a very important task of the GA. For all the generations, the evaluation of each individual of a population is based on its fitness value. If the fitness value of an individual is higher, it means it has a higher probability of carrying into the next generation [31]. To find out the fitness value i.e., the closeness between a chromosome and the target chromosome (or optimal solution), the fitness function is required [31]. Generally, the objective function of any problem is considered as the fitness function. The objective of this step is to check what percentage of genes (i.e., the binary digits) of a chromosome is identical to the corresponding genes of the target chromosome. Therefore, the following fitness function has been used in our work:

$$fitnessvalue(fv) = \frac{count(g(c)==g(tc))}{sizeof(c)}$$
(1)

where,

$c =$ chromosome, $tc =$ target chromosome, and $fv =$ fitness value

- **Step 10: Selecttwo parent chromosomes based on fitness values**
After the fitness value calculation, the next step is to select any two chromosomes with the highest fitness value for the crossover purpose so that the new population can be formed. The selection process includesthe selection of any two chromosomes randomly based on their fitness value, i.e., the chromosome with the highest fitness has more chance for its selection [32]. In this step, tournament selection method has been used because of its effectiveness and ease of implementation; in which some individuals are chosen randomly from a population to compete against each other [33]. The individual who achieves the best level of fitness succeeds and becomes a member of the next generation population.

- **Step 11: Crossover operation**
The crossover is a fundamental operation in GA that is used to create a new child chromosome (i.e. offspring) by merging parts of selected parent chromosomes [31]. A new population is generated when each new born child chromosome replaces its first parent chromosome from the current population. There are many types of crossovers like one-point, two-point, multi-point, and uniform crossover. The one-point or two-point crossover operations do not produce the required outcomes when the chromosomes are exceedingly lengthy; while increasing the number of crossover points yields acceptable results. Thus, in this step multi-point crossover is used during the computational process.

- **Step 12: Mutation operation**
During the crossover process, the genes are exchanged between the chromosomes. The problem of this mechanism is that it does not result in new genes that vary from those in the parents. So, GA additionally takes into account the mutation process. As the crossover process completed the mutation process begins and it creates a new chromosome by changing the values of one or more segments of an existing chromosome [33].

- **Step 13: Create a new population $(NP)$**
A set of child chromosomes is formed by the processes of crossover and mutation over the parent chromosomes selected from the given population. This set is termed as a new population $(NP)$ that has genetically (i.e. property-wise) different chromosomes.

- **Step 14: Is population size of chromosomes reached to some limit?**
In this step, the population size is checked. If the population size is reached to some predefined limit then go to step 8, otherwise go to step 10.

- **Step 15: Select the best chromosome**
After a number of iterations performed by GA on the basis of the termination criteria, we got a final generation of population. Then, the best chromosome on the basis of its fitness value is selected and decoded from binary to the decimal value. This value has been used by the fuzzy TOPSIS method as the FMF to perform the evaluation of FRs based on some given NFRs.

- **Step 16: Calculation of AFW and AFR for the NFRs and FRs respectively**
The fuzzy rating for FR $(A_i)$ based on the NFR $(C_j)$ by the $n^{th}$ decision maker can be represented as Eq. 2, and the weight of NFR can be represented as Eq. 3.

$$\tilde{r}_{ij}^{n} = (\tilde{a}_{ij}^{n}, \tilde{b}_{ij}^{n}, \tilde{c}_{ij}^{n}) \qquad (2)$$

and the weight of $C_j$ is represented as

$$\widetilde{w}_{j}^{n} = (\widetilde{w}_{j1}^{n}, \widetilde{w}_{j2}^{n}, \widetilde{w}_{j3}^{n}) \qquad (3)$$

The AFW for NFR is calculated by using the Eq. 4, whereas the calculation of the AFR for the NFRs is performed by using the Eq. 5.

$$w_{j1} = \min_{n}\{w_{j1}^{n}\},\ \ w_{j2} = \frac{1}{n}\sum_{n=1}^{n} w_{j2}^{n}, w_{j3} = \max_{n}\{w_{j3}^{n}\} \ (4)$$

$$a_{ij} = \min_{n}\{a_{ij}^{n}\},\ \ b_{ij} = \frac{1}{n}\sum_{n=1}^{n} b_{ij}^{n}, c_{ij} = \max_{n}\{c_{ij}^{n}\} \qquad (5)$$

- **Step 17: Calculation of NFDM and WNFDM**
The calculation of NFDM is performed by using Eq. 6, and on the other hand the WNFDM is calculated by using Eq. 7.

$$\widetilde{m}_{ij} = \left(\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*}\right) where c_j^* = \max_{i}\{c_{ij}\} \qquad (6)$$

$$\tilde{n}_{ij} = \widetilde{m}_{ij} \times w_j \qquad (7)$$

- **Step 18: Calculation of fuzzy FPIS and fuzzy FNIS**
The values of the fuzzy FPIS and the fuzzy FNIS are calculated by using the Eq. 8 and Eq. 9 respectively [14].

$$A^* = (\tilde{n}_1^*, \tilde{n}_2^*, \dots, \tilde{n}_j^*), where \tilde{n}_j^* = \max_{i}\{n_{ij3}\}; \qquad (8)$$
$$A^- = (\tilde{n}_1^-, \tilde{n}_2^-, \dots, \tilde{n}_m^-), where \tilde{n}_j^- = \max_{i}\{n_{ij1}\}; \qquad (9)$$

- **Step 19: Calculation of distance of each FR from fuzzy PIS and fuzzy NIS**
The computation of the distance of each FR from the fuzzy PIS is performed by using Eq. 10, whereas Eq. 11 is used to calculate the distance of each FR from the fuzzy NIS.

$$d_i^* = \sum_{j=1}^{m} d(\tilde{n}_{ij}, \tilde{n}_j^*) \qquad (10)$$

$$d_i^- = \sum_{j=1}^{m} d(\tilde{n}_{ij}, \tilde{n}_j^-) \qquad (11)$$

- **Step 20: Calculation of closeness coefficient (CCof)**
  The value of closeness coefficient ($CCof$) is calculated on the basis of the values of distances of each FR from fuzzy PIS and fuzzy NIS by using the Eq. 12. The ranking order of FRs is decided on the basis of the values of $CCof$.
  $$CCof = \frac{d_i^-}{d_i^- + d_i^*} \qquad (12)$$

## V. EXPERIMENT AND RESULTS

The objective of this section is to apply the proposed method for the selection of the requirements of an examination system in which genetic algorithm is used for generating the fuzzy membership function (FMF) values. In this study, fifteen requirements of FRs of an IES have been used. The following linguistic values have been used in this work. This FMF is converted into a target chromosome of 60 (5×12) bits by using the binary encoding technique, as shown in Table I.

VL= (1,1,3),L= (1,3,5),M= (3,5,7),H= (5,7,9), and H= 7,9,9).

With the help of GA, a random population of size 15 is initialized. It is named as generation-0; and it isshown in Table II. After the computation, we have got the average fitness value of all the chromosomes = 0.814in generation-0. By performing the process of reproduction over the generation-0 by using the selection, crossovers, and mutation operators, generation-1 is formed. This process of reproduction continued till 23 different generations. The population and fitness values of all chromosomes in Generation-1 are generated. After that fitness value of the chromosomes are computed. As a result the average fitness value in generation-1 is 0.8278.

The same process was repeated for 23rd generation. From the last generation (i.e., Generation-23) the best chromosome is selected. The fitness value of this chromosome is 0.8467 while the average fitness value is 0.8467. The variation in the average fitness values of the chromosomes from one generation to another is exhibited in Fig. 2.

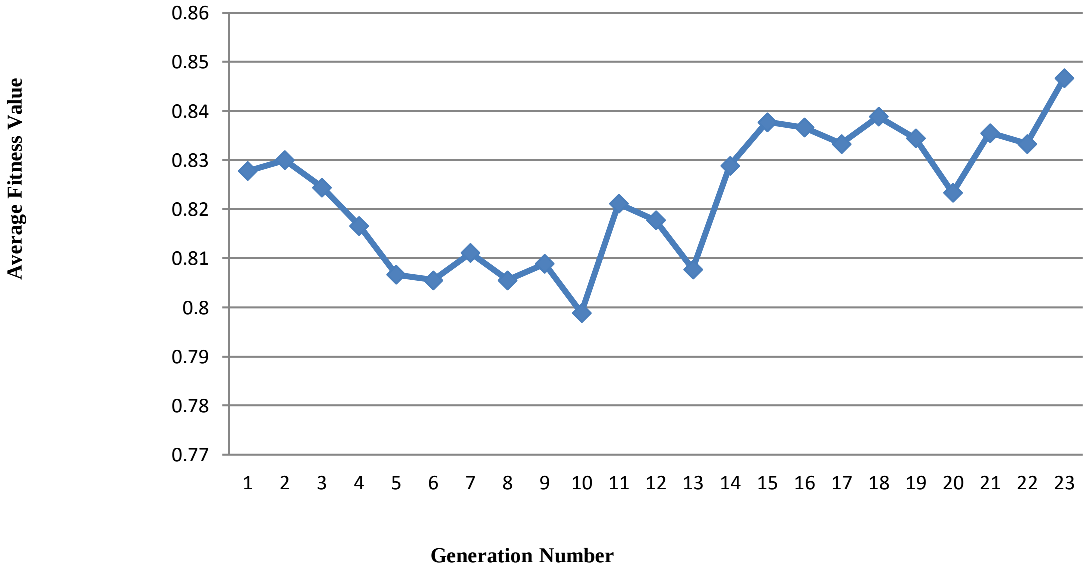

Fig. 2: Variation in the fitness values

| Linguistic Variables | VL, L, M, H, VH |
|---|---|
| Fuzzy Membership Values | (1,1,3), (1,3,5), (3,5,7), (5,7,9), (7,9,9) |
| Target Chromosome | 000100010011000100110101001101010111010101111001011110011001 |

Table 1: Binary encoded target chromosome

| Generation-0 | |
|---|---|
| Chromosome1 | 000100010011000100110101001101000101010001010111010110101010 |
| Chromosome2 | 000100010011000100100011001001000110010001101000011010011001 |
| Chromosome3 | 000000000110000000100100001000110101001101011000010110101010 |
| Chromosome4 | 000100010010000100110101001101010111010101101000011010011001 |
| Chromosome5 | 000000000110000000100100001000110101001101101001011010011001 |
| Chromosome6 | 000000000100000000010011000101000111010001101000011010011001 |
| Chromosome7 | 000100010011000100110101001101000101010001101000011010011001 |
| Chromosome8 | 000100010011000100100011001001000111010001101001011010101010 |
| Chromosome9 | 000100010010000100110101001101000110010001101001011010011001 |
| Chromosome10 | 000000000100000000100101001001000110010001101001011010101010 |
| Chromosome11 | 000100010011000100100011001001000110010001010111010110101010 |
| Chromosome12 | 000100010011000100100011001000110101001101011000010110101010 |
| Chromosome13 | 000100010010000100100100001001000111010001101001011010011001 |
| Chromosome14 | 000100010011000100100100001001000111010001101000011010011001 |
| Chromosome15 | 000100010010000100100100001000110101001101101001011010011001 |

Table 2: Population in Generation-0

The maximum average fitness value is 0.846666667, whereas the minimum fitness value is 0.798888889.The total variation in the average fitness value is 0.047777778. The decoded FMF value of the high ranked chromosome is (1,1,3), (1,3,7), (2,4,7), (4,6,8), (6,9,9). This value is used by the fuzzy TOPSIS method to create the decision matrices on the basis of the preferences given by various decision makers in the form of some linguistic variables. The ranking of the FRs is generated by using the proposed method and the results are shown in Table III.

| Functional Requirements | Ranking | |
|---|---|---|
| | **Fuzzy TOPSIS method** | **Proposed method** |
| $FR_1$ | 4 | 4 |
| $FR_2$ | 12 | 12 |
| $FR_3$ | 14 | 10 |
| $FR_4$ | 7 | 8 |
| $FR_5$ | 15 | 15 |
| $FR_6$ | 3 | 3 |
| $FR_7$ | 13 | 14 |
| $FR_8$ | 11 | 13 |
| $FR_9$ | 8 | 7 |
| $FR_{10}$ | 9 | 6 |
| $FR_{11}$ | 1 | 1 |
| $FR_{12}$ | 5 | 9 |
| $FR_{13}$ | 10 | 11 |
| $FR_{14}$ | 6 | 5 |
| $FR_{15}$ | 2 | 2 |

Table 3: Ranking orders of FRs by using Fuzzy TOPSIS and Proposed methods

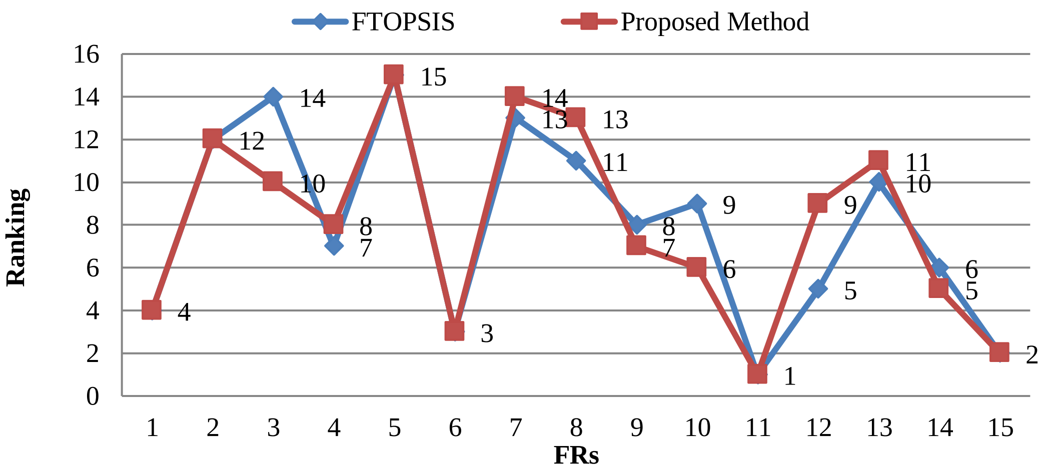

Fig. 3: Comparison between FTOPSIS and proposed method

We also generate the ranking of FRs by using fuzzy TOPSIS method and the results are summarized in Table IV. There were some variations in the ranking order of the requirements and it is shown in Fig. 3.

## VI. CONCLUSION AND FUTURE WORK

This paper presents a method for the SRs selection in which genetic algorithm is used with fuzzy TOPSIS method for computing the ranking order of the requirements of an IES. In our work, the value of the FMS is computed by using the genetic algorithm. After applying the proposed method, it is found that top three requirements are $FR_{11}$, $FR_{15}$, and $FR_6$. The ranking order of the requirements produced by the proposed method is compared with the fuzzy TOPSIS. As a result, it is found that both the methods produced the same results for the top $n$ requirements but there are some variations in the ranking order for some requirements. As we know that an appropriate ranking order of requirements may lead to the failure of software. So, eliciting the ranking order of requirements is one of the key activities of software development process. The future research agenda includes the following: (a) to apply the proposed method on large data set having more than 100 requirements of a system (b) to integrate genetic algorithm with fuzzy AHP and analyze the results.

## REFERENCES

[1.] Zhou, J. Ying, “Functional requirements and non-functional requirements: a survey”, Diss. Concordia University, 2004.

[2.] Eckhardt, Jonas, A.Vogelsang, and D. M. Fernández, "Are non-functional requirements really non-functional? an investigation of non-functional requirements in practice", Proceedings of the 38th International Conference on Software Engineering, 2016.

[3.] P. Jalote, “An Integrated Approach to Software Engineering”, 3rd edition, Narosa Publishing house, India, 2005.

[4.] B.H.C. Cheng and J.M. Atlee, "Research Directions in Requirements Engineering", Future of Software Engineering (FOSE '07), Minneapolis, MN, pp. 285-303, 2007.

[5.] S. Komi-Sirvio& M. Tihinen, “Great Challenges and Opportunities of Distributed Software Development - An Industrial Survey”, Fifteenth International Conference on Software Engineering and Knowledge Engineering, SEKE2003, San Francisco, 2003.

[6.] M. Nazim, C. W. Mohammad, and M. Sadiq. "Analysis of Fuzzy AHP and Fuzzy TOPSIS Methods for the Prioritization of the Software Requirements", Multiple Criteria Decision Making, Springer, Singapore, pp. 79-90, 2022.

A. M. Hickey, A. M., Davis, “Elicitation Technique Selection: How Do Experts Do It?”, In: 11th IEEE International Conference on Requirement Engineering, 2003.

[7.] Perini, A. Susi, F. Ricca, C.Bazzanella, “An Empirical Study to Compare the Accuracy of AHP and CBRanking Techniques for Requirements Prioritization”, In: *Fifth International Workshop on Comparative Evaluation in Requirements Engineering*, pp. 23-35, 2007.

[8.] M. Pitangueira, R.S.P. Maciel, and M. Barros, “Software requirements selection and prioritization using SBSE approaches: A systematic literature review and mapping of literature”,Journal of Systems and Software, Vol. 103, pp. 267–80, 2015.

[9.] M. Sadiq, V.S. Devi, “Prioritization and Selection of the Software Requirements using Rough-Set Theory”, IETE Journal of Research, pp. 1-18, 2021.

[10.] **M. Nazim,** C.W. Mohammad, and M. Sadiq, “Fuzzy based methods for the selection and prioritization of software requirements: A systematic literature review,” 9th International Conference on Frontiers of Intelligent Computing: Theory and Applications, Springer, NIT Mizoram, India, June 25, 2021.

[11.] K.B. Ijaz, I. Inayat, M. Daneva, F.A. Bukhsh, “Value-Based Fuzzy Approach for Non-functional Requirements Prioritization”, In: Shishkov B. (eds) Business Modeling and Software Design. BMSD 2021. Lecture Notes in Business Information Processing, Vol. 422. Springer, Cham, 2021.

[12.] Y. V. Singh, B. Kumar and S. Chand, "A Novel Approach of Requirements Prioritization Using Logarithmic Fuzzy Trapezoidal AHP for Enhancing Academic Library Service", International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 1164-1172,2018.

[13.] M. Sadiq, S. Khan, and C.W. Mohammad,“Selection of software requirements using TOPSIS under fuzzy environment”, International Journal of Computers and Applications, pp. 503-512, 2020.

[14.] D. Mougouei, D.M.W. Powers, “Dependency-aware software requirements selection using fuzzy graphs and integer programming”, Expert Systems with Applications, Vol. 167, 2021.

[15.] D. Mougouei, D.M.W. Powers, and M. Elahe, “A Fuzzy Framework for Prioritization and Partial Selection of Security Requirements in Software Projects”,pp. 2671 – 2686, 2019.

A. Hassan, N. Ramadan,“A Proposed Hybrid Prioritization Technique for Software Requirements based on Fuzzy Logic”, CiiT International Journal of Fuzzy Systems, Vol. 10, No. 2, pp. 45-52, 2018.

[16.] Y. V. Singh, B. Kumar, B., S. Chand, J. Kumar, “A comparative analysis and proposing ‘ANN fuzzy AHP model’ for requirements prioritization”, International Journal of Information Technology and Computer Science, Vol. 4, pp. 55-65, 2018.

[17.] D. Mougouei, A. Ghose, H. Dam, M. Fahmideh and D. Powers, "A Fuzzy-Based Requirement Selection Method for Considering Value Dependencies in Software Release Planning", IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6, 2021.

[18.] M. Alrashoud, A. Abhari, “Perception-based software release planning”, Intelligent Automation and Soft Computing, Vol. 21, Issue 2, pp. 175-195, 2015.

[19.] Jawale, A.T. Bhole, “Adaptive fuzzy hierarchical cumulative voting: a novel approach toward requirement prioritization”, International Journal of Research in Engineering and Technology, Vol. 4, no. 5, pp. 365-370, May 2015.

A. Afrin and M. Sadiq, “An integrated approach for the selection of software requirements using fuzzy AHP and fuzzy TOPSIS method”, International Journal of Advance Research and Development, Vol. 2, no. 6, pp. 170-183, 2017.

[20.] K.S. Ahmad, N. Ahmad, H. Tahirand S. Khan, “Fuzzy_MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements”, International Conference on Intelligent Computing, Instrumentation and Control Technologies, pp. 433-437, 2017.

[21.] V.C. Gerogiannisand G. Tzikas, “Using fuzzy linguistic 2-tuples to collectively prioritize software requirements based on stakeholders evaluations”, PCI, 2017.

[22.] M. Sadiq, V. S. Devi, J. Ahmad, and C.W. Mohammad, “Fuzzy logic driven security requirements engineering process”, Journal of Information and Optimization Sciences, Vol. 42, Issue 7, pp. 1685-1707, 2021.

[23.] P. Achimugu, A. Selamat, R. Ibrahim, and M.N. Mahrin, “An adaptive fuzzy decision matrix model for software requirements prioritization”, Advanced Approaches to Intelligent Information and Database Systems, N. T. Nguyen and R. Kowalczyk, Eds. Cham, Switzerland: Springer, pp. 129_138, 2014.

[24.] N. Sharif, K. Zafar, and W. Zyad, “Optimization of requirement prioritization using computational Intelligence technique”, International Conference on Robotics and Emerging Allied Technologies in Engineering, pp. 228-234, 2014.

[25.] H. Momeni, H. Motameni H., and M. Larimi, “A neuro-fuzzy based approach to software quality requirements prioritization”, International Journal of Applied Information Systems (IJAIS), Vol. 7, no. 7, 2014.

[26.] M. Sadiq, and S. Jain, “Applying fuzzy preference relation for requirements prioritization in goal oriented requirements elicitation process”, International Journal of System Assurance Engineering and Management, Vol. 5, 2014.

[27.] J.H. Holland, “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, 1975.

[28.] S. Katoch, S.S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future”, Multimedia Tools and Applications,Vol. 80,pp. 8091–8126, 2021.

[29.] M. Abdelaziz, “Distribution network reconfiguration using a genetic algorithm with varying population size”, Electric Power Systems Research, Vol. 142, pp.9-11, 2017.

[30.] N.M. Razali, and J. Geraghty, “Genetic algorithm performance with different selection strategies in solving TSP”, In Proceedings of the world congress on engineering, Hong Kong, China: International Association of Engineers, Vol. 2, No. 1, pp. 1-6, 2011.