Selection of Learning Apps to Promote Critical Thinking in Programming Students using Fuzzy TOPSIS

Kesarie Singh[1], Nalindren Naicker[2], Mogiveny Rajkoomar[3]
Department of Information Technology, Durban University of Technology, Durban, South Africa[1]
Department of Information Systems, Durban University of Technology, Durban, South Africa[2, 3]

***Abstract*—The aim of this research was to use intelligent decision support systems to obtain student-centred preferences for learning applications to promote critical thinking in first year programming students. This study focuses on the visual programming environment and critical thinking as the gateway skill for student success in understanding programming. Twenty-five critical thinking criteria were synthesized from the literature. As a quantitative study, 217 randomly selected students from an approximate target population of 500 programming students to rate four learning Apps, namely, Scratch, Alice, Blockly and MIT App Inventor, against critical thinking criteria to establish the App that best promotes critical thinking among first year programming students. There were 175 responses received from the 217 randomly chosen programming students who willingly contributed to the study. Consequently, the distinctiveness of this paper lies in its use of the Fuzzy TOPSIS (Technique for Order Preference by Similarity to Ideal Situation) multi-criteria decision-making algorithm to rank criteria for critical thinking, calculate their weights on the basis of informed opinion and hence scientifically deduce the best rated App among the available alternatives that promote critical thinking among first year programming students. The results showed that Scratch promoted critical thinking skills the best in first year programming students whilst Blockly promoted critical thinking skills the least. As a contribution to the study, policy-makers and academic staff can be potentially supported to make informed decisions about the types of learning Apps to consider for students when confronted with multiple selection criteria.**

Keywords*—Critical thinking; visual programming environment; multi-criteria decision-making; fuzzy TOPSIS

I. Introduction

The problem associated with teaching programming to novice learners is exacerbated by the complex and abstract nature of the field and the heavy reliance on 21st century skills such as critical and computational thinking. As a result, a kaleidoscope of research into programming self-efficacy, the complexity of the field, teaching methods and a variety of teaching tools, have emerged over the recent past. Furthermore, learning to program can be perceived as both tedious and difficult to the novice programmer for reasons such as the abstract nature of computational thinking and the burdensome syntax and semantics associated with many programming languages [1-3]. The literature has also repeatedly revealed that visual block-based programming environments have a positively powerful impact on the performances of novices learning to program [4]. In a case study performed by Pinto-Llorente et al. in [5] on the impact of the visual programming environment (VPE) among primary school learners, it was determined to have promoted their critical thinking skills and problem solving abilities. Various other studies have revealed that visual programming environments support the constructivist approach to learning and contribute to enhancing one's skills of independent learning, creative thinking, problem-solving, reflection and collaboration [6-12]. This has resulted in an upsurge of educational programming environments and tools aimed at stimulating the learners' interest, making programming accessible to people of all interests and ages and minimizing the challenges of understanding programming [13].

The significant growth in diverse learning Apps in the last decade and the varying commercial aspirations these Apps may be designed to serve, have impacted on the need for university academics to make rigorous and well-considered decisions around their choice of learning App. Further, an online learning App becomes especially appropriate post Covid-19, where higher education institutions are strategically searching for alternate teaching methodologies to the traditional teaching approach. Interestingly, a number of learning Apps originated over the years with a variety of different characteristics, each serving a slightly different need [4]. Multi-criteria group decision-making (MCDM) entails weighing each of several alternatives against multiple, conflicting, quantitative decision criteria executed by many decision-makers to derive the optimal alternative [14-16]. The various criteria will be synthesized from an in-depth analysis of the literature in the relevant field.

According to Başaran and Haruna in [17], manually selecting these Apps can be biased, tedious, inconsistent and time-consuming, ultimately resulting in a premature selection. Therefore, to address ambiguity and uncertainty associated with human judgements, this quantitative study will use the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS), a scientific and mathematical technique in multi-criteria decision-making (MCDM) that preserves integrity and objectivity of process [16, 18]. The TOPSIS method, developed in 1981 by Yoon and Hwang, was grounded on the fundamental principle that the best option from a range of alternatives will be positioned closest to the positive ideal solution and farthest from the negative ideal solution [19].

In a study conducted by Junior, Osiro and Carpinetti in [39], a comparison of the Fuzzy AHP and Fuzzy TOPSIS methods was undertaken to choose from a number of suppliers. The criteria central to the comparison were: “adequacy to changes of alternatives or criteria; agility in the decision process; computational complexity; adequacy to supporting group decision-making; the number of alternatives and criteria, and modelling of uncertainty”. The study revealed that Fuzzy TOPSIS provided more rigor in the decision process, lower time complexity when there was a large number of decision-makers, offered no restrictions on the number of criteria and alternatives and became the preferred choice when there were changes to alternatives or criteria as rank reversal posed a potential problem with Fuzzy AHP.

Fuzzy set theory, which was introduced by Zadeh in 1965, can be used to manage the uncertainties and subjectivity of the evaluation process [15]. More specifically, this study will implement an expansion of the TOPSIS method to accommodate fuzziness in the decision problem setting. Fuzzy TOPSIS is widely and extensively applied among researchers in diverse fields of economics, business management, engineering, ICT and education, to solve various MCDM problems associated with site selection in mining and engineering, equipment selection, risk analysis for complex infrastructure projects, portfolio selection, selection of e-Learning approaches, supplier selection and software selection [16-23]. The advantages of Fuzzy TOPSIS include its ease of implementation, capacity to compare the strongest and weakest alternatives quantitatively, that it enables linguistic expressions to be represented as fuzzy numbers, and its practicality and ability to handle incomplete or partial quantitative data [24]. A questionnaire disseminated to 217 first year programming students is used as a tool to gather the ratings of alternatives against multiple selection criteria, and the different weightings per criteria are obtained from academic experts in the field. These sets of evaluations use logical interval judgements represented by linguistic terms which can be parameterized by triangular fuzzy numbers [24].

Choosing an optimal learning application (App) is therefore a complex problem requiring an intelligent decision support system that will use multiple decision-makers to simultaneously ruminate multiple criteria such as the prior skills of the learner, the technical infrastructure of the learning environment and the 21st century skills required for success in the higher order thinking of computer programming [25]. There are no studies in the extant literature that use an intelligent decision support system that is able to find the optimal choice of learning App to suit the needs of the instructors and learners in acquiring improved student success not only to encourage their interest and excitement for programming, but also to acquire a proficiency to create their own applications. The aim of this paper is to use the fuzzy TOPSIS multi-criteria decision-making technique to rank learning Apps according to its potential to promote critical thinking among programming students. The paper comprises five sections, namely, section 1 introduces the topic, section two surveys the extant literature, section three presents the fuzzy TOPSIS method, section four explains the results and section five culminates with the findings of the study.

II. Literature Review

The definition and promotion of critical thinking for learning has become increasingly important in the 21st century information and digital age, where organizations are competing for applicants who have the ability to investigate, apply, and transform data, to collaborate and innovate, to reflect, analyze and think critically, and to arrive at optimal decisions [26]. The literature defines the higher order thinking processes associated with programming as closely aligned with the concept of computational thinking (CT), where problem-solving is implemented through the mental process of abstraction, logic, analysis, synthesis and constructive thinking [5, 27-28]. It has been justified through various studies that CT skills should not just be accessible to everyone at a much younger age, but also have the potential to be applied to various disciplines and hence to be incorporated into the basic curriculum [2, 5].

There are a variety of learning resources and learning Apps to develop CT [29, 30] and each of these include aspects of robotics [2, 4, 31-32], gaming [33-34], augmented reality simulation [35] or multimedia such as graphics, sound and animation, with the latter referred to as visual programing environments [4, 9, 28]. This study is significant in its application as it aims to extend multi-criteria decision-making using Fuzzy TOPSIS with multiple decision-makers choosing the most appropriate learning App based on well-synthesized critical thinking criteria.

Multi-criteria decision analysis (MCDA) has successfully been applied to solve a wide range of complex real-life decision problems in a variety of fields [36], despite researchers encountering many challenges associated with partial ignorance and unquantifiable, incomplete, unobtainable, uncertain, ambiguous and vague information [15]. The choice of MCDA technique rests predominantly on the nature of the decision problem and its ability to address many of the afore-mentioned challenges. The fuzzy approach is best suited to scenarios that are prone to ambiguity, subjectivity and uncertainty when the decision-maker is expected to make an evaluation [18, 37]. Furthermore, the TOPSIS method suggested by Hwang and Yoon in 1981, is a popular method that can rank the best alternatives quickly, address conflicting situations, is easy to use and can be integrated with other decision-making methods [18]. The fuzzy set theory combined with TOPSIS has the added advantage of addressing any uncertainty encompassing the assessment process [15].

Several refined MCDA methods have been introduced over the years. Some of the popular methods include the analytic hierarchy process (AHP) requiring many comparisons based on human preferences, with the possibility of compromising the consistency of the decisions when compared to the best-worst method developed to reduce the number of comparisons in traditional AHP; the TOPSIS method and simple additive weighting (SAW), which depend on mathematical operations, and the outranking models like ELECTRE and PROMETHEE, which are premised on the assumption that the decision is a process whereby decision-makers can alter their preferences after thorough reflection. The hierarchical model of the AHP method requires the decision-maker to pairwise-compare multiple criteria and uses ratio and semantic scales to arrive at the decision-maker's preference. Memari et al. in [21] compared the TOPSIS method with two outranking methods namely, PROMETHEE and ELECTRE. According to the authors, TOPSIS is easy to learn and apply compared to outranking methods which proved to be more complex and less transparent to decision-makers. Sahin et al. in [38] included TOPSIS in their study due to its unique approach to the problem and it's intuitively appealing and easy to understand qualities. Palczewski and Sałabun in [36] further highlighted simplicity, computational efficiency and comprehensive mathematical concept as well as its support for group decision-making as key contributing factors to the popularity of the Fuzzy TOPSIS technique.

In the learning App selection problem, the multi-criteria decision-making approach to be applied needs to be carefully considered. The number of decision-makers, the various types of measures to promote critical thinking, the method used to weight the various criteria and the degree of uncertainty and vagueness encompassing the problem scenario are some of the qualifying factors. This study involves a sample of 217 first year programming students and questions on the performance of each alternative against multiple critical thinking criteria, some more easily measurable than others. Nursal, Omar and Nawi in [14], identify the TOPSIS method with assigning crisp or exact numerical values when rating alternatives against criteria and when rating criteria importance, which are also inadequate to simulate human judgements in real-life scenarios. Therefore, to accommodate the imprecise or vague nature of the group decision-making assessment pertaining to critical thinking criteria, this study will use a more authentic approach by enabling the use of linguistic variables to capture the decision-maker's rating more accurately and consistently [40]. Hence the Fuzzy TOPSIS approach is being more suitably applied to model the linguistic ratings as a fuzzy triplet (a, b, c) because the most promising value b will better capture the rating compared to parameter *a* representing the smallest possible value and parameter *c* representing the largest possible value in a fuzzy event [40, 41]. This becomes especially important in the group decision-making process.

The research has demonstrated various MCDM problems being solved through the application of fuzzy logic. These include facility location selection, machine tool selection problem, plant layout design problem and robot selection [42] to equipment selection [16] and supplier selection [21, 36, 40]. Although this study uses the quantitative approach, it also identifies some subjective criteria that are difficult to measure, hence the need for fuzzy sets which are capable of representing vague data [41]. According to Kahraman et al. in [42], the strength of fuzzy logic lies in its ability to imitate human reasoning capabilities during the cognitive process of decision-making, to be captured mathematically and more precisely, resulting in a better expected performance in this case scenario.

In everyday pragmatic circumstances which are frequently unpredictable, it is often inadequate to describe phenomena in crisp and precise terms. Therefore, fuzziness occurs in many areas of human judgement, reasoning, evaluations and decision-making. In this study, for example, the evaluation of learning Apps against various subjective critical thinking criteria are more easily articulated in natural language terms such as very good, poor or satisfactory, which may affect vagueness and ambiguity. However, fuzzy set theory, introduced by Zadeh (1965), offers a rigorous mathematical framework to efficiently resolve the indistinctness associated with the subjectivity of human judgements, and in which vagueness and uncertainty can be precisely and rigorously studied [43].

Rashidi and Cullinane in [40] highlighted fuzzy sets and fuzzy logic (FL) as "powerful mathematical tools for modelling uncertain systems in industry", while de Barros et al. [44] define fuzzy logic as a "multivalued logic, that uses intermediary values between conventional evaluations such as true/false, yes/no or high/low". According to de Barros et al. [44], the language of linguistics is helpful in describing uncertain and ill-defined qualitative data, which otherwise cannot be subjected to quantitative analysis. These linguistic variables can then be represented as computational-efficient fuzzy-triples for appropriately quantifying vague information [37].

According to Samaie et al. in [37], elements in a fuzzy set have different degrees of membership, designated by membership functions, which allocates a grade of membership ranging between 0 and 1. In applications that use fuzzy techniques the decision problem can have complex mathematical membership functions such as triangular or trapezoidal. According to Rajak and Shaw in [18], modelling the decision problem using triangular fuzzy numbers yields a better result. The graphical representation of a membership function F(x) of a triangular fuzzy number Ã, is illustrated in Fig. 1 where the x axis represents the universe of discourse and the y axis represents the degrees of membership in the [0,1] interval [37, 40].

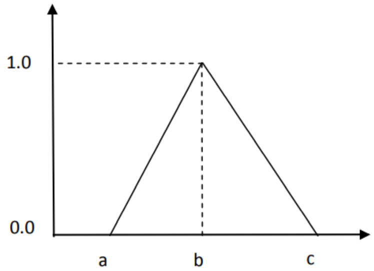

Fig. 1. A Membership Function F(X) of a Triangular Fuzzy Number ã (Han and Trimi 2018).

A triangular fuzzy number is defined in Fig. 1 and expressed as a membership function in Fig. 2.

$$F(x) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & otherwise \end{cases}$$

Fig. 2. Membership Function of a Triangular Fuzzy Number (Han and Trimi 2018).

TABLE I. LINGUISTIC TERMS FOR CRITERIA AND ALTERNATIVES RATINGS (HAN AND TRIMI 2018)

Criteria	Alternatives	
	Linguistic term	Triangular fuzzy numbers
Very low (VL)	Very poor (VP)	(1,1,3)
Low (L)	Poor (P)	(1,3,5)
Medium (M)	Fair (F)	(3,5,7)
High (H)	Good (G)	(5,7,9)
Very high (VH)	Very good (VG)	(7,9,9)

In fuzzy set theory, linguistic terms are converted into fuzzy triples using conversion scales. Usually, a scale of 1 to 9 is applied for weighting the criteria and ranking the alternatives, as illustrated in Table I. Uniform intervals are used to represent the fuzzy triangular numbers for each of the five linguistic ratings [23].

III. METHODOLOGY

In this research paper fuzzy set theory is introduced to model ambiguity and uncertainty in a MCDM problem and integrated with TOPSIS to appreciate the benefits of its practicality and ease of use, enabling evaluations to be expressed in a linguistic language and then converted to triangular fuzzy numbers and implementing the algorithm using a software tool.

Assume the decision problem has k decision-makers (D$_1$, D$_2$ D$_k$), with m possible alternatives (A$_1$, A$_2$ A$_m$), which is evaluated against n criteria (C_1, C_2 … C_n). The rating of criteria and weight with respect to each criterion can be accurately represented in the form of matrices for each decision-maker.

$$\widetilde{D} = \begin{array}{c} \\ A_1 \\ A_2 \\ \\ A_m \end{array} \begin{array}{c} C_1 \quad C_2 \quad\quad C_n \\ \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1n} \\ \tilde{x}_{21} & \tilde{x}_{22} & \dots & \tilde{x}_{2n} \\ \dots & \dots & \tilde{x}_{ij} & \dots \\ \tilde{x}_{m1} & \tilde{x}_{m2} & \dots & \tilde{x}_{mn} \end{pmatrix} \end{array} \qquad (1)$$

$$\widetilde{W} = (\widetilde{w}_1, \widetilde{w}_2, \dots, \widetilde{w}_n)$$

Where for all x_{ij} and w_{ij} i = 1, 2 … m and j=1, 2 … n.

Hence $x_{ij} = (a_{ij},\ b_{ij},\ c_{ij})\ and\ w_j = (a_j, b_j, c_j)$ are triangular fuzzy numbers representing linguistic variables.

The Fuzzy TOPSIS procedure includes the following steps [24]:

Step 1: Assign ratings to the criteria and alternatives. The criteria weights are denoted by

W_{NK} (N = 1,2,…, n; K = 1,2,…, k)

and the performance ratings of alternatives with respect to criteria by experts are denoted as

X_{NKM} (N = 1,2,…, n; K = 1,2,…, k; M = 1,2,…, m).

Step 2: Aggregate the evaluation of the criteria and alternatives. fuzzy ratings W_{NK} and X_{NKM} is described as triangular fuzzy numbers (a_K, b_K, c_K) where K = 1,2,…, k, then the aggregated importance can be evaluated as:

$$a = min\{a_K\}\ \ b = \frac{1}{K}\sum_{K=1}^{K} b_K\ \ c = max\{c_K\} \qquad (2)$$

Step 3: Normalize triangular fuzzy numbers. The raw data is normalized using linear scale transformation to bring the various criteria scales into a comparable scale. If W_{NK} represents benefit criteria, then:

$$\left(\frac{a_K}{C}, \frac{b_K}{C}, \frac{c_K}{C}\right) \qquad (3)$$

If W_{NK} represents cost criteria, then:

$$\left(\frac{a}{c_K},\ \frac{a}{b_K},\ \frac{a}{a_K}\right) \qquad (4)$$

Where $(a = min\{a_K\}, c = max\{c_K\})$

Step 4: Compute weighted normalized fuzzy values. $W_{NK}*$ becomes W_{NK} after normalization, $X_{NKM}*$ is new X_{NKM} after aggregation. Let the weighted normalized value be V_{NKM} .

$$V_{NKM} = W_{NK} * \text{x}\ X_{NKM} * \qquad (5)$$

where N = 1,2 … n; K = 1,2 … k; M = 1,2 … m;

The corresponding triangular fuzzy number of V_{NKM} is

$$\left(a_{V_{NKM}},\ b_{V_{NKM}},\ c_{V_{NKM}}\right) \qquad (6)$$

Step 5: Calculate fuzzy positive ideal solutions (FPIS) and fuzzy negative ideal solutions (FNIS);

$$FPIS = (C_{V,}\ C_{V,} C_V)\ where\ C_V = \max\{C_{V_{NKM}}\} \qquad (7)$$

$$FNIS = (a_{V,}\ a_{V,} a_V)\ where\ a_V = \min\{a_{V_{NKM}}\} \qquad (8)$$

Step 6: Calculate the distance of each alternative from $FPIS\ (d^+)$ and $FNIS(d^-)$ which is calculated, respectively, as follows:

$$d^+ = \sqrt{\frac{1}{3}}\left[\left(a_{V_{NKM}} - c_V\right)^2 + \left(b_{V_{NKM}} - c_V\right)^2 + \left(c_{V_{NKM}} - c_V\right)^2\right] \qquad (9)$$

$$d^- = \sqrt{\frac{1}{3}}\left[\left(a_{V_{NKM}} - a_V\right)^2 + \left(b_{V_{NKM}} - a_V\right)^2 + \left(c_{V_{NKM}} - a_V\right)^2\right] \qquad (10)$$

Step 7: Calculate the closeness coefficient (CC$_M$)

$$CC_M = \frac{\sum_1^n d_{NM}^-}{\sum_1^n d_{NM}^+ + \sum_1^n d_{NM}^-}, N = 1,2 \dots n; M = 1,2 \dots m \qquad (11)$$

The CC_M value is then used to determine the ranking order of all alternatives for the purpose of selecting the best one from among a set of feasible alternatives.

IV. RESULT

In this section, the results produced from the Fuzzy TOPSIS software application implemented on MatLab R2020a is presented with explanations in steps according to input and outputs of the application. This application utilized complex mathematical equations on the backend which corresponds with the Fuzzy TOPSIS method presented in the methodology section. The results are based on preferences shown by decision-makers (first year programming students) on choice of visual programming learning applications based on set critical thinking criteria.

Table II shows the fuzzy rating scale where linguistic terms, namely, Not Important (N), Slightly Important (SI), Moderately Important (MI), Important (I) and Very Important (VI) are expressed as a fuzzy triple using integers in the range 1 to 9. These linguistic terms are used to rate the criteria.

Table III presents the measures for critical thinking (criteria) synthesized from the extant literature. Symbols C1 to C25 are used to represent the critical thinking criteria for selection of programming learning applications for first year university students.

TABLE II. LINGUISTIC AND FUZZY TRIPLE FOR RATING CRITERIA

Linguistic term	Membership function
Not Important (N)	(1,1,3)
Slightly Important (SI)	(1,3,5)
Moderately Important (MI)	(3,5,7)
Important (I)	(5,7,9)
Very Important (VI)	(7,9,9)

TABLE III. MEASURES FOR CRITICAL THINKING (CRITERIA)

No.	Critical Thinking Criteria
C1	Feedback: The App helped me correct my errors while I was coding
C2	Feedback: The App alerted me to incorrect code
C3	Interactivity: The App tells me where my error lies
C4	Problem-solving: The App allows me to solve large, complex, real world, authentic problem scenarios
C5	Collaboration: The App allows me to work on a shared program with my friends
C6	Collaboration: The App allows me to re-use my code or re-use my peers’ solutions
C7	Collaboration: The App allows me to communicate with my friend about my questions and queries about our projects
C8	Metacognition: The App allows me to easily and repeatedly make changes to my solution
C9	Logic: The App helped me to improve my logic skills
C10	Logic and reasoning: The App quickly alerts me when the sequencing of steps in my solution is logically incorrect
C11	Evaluation: The App helps me to make my code more efficient
C12	Evaluation: The App is able to evaluate my work
C13	Evaluation: The App allows me to compare my solution against my peers
C14	Alternate solutions: The App helps me to think about solving the problem in different ways
C15	Synthesis: The App allows me to solve problems where I have to draw my knowledge from different programming topics
C16	Application: The App allows me to apply the concepts of sequence, selection and iteration
C17	Metacognitive monitoring: The App interface is designed in a way that encourages me to improve my solution
C18	Multimedia: The interface is visually rich in multimedia and includes sound, color, graphics and animation
C19	Simulation programming: The App uses simple statements to mimic a high-level programming language in an active learning environment
C20	Creativity: The App allows me to create useful and original applications like games and movies
C21	Creativity: The App has various features that enable me to use my creative skills
C22	Creativity: The App supports the simulation of many creative ideas when solving the problem
C23	Analysis: When interpreting the problem statement, the App interface gives me clues on how to solve the problem
C24	Complexity of problem: The App allows me the flexibility to elaborate or build on my idea
C25	Disposition: The App forces me to have an enquiring mind

TABLE IV. LINGUISTIC AND FUZZY TRIPLE FOR THE CRITICAL THINKING ATTRIBUTES (CRITERIA)

	CRITERIA																								
Linguistic Values	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25
D1	I	VI	MI	VI	MI	MI	I	I	VI	VI	VI	I	MI	VI	VI	VI	MI	SI	SI	MI	I	I	VI	I	I
D2	I	VI	I	MI	I	SI	I	MI	VI	VI	VI	I	MI	I	I	VI	I	MI	MI	I	MI	MI	MI	I	I
D3	VI	VI	I	VI	I	I	MI	I	I	VI	VI	MI	MI	I	I	I	MI	SI	SI	SI	SI	SI	I	I	I
D4	I	I	I	I	I	I	MI	VI	VI	VI	VI	I	MI	VI	VI	VI	MI	MI	MI	MI	MI	MI	I	I	VI
D5	VI	VI	I	MI	VI	VI	I	VI	VI	VI	I	I	MI	I	I	VI	I	MI	I	VI	I	VI	VI	VI	VI
D6	VI	VI	VI	MI	MI	MI	I	VI	VI	VI	VI	I	I	VI	MI	VI	I	I	I	MI	MI	MI	I	I	VI
D7	SI	VI	I	I	MI	SI	SI	I	I	VI	SI	SI	SI	VI	I	I	I	SI	SI	MI	SI	SI	MI	VI	VI
D8	I	I	I	VI	MI	MI	MI	VI	VI	VI	I	I	MI	I	VI	VI	I	I	I	I	MI	I	VI	VI	VI
D9	VI	MI	I	MI	I	VI	VI	VI	VI	VI	VI	I	I	VI	VI	VI	MI	MI	MI	VI	MI	I	VI	I	I
D10	VI	SI	VI	I	SI	SI	I	VI	VI	VI	VI	VI	VI	VI	VI	VI	I	SI	I	VI	SI	I	I	MI	I

Table IV shows the linguistic rating of the 25 criteria by 10 Academic Experts teaching programming at university level.

Their input values are represented by the symbols D1 to D10 in the table below.

The Best Non-fuzzy Performance value (BNP) for a criterion weighting j, can be calculated using the following equation [13]:

$$BNP_{wj} = [(Upper\ bound_{wj} - lower\ bound_{wj}) + (Middle\ bound_{wj} - lower\ bound_{wj})]/3 + lower\ bound_{wj} \quad (12)$$

The BNP values give an indication of the relative importance of the criteria. Table V shows the aggregated score for each criterion.

Table V also shows the calculated BNP value for each criteria using Equation 12. Linguistic Weights were assigned based on the BNP values according to descriptions given in Table II.

Table VI shows the ranking of the criteria based on the BNP values indicated in Table V.

Table VI shows that C10 (Logic and reasoning: The App quickly alerts me when the sequencing of steps in my solution is logically incorrect) was the most highly ranked criteria by the expert decision-makers. The second most highly recommended criterion is C9 (Logic: The App helped me to improve my logic skills), followed by C16 (Application: The App allows me to apply the concepts of sequence, selection and iteration) and C14 (Alternate solutions: The App helps me to think about solving the problem in different ways). The least important criterion to decision-makers is criterion C18 (Multimedia: The interface is visually rich in multimedia and includes sound, color, graphics and animation).

TABLE V. AGGREGATED SCORES AND BNP VALUES FOR CRITICAL THINKING ATTRIBUTES (CRITERIA)

Criteria	**Aggregated Fuzzy Score**	**BNP Value**	**Linguistic Weight**
C1	1, 7.4, 9	5,8	I
C2	1, 7.2, 9	5,733333	I
C3	1, 7, 9	5,666667	I
C4	1, 6, 9	5,333333	MI
C5	1, 5, 9	5	MI
C6	1, 4.4, 9	4,8	SI
C7	1, 5.4, 9	5,133333	MI
C8	1, 7.8, 9	5,933333	I
C9	5, 8.6, 9	7,533333	VI
C10	7, 9, 9	8,333333	VI
C11	1, 7.8, 9	5,933333	I
C12	1, 6.2, 9	5,4	MI
C13	1, 4.2, 9	4,733333	SI
C14	5, 8.2, 9	7,4	VI
C15	1, 7.6, 9	5,866667	I
C16	5, 8.6, 9	7,533333	VI
C17	1, 5.4, 9	5,133333	MI
C18	1, 3, 9	4,333333	SI
C19	1, 4, 9	4,666667	SI
C20	1, 5.4, 9	5,133333	MI
C21	1, 3.2, 9	4,4	SI
C22	1, 4.8, 9	4,933333	SI
C23	1, 7, 9	5,666667	I
C24	1, 7.2, 9	5,733333	I
C25	5, 8, 9	7,333333	VI

TABLE VI. RANKED CRITERIA BASED ON BNP VALUES

Ranked Criteria					
1.	C10	11.	C24	21.	C6
2.	C9	12.	C3	22.	C13
3.	C16	13.	C23	23.	C19
4.	C14	14.	C12	24.	C21
5.	C25	15.	C4	25.	C18
6.	C8	16.	C7		
7.	C11	17.	C17		
8.	C15	18.	C20		
9.	C1	19.	C5		
10.	C2	20.	C22		

Table VII shows the fuzzy rating scale for weighing critical thinking criteria. Linguistic terms such as Very Poor (VP), Poor (P), Satisfactory (S), Good (G) and Very Good (VG) are expressed as a fuzzy triple with integers in the range 1 to 9. The Not Sure (N) option expressed as (0, 0, 0) is included in the application to cater for students who are not sure what rating to assign to criteria.

Table VIII shows the numeric labels for Learning Apps (Alternatives).

Table IX presents a snapshot of the assigned ratings by 175 student decision-makers to assess Scratch using the 25 critical thinking criteria in linguistic terms as shown in Table VII.

TABLE VII. LINGUISTIC AND FUZZY TRIPLE FOR THE CRITICAL THINKING ALTERNATIVES (LEARNING APPS)

Linguistic term	**Membership function**
Not Sure (N)	(0,0,0)
Very Poor (VP)	(1,1,3)
Poor (P)	(1,3,5)
Satisfactory (S)	(3,5,7)
Good (G)	(5,7,9)
Very Good (VG)	(7,9,9)

TABLE VIII. NUMERIC LABELS FOR THE PROGRAMMING LEARNING APPS (ALTERNATIVES)

Programming Learning Apps	**Alternatives Label**
Alice	1
Scratch	2
Blockly	3
MIT App Inventor	4

TABLE IX. ASSIGNED RATING BY DECISION-MAKERS FOR SCRATCH

Criteria	C1	C2	C3	C4	C5	C6	C7	…	C25
DM1	G	G	VG	G	VG	VG	G	…	VG
DM2	P	S	S	G	G	S	G	…	G
DM3	G	VG	G	VG	G	VG	VG	…	VG
DM4	VG	VG	VG	G	VG	VG	VG	…	VG
DM5	VG	VG	VG	VG	G	VG	VG	…	VG
DM6	VG	G	VG	VG	VG	VG	VG	…	VG
DM7	G	G	G	VG	G	S	G	…	S
DM8	G	P	S	VG	G	G	S	…	VG
DM9	S	G	S	S	G	S	S	…	VG
DM10	P	P	VP	G	VP	VP	VP	…	VG
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
DM173	G	G	S	S	G	VG	VG	…	VG
DM174	VG	S	VG	VG	S	VG	VG	…	G
DM175	VG	S	G	G	G	G	VG	…	G

TABLE X. NORMALIZED FUZZY DECISION MATRIX

	Alice (A1)	Scratch (A2)	Blockly (A3)	MIT Inventor (A4)
C1	[0,0.657142857142857,1]	[0,0.761269841269841,1]	[0,0.639365079365079,1]	[0,0.671111111111111,1]
C2	[0,0.656507936507937,1]	[0,0.711111111111111,1]	[0,0.627936507936508,1]	[0,0.660952380952381,1]
C3	[0,0.666031746031746,1]	[0,0.700952380952381,1]	[0,0.615238095238095,1]	[0,0.657777777777778,1]
C4	[0,0.693333333333333,1]	[0,0.766984126984127,1]	[0,0.622222222222222,1]	[0,0.719365079365079,1]
C5	[0,0.639365079365079,1]	[0,0.687619047619048,1]	[0,0.596190476190476,1]	[0,0.661587301587302,1]
C6	[0,0.690158730158730,1]	[0,0.743492063492063,1]	[0,0.633650793650794,1]	[0,0.663492063492064,1]
C7	[0,0.651428571428572,1]	[0,0.685714285714286,1]	[0,0.594285714285714,1]	[0,0.619682539682540,1]
C8	[0,0.760000000000000,1]	[0,0.829206349206349,1]	[0,0.710476190476191,1]	[0,0.720634920634921,1]
C9	[0,0.739682539682540,1]	[0,0.828571428571429,1]	[0,0.716190476190476,1]	[0,0.721269841269841,1]
C10	[0,0.655238095238095,1]	[0,0.714920634920635,1]	[0,0.640634920634921,1]	[0,0.686984126984127,1]
C11	[0,0.704126984126984,1]	[0,0.789206349206349,1]	[0,0.674920634920635,1]	[0,0.725079365079365,1]
C12	[0,0.740952380952381,1]	[0,0.795555555555556,1]	[0,0.723809523809524,1]	[0,0.738412698412698,1]
C13	[0,0.664126984126984,1]	[0,0.721269841269841,1]	[0,0.645714285714286,1]	[0,0.656507936507937,1]
C14	[0,0.749841269841270,1]	[0,0.802539682539682,1]	[0,0.688888888888889,1]	[0,0.726984126984127,1]
C15	[0,0.704761904761905,1]	[0,0.751111111111111,1]	[0,0.636825396825397,1]	[0,0.700952380952381,1]
C16	[0,0.735873015873016,1]	[0,0.779047619047619,1]	[0,0.681904761904762,1]	[0,0.706031746031746,1]
C17	[0,0.730793650793651,1]	[0,0.800000000000000,1]	[0,0.693333333333333,1]	[0,0.716825396825397,1]
C18	[0,0.741587301587302,1]	[0,0.806984126984127,1]	[0,0.666666666666667,1]	[0,0.676825396825397,1]
C19	[0,0.733968253968254,1]	[0,0.780952380952381,1]	[0,0.678095238095238,1]	[0,0.686349206349206,1]
C20	[0,0.725714285714286,1]	[0,0.708571428571429,1]	[0,0.640000000000000,1]	[0,0.666031746031746,1]
C21	[0,0.771428571428572,1]	[0,0.798095238095238,1]	[0,0.687619047619048,1]	[0,0.729523809523810,1]
C22	[0,0.756825396825397,1]	[0,0.773333333333333,1]	[0,0.689523809523810,1]	[0,0.714920634920635,1]
C23	[0,0.641269841269841,1]	[0,0.673650793650794,1]	[0,0.634285714285714,1]	[0,0.620952380952381,1]
C24	[0,0.738412698412698,1]	[0,0.775873015873016,1]	[0,0.674285714285714,1]	[0,0.708571428571429,1]
C25	[0,0.727619047619048,1]	[0,0.791111111111111,1]	[0,0.726349206349206,1]	[0,0.749841269841270,1]

Table X presents the results of the Normalised Fuzzy Decision Matrix from the MatLab R2020a application. This matrix was generated using Equation 3.

Table XI shows the results for FNIS and FPIS generated from the MatLab R2020a Application using Equations 8 and 7 respectively.

Table XII gives the results of the Closeness Coefficient values generated by the MatLab application. These values are generated by the MatLab App using Equation 11.

The CCi value is a ratio scale rating showing the ranking of learning App by 175 decision- makers from highest to lowest,

CCi (Scratch) > CCi (Alice)> CCi (MIT Inventor) > CCi (Blockly).

The results show that Scratch is the most preferred application for learning programming as it had the highest CCi value. This is congruent with various studies that compared visual programming environments using different methods under different contexts and found Scratch to be the most preferred learning tool [4, 11, 45]. The App with the lowest CCi value is Blockly, which therefore is the least preferred application.

TABLE XI. NORMALIZED FUZZY DECISION MATRIX

FNIS (A-)	**FPIS (A+)**
[0,4.47555555555556,9]	[0,5.32888888888889,9]
[0,4.39555555555556,9]	[0,4.97777777777778,9]
[0,4.30666666666667,9]	[0,4.90666666666667,9]
[0,3.11111111111111,7]	[0,3.83492063492064,7]
[0,2.98095238095238,7]	[0,3.43809523809524,7]
[0,1.90095238095238,5]	[0,2.23047619047619,5]
[0,2.97142857142857,7]	[0,3.42857142857143,7]
[0,4.97333333333333,9]	[0,5.80444444444444,9]
[0,6.44571428571429,9]	[0,7.45714285714286,9]
[0,5.76571428571429,9]	[0,6.43428571428571,9]
[0,4.72444444444444,9]	[0,5.52444444444445,9]
[0,3.61904761904762,7]	[0,3.97777777777778,7]
[0,1.93714285714286,5]	[0,2.16380952380952,5]
[0,6.20000000000000,9]	[0,7.22285714285714,9]
[0,4.45777777777778,9]	[0,5.25777777777778,9]
[0,6.13714285714286,9]	[0,7.01142857142857,9]
[0,3.46666666666667,7]	[0,4,7]
[0,2,5]	[0,2.42095238095238,5]
[0,2.03428571428571,5]	[0,2.34285714285714,5]
[0,3.20000000000000,7]	[0,3.62857142857143,7]
[0,2.06285714285714,5]	[0,2.39428571428571,5]
[0,2.06857142857143,5]	[0,2.32000000000000,5]
[0,4.34666666666667,9]	[0,4.71555555555556,9]
[0,4.72000000000000,9]	[0,5.43111111111111,9]
[0,6.53714285714286,9]	[0,7.12000000000000,9]

TABLE XII. CLOSENESS COEFFICIENT

Rank Order	**Programming Learning App**	**CCi**
2	Scratch	0.9941
1	Alice	0.4345
4	MIT inventor	0.3258
3	Blockly	0.0064

V. CONCLUSION

The study used a population of 500 and a sample population of 217 decision-makers to rate four learning Apps against each of 25 critical thinking criteria. Applying the Fuzzy TOPSIS MCDM method to the decision problem would have made handling the matrices both inconvenient and cumbersome. The researcher created code using MatLab R2020a to automate the process. The implementation of the coding for the Fuzzy TOPSIS MCDM method resulted in the ranking of four learning Apps, from the App that promoted critical thinking the best to the one that promoted it least among 1st year programming learners; these Apps were Scratch, Alice, MIT App Inventor and Blockly. The study is significant for lecturers teaching introductory programming modules to novice learners across educational institutions, especially during the global Covid-19 pandemic, when online learning has become commonplace and its teaching tools have become a requirement. Although the research suggests the positive impact of the visual programming environment on one's critical thinking skills, further research is required to investigate its subsequent impact in a text-based programming environment.

REFERENCES

[1] Tsai, C.-Y. 2019. Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*, 95: 224-232.

[2] Xu, Z., Ritzhaupt, A. D., Tian, F. and Umapathy, K. 2019. Block-based versus text-based programming environments on novice student learning outcomes: A meta-analysis study. Computer Science Education, 29 (2-3): 177-204.

[3] Medeiros, R. P., Ramalho, G. L. and Falcão, T. P. 2018. A systematic literature review on teaching and learning introductory programming in higher education. IEEE Transactions on Education, 62 (2): 77-90.

[4] João, Nuno, Fábio and Ana. 2019. A cross-analysis of block-based and visual programming apps with computer science student-teachers. Education Sciences, 9 (3): 181.

[5] Pinto-Llorente, A. M., Casillas-Martín, S., Cabezas-González, M. and García-Peñalvo, F. J. 2018. Building, coding and programming 3D models via a visual programming environment. Quality & Quantity, 52 (6): 2455-2468.

[6] Noone, M. and Mooney, A. 2018. Visual and textual programming languages: a systematic review of the literature. Journal of Computers in Education, 5 (2): 149-174.

[7] Kaya, K. Y. and Yildiz, İ. 2019. Comparing three free to use visual programming environments for novice programmers. Kastamonu Eğitim Dergisi, 27 (6): 2701-2712.

[8] Siburian, J., Corebima, A. D. and Saptasari, M. 2019. The correlation between critical and creative thinking skills on cognitive learning results. Eurasian Journal of Educational Research, 19 (81): 99-114.

[9] Papadakis, S. and Orfanakis, V. 2018. Comparing novice programing environments for use in secondary education: App Inventor for Android vs. Alice. International Journal of Technology Enhanced Learning, 10 (1-2): 44-72.

[10] Harrison, T. R. and Lee, H. S. 2018. iPads in the mathematics classroom: Developing criteria for selecting appropriate learning apps. International Journal of Education in Mathematics, Science and Technology, 6 (2): 155-172.

[11] Ropii, N., Hardyanto, W. and Ellianawati, E. 2019. Guided inquiry scratch increase students' critical thinking skills on the linear motion concept: Can it be? JPPPF: Jurnal Penelitian & Pengembangan Pendidikan Fisika, 5 (1): 63-68.

[12] Jahnke, I. and Liebscher, J. 2020. Three types of integrated course designs for using mobile technologies to support creativity in higher education. Computers & Education, 146: 103782.

[13] Ivanović, M., Xinogalos, S., Pitner, T., and Savić, M. 2017. Technology enhanced learning in programming courses – international perspective. Education and Information Technologies, 22 (6): 2981-3003.

[14] Nursal, A.T., M.F. Omar, and M.N.M. Nawi, The application of Fuzzy TOPSIS to the selection of building information modeling software. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2018. 10 (1-10): p. 1-5.

[15] Salih, M.M., et al., Survey on fuzzy TOPSIS state-of-the-art between 2007 and 2017. Computers & Operations Research, 2019. 104: p. 207-227.

[16] Yavuz, M. 2016. Equipment selection by using fuzzy TOPSIS method. In: Proceedings of IOP Conference Series: Earth and Environmental Science, 44 (4).

[17] Başaran, S. and Haruna, Y. 2017. Integrating FAHP and TOPSIS to evaluate mobile learning applications for mathematics. Procedia Computer Science, 120: 91-98.

[18] Rajak, M. and Shaw, K. 2019. Evaluation and selection of mobile health (mHealth) applications using AHP and fuzzy TOPSIS. Technology in Society, 59: 101186.

[19] Balioti, V., Tzimopoulos, C. and Evangelides, C. 2018. Multi-criteria decision making using TOPSIS method under fuzzy environment. Application in spillway selection. In: Proceedings of Multidisciplinary Digital Publishing Institute Proceedings. 637.

[20] Ece, O. and A.S. Uludag, Applicability of fuzzy TOPSIS method in optimal portfolio selection and an application in BIST. International Journal of Economics and Finance, 2017. 9 (10): p. 107-127.

[21] Memari, A., et al., Sustainable supplier selection: A multi-criteria intuitionistic fuzzy TOPSIS method. Journal of Manufacturing Systems, 2019. 50: p. 9-24.

[22] Mohammed, H.J., M.M. Kasim, and I.N. Shaharanee, Evaluation of E-learning approaches using AHP-TOPSIS technique. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2018. 10 (1-10): p. 7-10.

[23] Ranganath, N., Sarkar, D., Patel, P. and Patel, S. 2020. Application of fuzzy TOPSIS method for risk evaluation in development and implementation of solar park in India. International Journal of Construction Management: 1-11.

[24] Han, H. and S. Trimi, A fuzzy TOPSIS method for performance evaluation of reverse logistics in social commerce platforms. Expert Systems with Applications, 2018. 103: p. 133-145.

[25] Kules, B. 2016. Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. In: Proceedings of the Association for Information Science and Technology, 53 (1): 1-6.

[26] Danczak, S. M., Thompson, C. D. and Overton, T. L. 2020. Development and validation of an instrument to measure undergraduate chemistry students' critical thinking skills. Chemistry Education Research and Practice, 21 (1): 62-78.

[27] Romero, M., Lepage, A. and Lille, B. 2017. Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education, 14 (1): 1-15.

[28] Turker, P. M. and Pala, F. K. 2019. A study on students' computational thinking skills and self-efficacy of block based programming. i-Manager's Journal on School Educational Technology, 15 (3): 18.

[29] Weintrop, D. and Wilensky, U. 2017. Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education (TOCE), 18 (1): 1-25.

[30] Šiaulys, T. 2020. Modeling the system for interactive tasks development: engagement taxonomy for introductory programming tools. Switzerland: Springer.

[31] Atmatzidou, S., Atmatzidou, S., Demetriadis, S., Demetriadis, S., Nika, P. and Nika, P. 2018. How does the degree of guidance support students' metacognitive and problem solving skills in educational robotics? Journal of Science Aducation and Technology, 27 (1): 70-85.

[32] Figueiredo, J. and García-Peñalvo, F. J. 2019. Teaching and learning strategies of programming for university courses. In: Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality. 1020-1027.

[33] Papadakis, S. and Kalogiannakis, M. 2017. Using gamification for supporting an introductory programming course: The case of classcraft in a secondary education classroom. In: Interactivity, game creation, design, learning, and innovation. Springer, 366-375.

[34] Tuloli, M., Latief, M. and Rohandi, M. 2019. Scratching our own itch: Software to teach software programming. In: Proceedings of International Conference on Education, Science and Technology. Redwhite Press, 115-121.

[35] Radianti, J., Majchrzak, T. A., Fromm, J. and Wohlgenannt, I. 2020. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. Computers & Education, 147: 103778.

[36] Palczewski, K. and W. Sałabun, The fuzzy TOPSIS applications in the last decade. Procedia Computer Science, 2019. 159: p. 2294-2303.

[37] Samaie, F., Meyar-Naimi, H., Javadi, S. and Feshki-Farahani, H. 2020. Comparison of sustainability models in development of electric vehicles in Tehran using fuzzy TOPSIS method. Sustainable Cities and Society, 53: 101912.

[38] Sahin, B., Yip, T. L., Tseng, P.-H., Kabak, M. and Soylu, A. 2020. An application of a fuzzy TOPSIS multi-criteria decision analysis algorithm for dry bulk carrier selection. Information, 11 (5): 251.

[39] Junior, F.R.L., L. Osiro, and L.C.R. Carpinetti, A comparison between Fuzzy AHP and Fuzzy TOPSIS methods to supplier selection. Applied Soft Computing, 2014. 21: p. 194-209.

[40] Rashidi, K. and Cullinane, K. 2019. A comparison of fuzzy DEA and fuzzy TOPSIS in sustainable supplier selection: Implications for sourcing strategy. Expert Systems with Applications, 121: 266-281.

[41] Chen, C.-T., Extensions of the TOPSIS for group decision-making under fuzzy environment. Fuzzy Sets and Systems, 2000. 114(1): p. 1-9.

[42] Kahraman, C., Ateş, N. Y., Çevik, S., Gülbay, M. and Erdoğan, S. A. 2007. Hierarchical fuzzy TOPSIS model for selection among logistics information technologies. Journal of Enterprise Information Management, 20 (2): 143-168.

[43] Zimmermann, H.J., Fuzzy set theory. Wiley Interdisciplinary Reviews: Computational Statistics, 2010. 2 (3): p. 317-332.

[44] de Barros, L. C., Bassanezi, R. C. and Lodwick, W. A. 2017. The extension principle of Zadeh and fuzzy numbers. In: A first course in fuzzy logic, fuzzy dynamical systems, and biomathematics. Springer, 23-41.

[45] Erol, O. and Kurt, A. A. 2017. The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. Computers in Human Behavior, 77: 11-18.