International Journal of Decision Support System Technology
Volume 15 • Issue 1

Fuzzy Distance-Based Approach for the Assessment and Selection of Programming Languages:
Fuzzy-Based Hybrid Approach for Selection of PL

Rakesh Garg, Amity University, Noida, India
https://orcid.org/0000-0001-7037-4209

Supriya Raheja, Amity University, Noida, India*

ABSTRACT

The desire to develop software with more and more functionalities to make human work easier pushes the industry towards developing various programming languages. The existence of the various programming languages in today's scenario raises the need for their evaluation. The motive of this research is the development of a deterministic decision support framework to solve the object-oriented programming (OOP) language's selection problem. In the present study, OOP language's selection problem is modeled as a multi-criteria decision-making, and a novel fuzzy-distance based approach is anticipated to solve the same. To demonstrate the working of developed framework, a case study consisting of the selection of seven programming languages is presented. The results of this study depict that Python is the most preferred language compared to other object-oriented programming languages. Selection of OOP languages helps to select the most appropriate language, which provides better opportunities in the business domain and will result in high success for engineering students.

KEYWORDS

Computer Engineering, DBA, Decision Support framework, Engineering Education, Fuzzy, Hybrid, Multi-criteria Decision Making, Object Oriented, Validation

Introduction

In the era of technology, computer Technocrats in engineering education desire to excel by being at par with the latest technologies and digital advancement. Programming languages are the necessary and inevitable tools to design new algorithms and software systems. The efficiency of these systems depends on the programmer and features provided by the programming language. Many programming languages are prevalent in the digital world, today offering a wide range of features. With the advancement in technology, software companies are upgrading their systems and programming languages that provide the best features applied to their business domain. Programming languages form the foundation of any software system, and their presence can be traced back to several decades. 'FORTRAN,' 'COBOL,' and 'LISP' were the oldest procedure-oriented programming languages developed during 1957-1960. In the 1970s, another procedure-oriented language, 'PASCAL,' came

DOI: 10.4018/IJDSST.315761
*Corresponding Author

into existence, followed by the 'C' language that became the base language for many new prominent languages. Further, many new languages like 'C++,' 'Java,' 'Pearl,' 'Ruby,' 'Python,' 'PHP' etc. came into existence for the software development known as OOP languages (Cook, 1986; Parker et al., 2006; Henderson & Zom, 1994; Bosch, 1997; Lesani et al., 2014; Maplesden et al., 2015; Onu F.U et al., 2016; Anfurrutia et al., 2017; Yadav et al.,2017).

Further, to get placed in the best position in software companies, engineering students need to know the latest trends in programming languages. OOP languages occupy a significant position due to their modularity and reusability. There are many OOP languages present with a different set of features. So, it becomes a tedious task for the students who aspire to become a programmer to narrow their decision to a single OOPL that will set their career path. Further, selecting a suitable OOP language may depend on many conflicting attributes. Hence, the task of choosing an OOP language from the trending languages can be considered as an MCDM problem. MCDM is an optimization technique that helps to deduce a single optimum solution from the set of options available (Mardani et al.,2017; Biswas et al., 2021; Goswami et al., 2021; Garg et al., 2021; Chodha et al., 2022; Bansal et al., 2022). Further, the high use of various MCDM approaches is observed for solving many selection problems as university/school selection, teachers evaluation and selection, e-learning website selection, and funds allocation in the field of Education (Erdoğan, & Kaya, 2014; Baykasoglu & Durmusoglu, 2014; Gürbüz, & Albayrak, 2014; Chiang, 2015; Karmaker & Saha, 2015; Jain et al., 2015; Jain et al., 2016; Chang, & Wang, 2016; Chakraborty et al., 2017; Özdemir, & Tüysüz, 2017; Garg & Jain, 2017; Garg, 2017; Handoko et al., 2017; Garg et al., 2018; Hasan et al., 2021; Ghorui et al., 2021).

In this study, the FDBA method is proposed for solving the problem of OOP language selection based on ten selection parameters. The proposed FDBA method integrates fuzzy set theory (FST) and Distance-based approach (DBA). Here, fuzzy set theory is used to get the priority weights of selection criteria and the performance rating of OOP languages concerning selection criteria. DBA is used to get the comprehensive ranking of OOP languages for their selection purpose.

The rest of the paper is structured as follows: Section 2 highlights the selection criteria and MCDM methods used by the researchers to solve the present selection problem, whereas section 3 describes the proposed MCDM method, i.e., FDBA and the developed decision support framework. An illustrated example for OOP language selection is provided in section 4, followed by methodology validation in section 5. The results are discussed in section 6, followed by the significance of the FDBA in section 7. The conclusion and future scope of the research are given in section 8.

Literature Review

The present study focuses on implementing an integrated MCDM approach to solve the programming languages selection problem. A comprehensive literature review is carried out to get detailed information about the various selection criteria and methodologies used in the past. This section is divided into two subsections as (i) Related work and (ii) Motivation for present research.

Related Work

In 1994, authors modeled the programming language selection as an MCDM problem and compared four different languages which majorly supports the inheritance, dynamic dispatch, code reuse and information hiding based on five selection criteria: compiler, optimized time, compile time, object size, and binary size (Henderson & Zorn, 1994). Al Ahmar (2010) presented the prototype of an expert system that supports software project managers and software engineers in selecting the appropriate software development methodology. However, the author discussed on the development methodologies and does not discuss regarding programming languages.

In the contemporary work, Parker et al. (2006) introduced some more selection criteria and implemented the Analytical Hierarchical Process (AHP) approach to solve the programming language selection problem. AHP mainly works on comparing the alternatives to each other (Bakır et al., 2021;

Karamasa et al., 2021). The major problem with this approach is the high complexity in the case of many selection criteria and alternatives. In 2014, authors combined the AHP approach with fuzzy set theory to solve the same problem (Lesani et al., 2014). They applied a restricted set of criteria and selected best language from a small set of languages. Later, many of the authors observed the present programming language selection problem is dependent on many other selection criteria based on their features and functionality provided (Maplesden et al., 2015; Anfurrutia et al., 2017; İmamoğlu & Çetinkaya, 2017; Yıldızbaşı & Rouyendegh, 2018) like 'Academic acceptance' 'Industry acceptance' 'Ease of use', 'Purpose of language', 'Methodology of language' 'Ability of Language' etc. Many of the authors mainly considered the top OOPs languages namely "Python", "Java", "C++" and "C#".

With time some other MCDM approaches as Technique for Order Preference Similarity to Ideal Solution (TOPSIS) and Multi-Attribute Border Approximation Area Comparison (MABAC), came into existence (Yadav et al., 2017; Farshidi et al., 2021). TOPSIS works on calculating two distance measures for obtaining the preference index of the alternatives. Recently MABAC method is introduced by Pamučar & Cirovic (2015). It has a systematic computation procedure and a reasonable logic that shows the basis for decision-making. Normally, people follow the trend in software development industry and accepted the selection of programming language as a challenging task. The various selection criteria and the methodologies available in the literature are listed in Table 1.

Motivation for the Present Research

The following inferences are drawn from the extensive study of literature review that motivates to carry out the presented study:

1. The literature review study reveals the existence of many programming languages that are part of the course curriculum for engineering education adopted by academic organizations. This high availability of programming languages makes their selection very difficult for engineering students.
2. Most authors argued that the programming language's selection problem could be shaped as an MCDM problem due to multiple conflicting selection attributes and implemented various MCDM methods such as AHP, TOPSIS, etc. These MCDM methods suffer from many issues such as high complexity in the case of many alternatives and selection attributes, no consideration of priority weights of selection attributes, time-consuming, etc. So, there is a need to develop a decision-making framework to help a more efficient MCDM method solve the programming language's selection problem.

Research Methodology

This section provides insights about the proposed methodology, i.e., FDBA and the developed decision-making framework for the programming languages selection problem presented in this research.

Fuzzy Distance-Based Approach (FDBA)

The proposed MCDM method, i.e., FDBA, combines the fuzzy set theory (FST) to deal with the vagueness of the data with a distance-based approach (DBA) to get the alternative's ranking for their selection purpose.

Fuzzy set theory (FST): FST is a mathematical approach developed by Lotfi A. Zadeh (1965) to deal with the various ambiguity aspects such as incompleteness, unpredictability, impreciseness, and fuzziness related to any data.

Table 1. Existing Selection Criteria and Methodologies

Author/year	Henderson & Zorn, 1994	Parker et al., 2006	Lesani et al., 2014	Maplesden et al., 2015	Anfurrutia et al., 2017	Yadav et al., 2017	İmamoğlu, & Çetinkaya, 2017	Yıldızbaşı & Rouyendegh, 2018	Mishra et al, 2020	Farshidi et al., 2021
Selection criteria										
Compile time	✓	X	X	X	X	X	X	X	X	X
Object size	✓	X	X	X	X	X	X	X	X	X
Academy acceptance	X	✓	✓	X	X	X	X	✓	✓	X
Industrial acceptance	X	✓	✓	X	X	X	X	✓	✓	X
Software properties	X	X	✓	X	ü	X	X	✓	X	X
Ease of use	X	X	✓	X	ü	X	X	✓	X	X
Purpose of language	X	X	✓	X	X	X	X	✓	X	X
Programming Paradigm	X	X	✓	X	X	X	X	✓	X	ü
Availability of languages	X	✓	✓	X	ü	X	✓	X	X	X
Language Ability	X	X	X	X	X	X	X	✓	X	X
Cost	X	✓	X	✓	X	X	✓	X	X	X
Market Reputation	X	✓	X	X	X	X	X	X	X	X
Proprietary/ open source	X	✓	X	X	X	X	X	X	X	X
Development environment	X	✓	X	X	X	X	X	X	X	✓
Support for Secure Code	X	✓	X	X	X	X	X	X	X	X
Memory access	X	X	X	✓	X	X	X	X	X	X
Expandability	X	X	X	X	X	✓	X	X	X	X
Fault tolerance	X	X	X	X	X	✓	X	X	X	X
Modularity	X	X	X	X	X	✓	X	X	X	X
Operability	X	X	X	X	X	✓	X	X	X	X
Application Domain	X	X	X	X	X	X	X	X	X	✓
Development Stack	X	X	X	X	X	X	X	X	✓	✓
Efficiency	X	X	X	X	X	X	X	X	✓	X
Methodology Used										
FAHP	X	X	✓	X	X	X	X	X	X	X
TOPSIS	X	X	X	X	X	X	X	✓	X	X
AHP	X	✓	X	X	X	X	X	X	X	X
AHP-FTOPSIS	X	X	X	X	X	✓	X	X	X	X
MABAC	X	X	X	X	X	X	X	X	ü	X

Fuzzy Sets: Fuzzy set B as the sets on universe N= $\left\{s_1, s_2, \ldots\ldots, s_n\right\}$ which can accommodate the degree of membership then $\left\{\left(s_i,\ \mu_B\left(s_i\right)\right)\right\}$ where $s_i \in N, \mu_B : N \to \left[0,1\right]$ is the membership function of B and $\mu_B : \left(s\right) \in \left[0,1\right]$ is the degree of membership of set 's' in B.

Fuzzy Number: Fuzzy set B in universal set U is the fuzzy number if B is a normal fuzzy set with closed interval $B \in \left(0,1\right]$. Fuzzy numbers depict the real number more realistically than single-valued numbers.

Triangular fuzzy number (TFN): TFNs represent a human expert's uncertainty in decision-making through $Z = \left(z_1, z_2, z_3\right)$. Parameters z_1, z_2, z_3 are maximum, internal, and minimum values are, depicting indecisiveness.

Mathematically, membership function for representing TFN is denoted as in equation (1):

$$\mu_y\left(Z\right) = \begin{cases} 0 & y < z_1 \\ \left(y - z_1\right) / \left(z_2 - z_1\right) & z_1 \le y \le z_2 \\ \left(z_3 - c\right) / \left(z_3 - z_2\right) & z_2 \le y \le z_3 \\ 0 & y > z_3 \end{cases} \tag{1}$$

Arithmetic Operations are applied on TFNs such as addition, subtraction, minimum, maximum, multiplication, division, average, etc. In this study, addition and average arithmetic operations used on two TFNs $Z_1 = \left(z_1^u, z_1^m, z_1^l\right)$ and $Z_2 = \left(z_2^u, z_2^m, z_2^l\right)$ can be stated as in equation (2) and equation (3):

Addition: $$Z_1 \oplus Z_2 = \left(z_1^u + z_2^u, z_1^m + z_2^m, z_1^l + z_2^l\right) \tag{2}$$

Division: $$Z_1 \oslash Z_2 = \left(z_1^u \,/\, z_2^u, z_1^m \,/\, z_2^m, z_1^l \,/\, z_2^l\right) \tag{3}$$

Here, 'u,' 'm', and 'l' represent maximum internal and minimum values, respectively.

Distance-based approach (DBA): DBA is a deterministic quantitative method widely used to solve the MCDM problems (Kumar & Garg, 2010; Amit et al., 2014; Garg et al.,2016; Sandhya & Garg,2016; Bibyan & Anand,2022). The basic concept of the DBA method is to find the distance of each alternative from the ideal point. Here, the ideal point refers to having ideal values for all the parameters considered in the alternative's evaluation.

Algo: Distance-Based Approach (DBA)

Consider a decision-making problem 'P' with a set of alternatives 'A' and a set of selection parameters 'C' as given below.

$$A = \left\{A_1, A_2, A_3, \ldots, A_n\right\}$$

$$C = \left\{C_1, C_2, C_3, \ldots, C_m\right\}$$

1. Initialize a decision matrix $\left[A_{ij}\right]_{nxm}$ consisting of the ratings of 'n' alternatives w.r.t. 'm' selection parameter. Here, A_{ij} represent the rating of alternative w.r.t. selection parameter and $i=\left(1,2,3,\ldots,n\right)$ and $j=\left(1,2,3,\ldots,m\right)$.
2. Define the criteria type and the weight of all selection parameters as:

$\left\{C_b\right\} \in C; C_b$ is the set of beneficiary parameters.

$\left\{C_{nb}\right\} \in C;$ Cnb is the set of non-beneficiary parameters.

$\left[w_j\right] = \left[w_1, w_2, w_3, \ldots w_m\right]; w_j$ is the weight of selection parameter

3. Append the decision matrix as $\left[OPT - A_{ij}\right]_{(n+1)\text{x}m}$

for $j = 1\ to\ m$
for i $= 1\ to\ n$
if ($C_j \in C_b$) then
$\left[OPT - A_{(n+1)j}\right] = max_j\left(A_{ij}\right)$
else $\left[OPT - A_{(n+1)j}\right] = min\left(A_{ij}\right)$

4. Now, standardize the $\left[OPT - A_{ij}\right]_{(n+1)\text{x}m}$ to eliminate the unit differences as $\left[STD - A_{ij}\right]_{(n+1)\text{x}m}$

for $j = 1\ to\ m$
for i $= 1\ to\ n$

$$\left[Avg_{ij}\right] = \frac{1}{n+1}\sum_{i=1}^{n+1} A_{ij}$$

$$\left[SDEV_{ij}\right] = \left[\frac{1}{n+1}\sum_{i=1}^{n+1}\left(A_{ij} - Avg_{ij}\right)^2\right]^{1/2}$$

$$STD - A_{ij} = \left[\frac{A_{ij} - Avg_{ij}}{SDEV_{ij}}\right]$$

5. Append the $\left[STD - A_{ij}\right]_{(n+1)\text{x}m}$ to find the weighted distances of each alternative from optimal point as given below.

for $j = 1\ to\ m$
for i $= 1\ to\ n$
if ($C_j \in C_b$) then

$$\left[DIS - A_{ij}\right] = \left(\left[STD - A_{(n+1)j}\right] - \left[STD - A_{ij}\right]\right) \times \left[w_j\right]$$

$$\text{else } \left[DIS - A_{ij}\right] = \left(\left[STD - A_{ij}\right] - \left[STD - A_{(n+1)j}\right]\right) \times \left[w_j\right]$$

6. Calculate the Euclidean distance (ED) for each alternative as:
for i $= 1$ *to* n

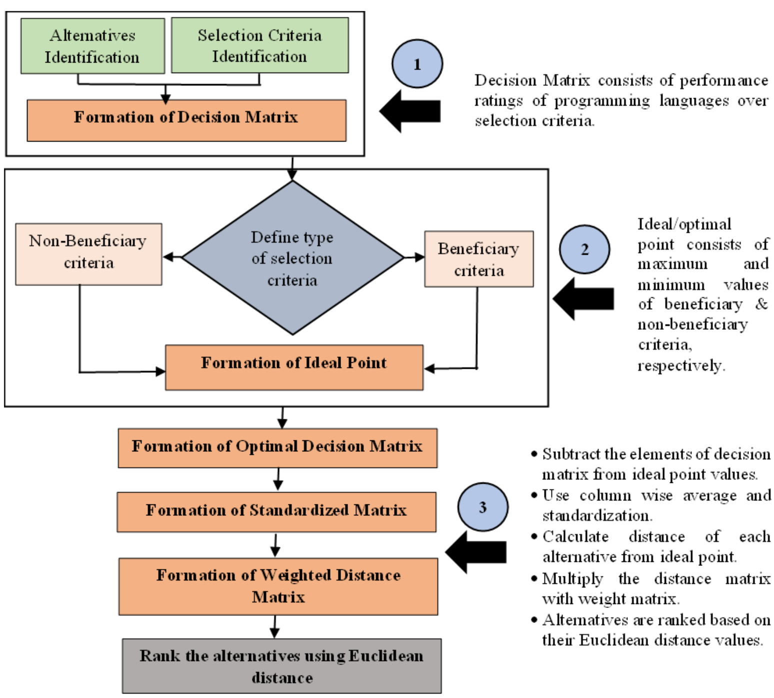
$$ED_i = \left[\sum_{j=1}^{m}\left(\left[DIS - A_{ij}\right]^2\right)\right]^{1/2}$$

7. Rank the alternatives based on the calculated Euclidean distance

Decision Making Framework for Programming Languages Selection

A three-phase decision-making framework is developed in this study to solve the present programming language's selection problem by shaping it as an MCDM problem. The first phase of the developed framework identifies the alternatives, i.e., programming languages and the selection parameters. The second phase estimates the priority weights of the selection attributes and the ratings of the alternatives. The third phase implements the proposed MCDM method, i.e., FDBA to rank the alternatives for making selection decisions. The developed DSF is further shown in Figure 1.

Figure 1. Decision-making framework for programming language's selection

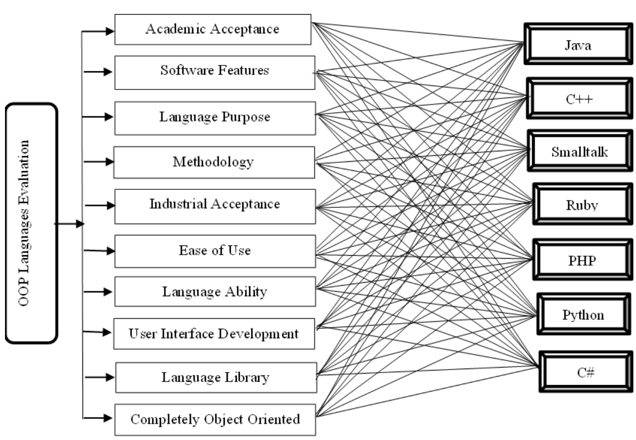


An Illustrated Example

This section presents the stepwise procedure for implementing FDBA to evaluate and select OOP languages.

1. *Identification and Selection of OOPLs:* Initially, 09 OOP languages are identified from the existing literature. After this identification, brainstorming sessions were conducted with a team of five decision-makers (D_1, D_2, D_3, D_4, D_5) of IT professionals and academicians working having a vast experience of more than 30 years experience. An elimination approach is used based on the current use of eight OOP languages earlier identified and finally, seven languages as 'Smalltalk,' 'C++,' 'Java', 'C#', 'Ruby', 'Python', 'PHP' were taken for evaluation and selection. Some of these languages may be used for specific projects and do not require any comparison with other languages. However, all the selected seven languages in this study support the object-oriented features that are the main reason for their consideration.
2. *Identification of Selection Criteria:* Ten selection attributes, namely, academic acceptance, software features, language purpose, methodology, industrial acceptance, ease of use, language ability, user interface development, completely object-oriented, and Language library, were identified for evaluating OOP languages. A brief description of all ten selection attributes is given below:
    - Academic Acceptance (C_1): To what extent the language is adopted in engineering education.
    - Software features (C_2): Features provided by the languages such as open source, reliability, independency of operating system, system requirements.
    - Language purpose (C_3): What is the primary purpose of programming languages to develop a web-based application or other application?
    - Methodology (C_4): empathetic and interpretation of code or expressions
    - Industrial acceptance (C_5): It concerns the usage of object-oriented programming languages in the industry.
    - Ease of use (C_6): How much is the language user-friendly?
    - Language ability (C_7): Evaluating qualified people to use these OOP languages.
    - User interface development (C_8): It provides a user-friendly environment, attractive and self-explanatory.
    - Completely object-oriented (C_9): To what extent the language supports the object-oriented concepts.
    - Language library (C_{10}): Availability of in-built libraries for development, debugging, and testing.
3. *Hierarchical structure formulation:* The present research presents the OOP language selection problem by modeling it as an MCDM problem. So, once the OOP languages and the selection parameters are identified, a hierarchical structure is formed, as given in Figure 2.

Figure 2. Hierarchical structure for OOP language's selection problem



4. *Questionnaire Formulation:* After the identification of selection attributes and the OOP languages, decision makers were asked to provide the date related to the priority weights of the ten selection attributes and the performance ratings of seven OOP languages over the selection attributes through a questionnaire. In order to get this data, two fuzzy scales were provided as (i) Scale-1 (Priority weights): Extremely less important 'ELI(0,0,0)', Very less important 'VLI(0.0,0.1,0.2)', Least important 'LI(0.2,0.3,0.4)', Important 'I(0.4,0.5,0.6)', More important 'MI(0.6,0.7,0.8)', Very more important 'VMI(0.8,0.9,1)', Extremely more important 'EMI(1,1,1)' and (ii) Sacle-2 (Performance ratings): Very low 'VL(0,0,0)', Low 'L(0.0,0.1,0.2)', Below average 'BA(0.2,0.3,0.4)', Average 'A(0.4,0.5,0.6)', Above average 'AA(0.6,0.7,0.8)', High 'H(0.8,0.9,1)', Very high 'VH(1,1,1)'.
5. *Determination of priority weights and performance ratings:* As stated in the previous step, decision-makers provide the data related to priority weights and performance ratings in linguistic terms as shown in Appendix-1 and Appendix-2, respectively, defined in fuzzy scales used. The obtained data is converted into a crisp value using averaging and aggregation operations of fuzzy set numbers.
6. *Reliability assessment of collected data:* In this study, secondary data is collected to evaluate and select OOP languages through questionnaires from the decision-makers. The consistency and reliability of data collected are checked by performing a reliability test using SPSS statistical software. In this test, the value of Cronbach's alpha is calculated as shown in Table 2.

Table 2. Reliability test statistics

Performance Ratings of OOP languages					
Cases		**N**	**%**	**Cronbach's Alpha**	**N of Items**
Based on Parameters	Valid	70	100	0.889	5
	Excluded	0	.0		
	Total	70	100		
Based on Experts	Valid	5	100	0.692	7
	Excluded	0	.0		
	Total	5	100.0		
Priority weights of selection attributes					
Cases		**N**	**%**	**Cronbach's Alpha**	**N of Items**
Based on Parameters	Valid	10	100	0.720	5
	Excluded	0	.0		
	Total	10	100		
Based on Experts	Valid	5	100	0.602	10
	Excluded	0	.0		
	Total	5	100		

7. *Formation of rating matrix:* The rating matrix is formed using Step1 of the DBA algorithm as given below:

$$\begin{bmatrix} 0.039 & 0.039 & 0.144 & 0.117 & 0.065 & 0.056 & 0.065 & 0.144 & 0.109 & 0.218 \\ 0.087 & 0.125 & 0.087 & 0.064 & 0.103 & 0.064 & 0.103 & 0.156 & 0.087 & 0.118 \\ 0.13 & 0.142 & 0.103 & 0.073 & 0.097 & 0.068 & 0.062 & 0.139 & 0.091 & 0.091 \\ 0.082 & 0.128 & 0.108 & 0.082 & 0.088 & 0.082 & 0.097 & 0.094 & 0.115 & 0.083 \\ 0.102 & 0.104 & 0.097 & 0.109 & 0.092 & 0.066 & 0.097 & 0.104 & 0.104 & 0.119 \\ 0.1 & 0.1 & 0.095 & 0.104 & 0.1 & 0.102 & 0.095 & 0.1 & 0.104 & 0.097 \end{bmatrix}$$

8. Formation of ideal and standardized matrices: Once, rating matrix is obtained, ideal matrix is formed using DBA algorithm. The average and standard deviation values are obtained as (0.0082, 0.0097, 0.0022, 0.0034, 0.0028, 0.0018, 0.002, 0.0016, 0.001, 0.0017) and (0.0043, 0.0048, 0.0025, 0.0023, 0.0013, 0.0016, 0.0013, 0.0017, 0.0007, 0.0021) respectively. Now, the standardized matrix is formed using step 4 of DBA algorithm and is given as:

$$\begin{bmatrix} -1.888 & -2.012 & 2.118 & 1.278 & -2.084 & -1.146 & -1.339 & 0.946 & 0.321 & 2.164 \\ -0.214 & 0.582 & -0.891 & -1.433 & 0.780 & -0.692 & 0.909 & 1.412 & -1.404 & -0.036 \\ 1.248 & 1.063 & -0.049 & -0.968 & 0.366 & -0.516 & -1.541 & 0.756 & -1.087 & -0.622 \\ -0.385 & 0.661 & 0.219 & -0.513 & -0.324 & 0.312 & 0.555 & -0.929 & 1.425 & -0.188 \\ 0.744 & -0.049 & -0.554 & 0.197 & 0.711 & 1.241 & 0.372 & -0.936 & 0.839 & -0.806 \\ 0.288 & -0.051 & -0.365 & 0.849 & 0.002 & -0.593 & 0.583 & -0.534 & -0.034 & -0.020 \\ 0.206 & -0.194 & -0.476 & 0.59 & 0.547 & 1.395 & 0.46 & -0.714 & -0.060 & -0.488 \end{bmatrix}$$

9. Formation of distance matrix: After forming a standardized matrix, another matrix, namely, a distance matrix, is formulated using equation step 5 of DBA as given below:

$$\begin{bmatrix} 0.224 & 0.197 & 0 & 0 & 0.088 & 0.053 & 0.031 & 0 & 0.004 & 0 \\ 0.048 & 0.004 & 0.165 & 0.11 & 0 & 0.036 & 0 & 0 & 0.028 & 0.010 \\ 0 & 0 & 0.085 & 0.075 & 0.001 & 0.030 & 0.037 & 0.001 & 0.022 & 0.017 \\ 0.06 & 0.003 & 0.065 & 0.048 & 0.013 & 0.009 & 0 & 0.023 & 0 & 0.012 \\ 0.005 & 0.025 & 0.13 & 0.017 & 0 & 0 & 0.001 & 0.024 & 0.001 & 0.019 \\ 0.02 & 0.026 & 0.112 & 0.002 & 0.006 & 0.032 & 0.000 & 0.016 & 0.007 & 0.010 \\ 0.024 & 0.033 & 0.123 & 0.007 & 0 & 0 & 0.001 & 0.019 & 0.007 & 0.015 \end{bmatrix}$$

10. Determination of composite distance: Finally, the composite distance value for each alternative, i.e., OOP languages, is calculated using equation step 6 of DBA.
11. Ranking of OOP Languages: The OOPLs ranked according to their composite distance values obtained in step 8. The OOP language having a minimum composite distance value is ranked at the top. The language with the most negligible composite distance value is ranked last, and other languages are ranked according to their respective composite distance values. The ranking of OOP languages obtained along with their composite distance values is provided in Table 3.

Table 3. Rankings of OOP languages obtained from FDBA

OOP Languages	Composite Distance	Rank	OOP Languages	Composite Distance	Rank
Smalltalk	0.7751	7	Python	0.4763	1
PHP	0.6367	6	C++	0.4872	3
C#	0.5224	5	Java	0.4828	2
Ruby	0.4881	4			

Methodolgy Validation

To validate implemented FDBA for OOP languages selection, two more well-known MCDM methods, namely, AHP and TOPSIS, are implemented on the same dataset. The ranking results obtained are compared with the results obtained from FDBA. The ranking of OOP languages obtained from AHP, TOPSIS, and FDBA and ranking differences is provided in Figure 3.

Further, Spearman's rank correlation test is also performed on the ranking results obtained from three MCDM methods: AHP, TOPSIS, and FDBA to check the significant relationship. The correlation test statistics are given in Table 4.

Figure 3. Comparative analysis of OOP language's rankings obtained from AHP, TOPSIS, and FDBA

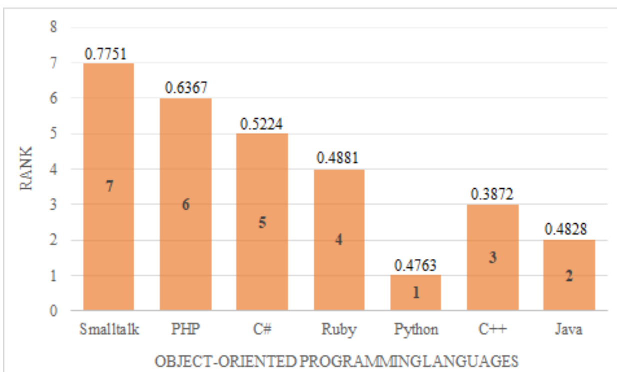


Table 4. Spearman's rank correlation test statistics

Ranking Differences	Squared Sum ($\sum D^2$)	Spearman's Rank Correlation Coefficient(R)
FDBA-AHP	20	0.6428
FDBA-TOPSIS	22	0.607

Results and Discussion

This study aims to implement a hybrid MCDM method, i.e., FDBA, to solve the present OOP language selection problem. The results obtained from the illustrated example to show the applicability and utility of the FDBA method are discussed below.

1. In this research, the data concerning the priority weights of the 10-selection attributes and the performance ratings of 7-OOP languages are collected by secondary means, i.e., questionnaires. A reliability test is done as explained in step-6 of the illustrated example in section-4. In this test, the value of Cronbach's alpha is calculated and is obtained as (0.889, 0.692) for the OOP language's performance ratings and (0.702, 0.602) for the selection parameter's priority weights. The calculated values of Cronbach's alpha are more significant than 0.5, affirming the high reliability of the data provided by the decision-makers through questionnaires.
2. The rankings of OOP languages in Table 3 depict that 'Python' is ranked at the top position, having the least composite distance value as 0.4763 followed by 'Java' language at rank-2 with composite distance value as 0.4828. 'Smalltalk' language is least preferred and is ranked at last position, i.e., rank-7 having maximum composite distance value as 0.7751 as compared to other programming languages. The ranking results also depict that 'Python' is a highly preferred OOP language for the students in the current competitive environment, whereas 'Smalltalk' is the least preferred.
3. The Figure 3 "Comparative analysis of OOP language's rankings obtained from AHP, TOPSIS, and FDBA" depicts some minor changes in the ranking of various languages as 'Rooby', 'Java' and 'Python' whereas the ranking of 'C#' is significantly changed. The ranking of 'C#' is obtained as '1', '5' and '1' from AHP, FDBA and TOPSIS respectively. The main reason behind such significant change is the non-consideration of criteria weights in the evaluation process followed by AHP and TOPSIS method. Simply, it is observed that there is no impact of criteria weights on the ranking results obtained from AHP and TOPSIS for any MCDM problem. So, there arises a need to validate the ranking results obtained from FDBA method. A novel attempt is made by performing Spearman's rank correlation test to validate the results obtained from the FDBA method, as given in table 3. Spearman's rank correlation coefficient is calculated to check the significant relationship between the ranking results obtained from FDBA, AHP, and TOPSIS. The value of this rank coefficient always lies between -1 to 1; the value closer to 1 shows a strong positive relationship, whereas the value closer to -1 shows a strong negative relationship between the two datasets. In the present case, Spearman's rank correlation coefficient is obtained as 0.6428 for FDBA-AHP and 0.6070 for FDBA-TOPSIS. Therefore, Spearman's rank correlation test statistics, as provided in table 8 affirm the solid positive significant relationship between the rankings of OOP languages obtained from FDBA, AHP, and TOPSIS.

Significance of Proposed Method (FDBA)

The advancement of computer technology is at such a fast pace that there is new progress every second. Prediction of technology performance and its evaluation has been a hot topic of research now a day. One of the primary measurement aspects is computer programming languages for engineering students. OOP Languages has the upper hand on procedural programming languages as this supports the code reusability based on the concept of objects. Immense diversity is present in the OOP language itself, making performance prediction of the latest trends inevitable. Despite the development of new OOP languages and changes in the newest programming languages, evaluation and selection of OOP languages remain an exquisite task for software organizations and aspiring computer programmers and software developers. The research presented here implements the fuzzy-DBA approach for the evaluation of major OOP languages worldwide by treating them as an MCDM problem. Although some other MCDM approaches like measurements of alternatives and rankings according to compromise solutions (MARKOS), Multi-attribute Border Approximation Area (MABAC), Multi Attribute Ideal Real Comparative Analysis (MAIRCA), VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR). Yet DBA approach possesses various significant advantages as

1. DBA is capable to accommodate the qualitative and quantitative attributes collectively.
2. The implantation of DBA method is very straight foreword based on simple matrix manipulations.
3. DBA is capable to handle the vast amount of selection parameters and alternatives.
4. MARKOS, MABAC, MAIRCA, VIKOR approaches work on the distance calculation from ideal and non-ideal solution. Here, ideal solution is based on all best values whereas non-ideal is based on all worst values corresponding to each criterion. Such distance calculation is from two solutions for each alternative is done in two steps whereas the DBA method integrates these into one step i.e., Euclidean distance calculation from optimal solution. Here, the optimal solution is the collection of either best value or worst values based on the nature of the criteria (cost or benefit).
5. Fuzzy set theory (FST) is used to obtain the weights of the selection parameters in this study instead of Best Worst method (BWM), Full Consistency method (FUCOM), AHP etc. These subjective methods for weight calculation become more complex in case of large number of selection parameters. Let us suppose, there are 'n' selection parameters in any decision-making problem, then (nxn) pairwise comparisons are required to get the criteria weight. The selection parameters weights in this study are calculated using the fuzzy operations as additions and averaging available in FST. These fuzzy operations are directly applied on the weights provided by the experts in linguistic terms after their conversion in triangular fuzzy numbers.

Conclusion

The study addressed the problem of selecting preferred object-oriented programming language as MCDM problem. This research developed a hybrid decision-making framework FDBA to model the problem. FDBA framework integrated the fuzzy set theory with a distance-based approach. FDBA method improves the decision-making process by handling ambiguity and uncertainty of data and accommodating many alternatives and the selection criteria. FDBA approach carries out the analysis with qualitative analysis to make subjective estimates more objective. According to FDBA, Python has the largest value due to its highest efficiency. However, SmallTalk has the smallest value and is ranked last in the programming language list. Spearman's rank correlation coefficient is calculated to check the significant relationship between the ranking results obtained from FDBA, AHP, and TOPSIS. This model helps in identification of the most preferrable object-oriented programming language in software industry and in education sector. The choice of programming languages is an inevitable necessity that helps organizations keep up with the latest trends and help select the most appropriate language, providing better opportunities in the business domain. The proper selection of the programming languages will undoubtedly result in high success for the students. The further enhancements of this work may be (i) performing 1-way or 2-way sensitivity analysis and (ii) the use of type-1 fuzzy sets, type-2 fuzzy sets, intuitionistic fuzzy sets, etc. (iii) computing weights using the level-based weight assessment (LBWA).

In future, the work can be extended by computing the weights using the level-based weight assessment (LBWA).

References

Amit, G., Ramesh, K., & Tewari, P. C. (2014). Ranking of inventory policies using distance based approach method. *International Journal of Mechanical and Mechatronics Engineering*, *8*(2), 395–400.

Anfurrutia, F. I., Álvarez, A., Larrañaga, M., & López-Gil, J. M. (2017). Visual Programming Environments for Object-Oriented Programming: Acceptance and Effects on Student Motivation. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, *12*(3), 124–131.

Bakır, M., & Atalık, Ö. (2021). Application of fuzzy AHP and fuzzy MARCOS approach for the evaluation of e-service quality in the airline industry. *Decision Making: Applications in Management and Engineering*, *4*(1), 127–152.

Bansal, A., Gupta, N., & Garg, R. (2022). Fuzzy multi-attribute decision-making approach for the selection of software effort estimation models. *International Journal of Advanced Intelligence Paradigms*, *21*(1-2), 174–188.

Batdalov, R., Ņikiforova, O., & Giurca, A. (2016). Extensible model for comparison of expressiveness of object-Oriented programming languages. *Applied Computer Systems*, *20*(1), 27–35.

Baykasoglu, A., & Durmusoglu, Z. D. (2014). A hybrid MCDM for private primary school assessment using DEMATEL based on ANP and fuzzy cognitive map. *International Journal of Computational Intelligence Systems*, *7*(4), 615–635.

Bibyan, R., & Anand, S. (2022). Ranking of Multi-release Software Reliability Growth Model Using Weighted Distance-Based Approach. In *Optimization Models in Software Reliability* (pp. 355–373). Springer.

Biswas, T. K., Abbasi, A., & Chakrabortty, R. K. (2021). An MCDM integrated adaptive simulated annealing approach for influence maximization in social networks. *Information Sciences*, *556*, 27–48.

Bosch, J. (1997). *Object-oriented frameworks: Problems & experiences.* Hogskolan Karlskrona Ronneby..

Chakraborty, S., Paul, D., & Agarwal, P. K. (2017). Evaluation of educational performance of Indian states using PROMETHEE-GIS approach. Benchmarking. *International Journal (Toronto, Ont.)*.

Chang, T. C., & Wang, H. (2016). A multi criteria group decision-making model for teacher evaluation in higher education based on cloud model and decision tree. Eurasia Journal of Mathematics. *Science and Technology Education*, *12*(5), 1243–1262.

Chiang, Y. H. (2015). MCDM modelling on learning achievement determinants of undergraduate students in Taiwan. *International Journal of Education Economics and Development*, *6*(2), 130–141.

Chodha, V., Dubey, R., Kumar, R., Singh, S., & Kaur, S. (2022). Selection of industrial arc welding robot with TOPSIS and Entropy MCDM techniques. *Materials Today: Proceedings*, *50*, 709–715.

Cook, S. (1986). Languages and object-oriented programming. *Software Engineering Journal*, *1*(2), 73–80. doi:10.1049/sej.1986.0013

Erdoğan, M., & Kaya, İ. (2014, July). A type-2 fuzzy MCDM method for ranking private universities in İstanbul. In *Proceedings of the World Congress on Engineering* (Vol. 1, pp. 2-4). IEEE.

Farshidi, S., Jansen, S., & Deldar, M. (2021). A decision model for programming language ecosystem selection: Seven industry case studies. *Information and Software Technology*, 106640.

Garg, R. (2017). Optimal selection of E-learning websites using multiattribute decision-making approaches. *Journal of Multi-Criteria Decision Analysis*, *24*(3-4), 187–196.

Garg, R. (2017). E-learning website evaluation and selection using multi-attribute decision making matrix methodology. *Computer Applications in Engineering Education*, *25*(6), 938–947.

Garg, R., & Jain, D. (2017). Fuzzy multi-attribute decision making evaluation of e-learning websites using FAHP, COPRAS, VIKOR, WDBA. *Decision Science Letters*, *6*(4), 351–364.

Garg, R., Kumar, R., & Garg, S. (2018). MADM-based parametric selection and ranking of E-learning websites using fuzzy COPRAS. *IEEE Transactions on Education*, *62*(1), 11–18.

Garg, R., Raheja, S., & Garg, R. K. (2021). Decision Support System for Optimal Selection of Software Reliability Growth Models Using a Hybrid Approach. *IEEE Transactions on Reliability*, *71*(1), 149–161.

Garg, R., Sharma, R., & Sharma, K. (2016). Ranking and selection of commercial off-the-shelf using fuzzy distance based approach. *Decision Science Letters*, *5*(2), 201–210.

Ghorui, N., Ghosh, A., Mondal, S. P., Kumari, S., Jana, S., & Das, A. (2021). Evaluation of performance for School Teacher Recruitment using MCDM techniques with Interval Data. *Multicultural Education*, *7*(5).

Goswami, S. S. S., Behera, D. K. K., Afzal, A., Razak Kaladgi, A., Khan, S. A. A., Rajendran, P., & Asif, M. (2021). Analysis of a robot selection problem using two newly developed hybrid MCDM models of TOPSIS-ARAS and COPRAS-ARAS. *Symmetry*, *13*(8), 1331.

Gürbüz, T., & Albayrak, Y. E. (2014). An engineering approach to human resources performance evaluation: Hybrid MCDM application with interactions. *Applied Soft Computing*, *21*, 365–375.

Handoko, D., Mesran, M., Nasution, S. D., Yuhandri, Y., & Nurdiyanto, H. (2017). Application Of Weight Sum Model (WSM) In Determining Special Allocation Funds Recipients. *The IJICS (International Journal of Informatics and Computer Science), 1*(2).

Hasan, L. F., Ibrahim, A. A. E. A., Bin, S. Z., & Idrus, S. (2022). *Make the Educational Decisions Using the Multi-Criteria Decision-Making MCDM in the Light of the Corona Pandemic.* PMC.

Henderson, R., & Zorn, B. (1994). A comparison of object-oriented programming in four modern languages. *Software, Practice & Experience*, *24*(11), 1077–1095. doi:10.1002/spe.4380241106

İmamoğlu, M. Y., & Çetinkaya, D. (2017, October). A rule based decision support system for programming language selection. In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)* (pp. 71-75). IEEE.

Jain, D., Garg, R., & Bansal, A. (2015). A parameterized selection and evaluation of e-learning websites using TOPSIS method. *International Journal of Research & Development in*, *22*(3), 12–26.

Jain, D., Garg, R., Bansal, A., & Saini, K. K. (2016). Selection and ranking of E-learning websites using weighted distance-based approximation. *Journal of Computers in Education*, *3*(2), 193–207.

Karamasa, Çaglar, Demir, E., Memis, S., & Korucuk, S. (2021). Weighting the factors affecting logistics outsourcing. *Decision Making: Applications in Management and Engineering*, *4*(1), 19–32.

Karmaker, C., & Saha, M. (2015). Teachers' recruitment process via MCDM methods: A case study in Bangladesh. *Management Science Letters*, *5*(8), 749–766.

Kumar, R., & Garg, R. K. (2010). Optimal selection of robots by using distance based approach method. *Robotics and Computer-integrated Manufacturing*, *26*(5), 500–506.

Lesani, S. H., Rouyendegh, B. D., & Erdebilli, B. (2014, June). Object-oriented programming language selection using fuzzy AHP method. In *annual meeting of the ISAHP* (*Vol. 29*, pp. 1-17). IEEE.

Maplesden, D., Tempero, E., Hosking, J., & Grundy, J. C. (2015). Performance analysis for object-oriented software: A systematic mapping. *IEEE Transactions on Software Engineering*, *41*(7), 691–710.

Mardani, A., Zavadskas, E. K., Khalifah, Z., Zakuan, N., Jusoh, A., Nor, K. M., & Khoshnoudi, M. (2017). A review of multi-criteria decision-making applications to solve energy management problems: Two decades from 1995 to 2015. *Renewable & Sustainable Energy Reviews*, *71*, 216–256. doi:10.1016/j.rser.2016.12.053

Mishra, A. R., Chandel, A., & Motwani, D. (2020). Extended MABAC method based on divergence measures for multi-criteria assessment of programming language with interval-valued intuitionistic fuzzy sets. *Granular Computing*, *5*(1), 97–117.

Muhammad, S., Maqbool, O., & Abbasi, A. Q. (2012). Evaluating relationship categories for clustering object-oriented software systems. *IET Software*, *6*(3), 260–274.

Onu, F. U., Uche-Nwachi, E. O., & Chigbundu, K. E. (2016). The Role of Object-Oriented Programming (OOP) in Modeling of Geographic Information Systems (GIS). *IOSR Journal of Mobile Computing & Application (IOSR-JMCA),* 40-46.

Özdemir, A., & Tüysüz, F. (2017). An integrated fuzzy DEMATEL and fuzzy ANP based balanced scorecard approach: application in Turkish higher education institutions. *Journal of Multiple-Valued Logic & Soft Computing, 28*.

Pamučar, D., & Ćirović, G. (2015). The selection of transport and handling resources in logistics centers using Multi-Attributive Border Approximation area Comparison (MABAC). *Expert Systems with Applications*, *42*(6), 3016–3028.

Parker, K. R., Chao, J. T., Ottaway, T. A., & Chang, J. (2006). A formal language selection process for introductory programming courses. *Journal of Information Technology Education: Research*, *5*(1), 133–151.

Sandhya, S., & Garg, R. (2016). Implementation of multi-criteria decision making approach for the team leader selection in IT sector. *Journal of Project Management*, *1*(2), 67–75.

Yadav, N., Saraswat, P., & Tripathi, R. P. (2017, December). Estimating the functionality of object oriented system using MCDM approach. In *2017 Fourth International Conference on Image Information Processing (ICIIP)* (pp. 1-6). IEEE.

Yıldızbaşı, A., & Daneshvar Rouyendegh, B. (2018). Multi-criteria decision making approach for evaluation of the performance of computer programming languages in higher education. *Computer Applications in Engineering Education*, *26*(6), 1992–2001.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, *8*(3), 199–249.

Appendix-1

Priority weights of selection attributes provided by decision makers

Criteria/ Decision Makers	E1	E2	E3	E4	E5
C_1	EMI(1,1,1)	MI(0.6,0.7,0.8)	VMI(0.8,0.9,1)	EMI(1,1,1)	EMI(1,1,1)
C_2	VMI(0.8,0.9,1)	EMI(1,1,1)	VMI(0.8,0.9,1)	MI(0.6,0.7,0.8)	MI(0.6,0.7,0.8)
C_3	LI(0.2,0.3,0.4)	VMI(0.8,0.9,1)	VMI(0.8,0.9,1)	EMI(1,1,1)	MI(0.6,0.7,0.8)
C_4	VMI(0.8,0.9,1)	VMI(0.8,0.9,1)	VMI(0.8,0.9,1)	MI(0.6,0.7,0.8)	LI(0.2,0.3,0.4)
C_5	MI(0.6,0.7,0.8)	EMI(1,1,1)	LI(0.2,0.3,0.4)	EMI(1,1,1)	MI(0.6,0.7,0.8)
C_6	LI(0.2,0.3,0.4)	MI(0.6,0.7,0.8)	MI(0.6,0.7,0.8)	I(0.4,0.5,0.6)	I(0.4,0.5,0.6)
C_7	MI(0.6,0.7,0.8)	LI(0.2,0.3,0.4)	LI(0.2,0.3,0.4)	I(0.4,0.5,0.6)	I(0.4,0.5,0.6)
C_8	EMI(1,1,1)	MI(0.6,0.7,0.8)	LI(0.2,0.3,0.4)	I(0.4,0.5,0.6)	LI(0.2,0.3,0.4)
C_9	LI(0.2,0.3,0.4)	MI(0.6,0.7,0.8)	VLI(0.0,0.1,0.2)	I(0.4,0.5,0.6)	VLI(0.0,0.1,0.2)
C_{10}	VLI(0.0,0.1,0.2)	VLI(0.0,0.1,0.2)	MI(0.6,0.7,0.8)	MI(0.6,0.7,0.8)	LI(0.2,0.3,0.4)

Appendix-2

Performance ratings of OOP languages provided by decision makers

Languages	Criteria	D_1	D_2	D_3	D_4	D_5
SmallTalk	C_1	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)	L(0.0,0.1,0.2)	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)
	C_2	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)	L(0.0,0.1,0.2)
	C_3	H(0.8,0.9,1)	L(0.0,0.1,0.2)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	BA(0.2,0.3,0.4)
	C_4	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	L(0.0,0.1,0.2)
	C_5	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)
	C_6	L(0.0,0.1,0.2)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)	BA(0.2,0.3,0.4)
	C_7	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)
	C_8	BA(0.2,0.3,0.4)	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)
	C_9	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	L(0.0,0.1,0.2)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)
	C_{10}	VH(1,1,1)	VH(1,1,1)	VH(1,1,1)	VH(1,1,1)	VH(1,1,1)
PHP	C_1	BA(0.2,0.3,0.4)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)
	C_2	H(0.8,0.9,1)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)
	C_3	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	L(0.0,0.1,0.2)
	C_4	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	L(0.0,0.1,0.2)	BA(0.2,0.3,0.4)
	C_5	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	H(0.8,0.9,1)	L(0.0,0.1,0.2)
	C_6	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	L(0.0,0.1,0.2)
	C_7	BA(0.2,0.3,0.4)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)
	C_8	H(0.8,0.9,1)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)
	C_9	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)
	C_{10}	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)

Table continued on next page

Table continued

Languages	Criteria	D_1	D_2	D_3	D_4	D_5
C#	C_1	VH(1,1,1)	H(0.8,0.9,1)	H(0.8,0.9,1)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)
	C_2	VH(1,1,1)	VH(1,1,1)	VH(1,1,1)	H(0.8,0.9,1)	H(0.8,0.9,1)
	C_3	H(0.8,0.9,1)	A(0.4,0.5,0.6)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)
	C_4	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)
	C_5	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	AA(0.6,0.7,0.8)
	C_6	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	AA(0.6,0.7,0.8)	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)
	C_7	BA(0.2,0.3,0.4)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)
	C_8	VH(1,1,1)	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)	H(0.8,0.9,1)
	C_9	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	BA(0.2,0.3,0.4)
	C_{10}	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)
Ruby	C_1	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)
	C_2	VH(1,1,1)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	VH(1,1,1)	H(0.8,0.9,1)
	C_3	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)
	C_4	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)
	C_5	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)
	C_6	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)
	C_7	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)
	C_8	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)
	C_9	VH(1,1,1)	VH(1,1,1)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)
	C_{10}	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)
Python	C_1	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	VH(1,1,1)	A(0.4,0.5,0.6)
	C_2	VH(1,1,1)	VH(1,1,1)	A(0.4,0.5,0.6)	H(0.8,0.9,1)	A(0.4,0.5,0.6)
	C_3	A(0.4,0.5,0.6)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)
	C_4	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	VH(1,1,1)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)
	C_5	A(0.4,0.5,0.6)	H(0.8,0.9,1)	VH(1,1,1)	VH(1,1,1)	AA(0.6,0.7,0.8)
	C_6	AA(0.6,0.7,0.8)	A(0.4,0.5,0.6)	H(0.8,0.9,1)	A(0.4,0.5,0.6)	H(0.8,0.9,1)
	C_7	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	A(0.4,0.5,0.6)	BA(0.2,0.3,0.4)	H(0.8,0.9,1)
	C_8	H(0.8,0.9,1)	A(0.4,0.5,0.6)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)
	C_9	AA(0.6,0.7,0.8)	VH(1,1,1)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	VH(1,1,1)
	C_{10}	VH(1,1,1)	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)	VH(1,1,1)
C++	C_1	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)	H(0.8,0.9,1)
	C_2	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)	H(0.8,0.9,1)	VH(1,1,1)
	C_3	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)
	C_4	H(0.8,0.9,1)	H(0.8,0.9,1)	H(0.8,0.9,1)	H(0.8,0.9,1)	VH(1,1,1)
	C_5	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)
	C_6	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)	VH(1,1,1)
	C_7	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)
	C_8	VH(1,1,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)	H(0.8,0.9,1)
	C_9	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)
	C_{10}	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	AA(0.6,0.7,0.8)

Table continued on next page

Table continued

Languages	Criteria	D_1	D_2	D_3	D_4	D_5
Java	C_1	VH(1,1,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)
	C_2	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)	VH(1,1,1)
	C_3	H(0.8,0.9,1)	AA(0.6,0.7,0.8)	VH(1,1,1)	VH(1,1,1)	AA(0.6,0.7,0.8)
	C_4	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)	H(0.8,0.9,1)	VH(1,1,1)
	C_5	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	VH(1,1,1)	VH(1,1,1)	H(0.8,0.9,1)
	C_6	VH(1,1,1)	AA(0.6,0.7,0.8)	VH(1,1,1)	H(0.8,0.9,1)	VH(1,1,1)
	C_7	AA(0.6,0.7,0.8)	VH(1,1,1)	AA(0.6,0.7,0.8)	VH(1,1,1)	H(0.8,0.9,1)
	C_8	VH(1,1,1)	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)	VH(1,1,1)
	C_9	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)	VH(1,1,1)	H(0.8,0.9,1)
	C_{10}	AA(0.6,0.7,0.8)	H(0.8,0.9,1)	H(0.8,0.9,1)	H(0.8,0.9,1)	VH(1,1,1)

Rakesh Garg received the B.E., M.Tech. and the Ph.D. degree all in computer science and engineering. He is currently an Associate Professor with the Department of Computer Science and Engineering, Amity University, Noida, India. He has authored or co-authored a number of research papers in reputed international journals with various publishing houses such as IEEE, Springer, and Wiley, and guided a number of doctoral and master's degree dissertations. His major publications include articles published in the IEEE Transactions on Reliability, IEEE Transactions on Education, IEEE Transactions on Cloud Computing, IEEE Transactions on Engineering Management, Computer Applications in Engineering Education, and Journal of Multi-Criteria Decision Analysis. His research interests include component-based software engineering, software reliability, and multicriteria decision-making optimization.

Supriya Raheja, is currently working as a Senior Assistant Professor in the Department of CSE & IT, School of Engineering & Technology, NCU. She has more than 10 years of extensive teaching experience at both post and under graduate level. She is a committed researcher in the field of Operating system, fuzzy set theory and vague set theory and has completed her PhD in the same area. Her areas of interest include Procedure Oriented Programming, Object Oriented Programming, Secure Coding & vulnerabilities, Operating System, Networks and Fuzzy Set Theory. She had guided 10 M.Tech Projects and more than 25 B.Tech Projects. She has published more than 25 papers in peer reviewed reputed International Journals and in IEEE/Springer International Conferences. She is the reviewer and assistant editor of various international journals. She has done certification in Software Security from University of Maryland.