

Article

Hybrid MCDM Framework for Selecting Visual Programming Software for Children with Special Educational Needs Using the ROC and PROMETHEE II Methods

Marija Krstić ^{1,*} , Dragan Soleša ²  and Lazar Krstić ¹ ¹ Department of Higher Business School Leskovac, Academy of Applied Studies Southern Serbia, Partizanska 7, 16000 Leskovac, Serbia; krstic.lazar@vpsle.edu.rs² Faculty of Economics and Engineering Management, University Business Academy in Novi Sad, Cvečarska 2, 21000 Novi Sad, Serbia; dragan.solesa@fimek.edu.rs

* Correspondence: krstic.marija@vpsle.edu.rs

Abstract

Visual programming using blocks and diagrams facilitates understanding of fundamental programming concepts, which is particularly important for children with special educational needs because it reduces their cognitive load and encourages interactive learning. This study aimed to develop and apply a hybrid multi-criteria framework to evaluate, rank, and select visual programming software solutions intended for children with special educational needs. Based on an analysis of the educational context and the target population's needs, a set of criteria was defined to evaluate and select the most suitable software solution. Data for the analysis were collected using a structured questionnaire, from which a decision matrix was developed. Within the proposed hybrid multi-criteria decision-making (MCDM) framework, criterion weights were determined using the Rank Order Centroid (ROC) method, and the ranking of alternatives was performed using the Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE II). Additionally, a sensitivity analysis was conducted to assess the stability and robustness of the obtained rankings in relation to changes in the criterion weights. The results indicate a stable ranking of alternatives and the identification of the most favorable solution in the majority of scenarios. The projection quality of 91.1% in the Geometrical Analysis for Interactive Aid (GAIA) plane confirmed the reliability of the visual interpretation of the results. The proposed framework improves the decision-making process and provides a foundation for further research in educational software evaluation.

Keywords: visual programming; educational software; children with special educational needs; hybrid MCDM framework; ROC method; PROMETHEE II; sensitivity analysis; evaluation of software solutions



Academic Editor: Jose María Alvarez Rodríguez

Received: 19 May 2026

Revised: 13 June 2026

Accepted: 22 June 2026

Published: 25 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In modern education, the importance of visual programming tools is increasingly emphasized because they help students master fundamental programming concepts more easily through intuitive graphical interfaces [1,2]. Research indicates that this approach fosters logical and algorithmic thinking through interactive, visual learning [1,3]. When working with children with special educational needs, selecting an appropriate software solution is further complicated by cognitive, motor, and perceptual limitations [4,5].

The research problem is the lack of a standardized, empirically validated, multicriteria framework for selecting visual programming software for children with special educational needs. Although many tools are available, comparing them is not straightforward because of numerous, often conflicting criteria that affect their applicability in the educational process. Consequently, the selection process in practice is often insufficiently structured and subject to subjective judgments, complicating the making of objective and transparent decisions. This situation indicates the need to develop a formalized hybrid MCDM framework to enable systematic evaluation, ranking, and selection of software solutions in this context.

MCDM is an approach to problem-solving under conditions involving multiple, often conflicting criteria [6,7], and has wide applications across various fields, including education [8–10]. MCDM methods enable the structured evaluation and comparison of alternatives against predefined criteria, thereby ensuring greater transparency and consistency in the decision-making process [6,8,9].

The evaluation criteria framework was developed based on an analysis of the educational context and the specific needs of the target population, as no single standardized set of criteria is available in the literature for this purpose. Empirical data were collected through a structured electronic questionnaire distributed to educational institutions specializing in educating children with special educational needs, enabling the construction of a matrix for multi-criteria analysis. The methodological approach included applying the ROC method to determine the criterion weights and the PROMETHEE II method to rank the alternatives. The stability of the results was further examined using a sensitivity analysis.

The main contribution of this study is the development of an integrated multi-criteria framework for evaluating and selecting visual programming software solutions for children with special needs, including a clearly defined set of criteria tailored to the target group's specific characteristics. This contribution is also reflected in the combined use of the ROC and PROMETHEE II methods in this context, as well as in a sensitivity analysis to assess the model's robustness. The proposed framework contributes to a more objective and transparent decision-making process and serves as a basis for further research in educational software evaluation.

Given the specific requirements of inclusive education, selecting an appropriate software solution is complicated by the interdependence and often conflicting relationships among relevant criteria, including ease of use, cognitive load, and pedagogical effectiveness. In this context, multi-criteria decision-making provides a systematic, methodologically grounded approach to decision-making problems with conflicting criteria, as confirmed by contemporary MCDM approaches [11,12].

2. Literature Review

The literature was searched across scientific databases, digital libraries, and open-access sources. The search covered papers published between 2020 and 2025 that relate to the application of multicriteria methods in education, the evaluation of educational software, and the use of visual programming with children with special educational needs. The search was organized around these three research directions, as no study integrating them into a single whole was identified. Papers were selected based on their thematic relevance to the research.

In education, multicriteria methods are widely used to evaluate teaching strategies and educational technologies. Troussas et al. developed a TOPSIS-based model to tailor teaching strategies to students' individual characteristics, thereby improving learning efficiency [13]. Similarly, Alshamsi et al. applied an MCDM approach to evaluate distance

learning systems, identifying key criteria such as content quality, interactivity, and user satisfaction [14].

Basciani et al. proposed a model for software quality assessment that integrates technical and user aspects, enabling a comprehensive evaluation of software solutions [15]. Shabrina et al. also analyzed systems for automatically generating feedback in programming learning, using a multicriteria approach that evaluates accuracy, efficiency, and user experience [16].

The application of multicriteria methods in inclusive education is a significant research area. Studies show that MCDM can be used to prioritize the needs of children with autism spectrum disorders and to evaluate different educational approaches [17]. Additionally, methods such as AHP and GRA have been used to analyze the environment and factors influencing the learning of children with autism [18]. Kumar et al. analyzed the application of MCDM methods for evaluating inclusive educational systems and identified key factors affecting the quality of education for children with special needs [19]. Cañete et al. used a multicriteria approach in product design, specifically for smart toys for children with autism, defining criteria that include sensory, functional, and economic aspects [20].

It is important to emphasize the role of digital technologies in supporting the development of children with special needs, particularly in communication, social interaction, and learning. Advances in digital technologies have further expanded the possibilities for supporting children with special needs. For example, modern technologies, such as machine learning and deep learning, enable the development of systems that can assist in identifying and addressing issues in children's learning and daily functioning [21,22]. These approaches indicate the significant potential for integrating intelligent systems and educational tools.

Visual programming is a significant approach to teaching programming to children, as it enables the acquisition of fundamental concepts in an intuitive and accessible way. Block-based tools are particularly noteworthy because they eliminate the need to learn syntax, thereby facilitating initial learning. However, their use with children with special needs requires additional adaptations to ensure adequate accessibility and effective learning. Research shows that numerous challenges arise in using these tools, including the complexity of the user interface, visual overload, and a lack of functionality tailored to users' specific needs [23]. To overcome these problems, guidelines have been proposed that include simplifying the interface, applying visual and auditory stimuli, and supporting individualized learning [23,24].

Despite the growing number of papers addressing multi-criteria analysis, inclusive education, and visual programming, the literature still lacks an integrated approach that unifies these research directions into a methodologically grounded framework. Existing research has mainly focused on individual aspects of the problem. At the same time, systematic evaluation and selection of software solutions that consider the specific needs of children with special educational needs remain insufficiently represented.

This shortcoming indicates the need for an approach that enables clearer, more objective comparisons of available solutions. Accordingly, this study aims to integrate existing knowledge from multi-criteria analysis, software evaluation, and inclusive education by proposing a hybrid framework for the systematic analysis and selection of software solutions for visual programming for children with special educational needs.

3. Materials and Methods

3.1. Research Design

This quantitative cross-sectional study included 66 respondents employed in schools serving children and students with developmental disabilities in the Republic of Serbia [25,26]. The

sample comprised computer science teachers and professional associates (pedagogues, psychologists, and special education teachers) who were key participants in selecting and implementing educational software solutions. In addition to respondents' socio-professional characteristics, the study captured characteristics of the educational environment in which they worked, including students' ages and the types of developmental and educational needs (e.g., learning difficulties, intellectual disabilities, developmental disorders, and multiple disabilities). These data provide a more precise understanding of the conditions under which visual programming software is applied in practice. Given the anonymous nature of the research, no identification data on individual schools were collected; only aggregated data on the type of institution (primary or secondary school) and the basic characteristics of the educational environment were gathered. The sample was formed using convenience sampling, constrained by limited access to specialized educational institutions and the specific nature of the target population.

3.2. Research Objectives and Research Questions

This study aimed to identify and quantify the criteria that teachers and professional associates use when selecting visual programming software for children with special educational needs, and to develop and apply a hybrid multi-criteria framework to support decision-making in evaluating and ranking these solutions. In line with this aim, the following research questions were formulated:

- RQ1: What are the most important criteria for evaluating and selecting visual programming software solutions for children with special educational needs?
- RQ2: How are software solutions ranked using the hybrid ROC–PROMETHEE II model according to the defined criteria?
- RQ3: How stable is the resulting ranking of software solutions under changes in criterion weights, as assessed by sensitivity analysis and GAIA visualization?

3.3. Data Collection and Instrumentation

The data were collected via an anonymous, structured questionnaire created in Google Forms and electronically distributed to 48 primary and secondary schools in the Republic of Serbia that serve children and students with developmental disabilities, yielding 66 valid responses from teachers and professional associates. The questionnaire included both closed- and open-ended questions to identify software solutions used for visual programming among children with special educational needs, to rank evaluation criteria, and to determine key factors influencing the choice of software solution. Criteria were ranked on an ordinal scale, with respondents assigning ranks from 1 to 11 (1—most important criterion, 11—least important), providing the basis for determining criterion weights using the ROC method in the next phase of the analysis. Participation was voluntary and anonymous, and no personal identification data were collected.

3.4. Criteria Definition

Based on a systematic review and analysis of relevant scientific and professional literature in the fields of educational technologies, inclusive education, and multicriteria analysis, a set of criteria was developed to evaluate software solutions for visual programming by children with special educational needs. The criteria were formulated to encompass the pedagogical, cognitive, technical, and inclusive aspects of software solutions and to enable quantification within the MCDM model.

Special emphasis is placed on the principles of inclusive education, including accessibility, equity in learning, and adapting educational content to the diverse needs of students [27,28]. Equity in learning means providing equal opportunities to achieve learn-

ing outcomes through tailored approaches, methods, and resources, in accordance with students' individual characteristics. Contemporary frameworks for evaluating digital educational tools that highlight the importance of usability, interactivity, and pedagogical value were also considered [29,30].

By synthesizing relevant sources, a structured set of criteria suitable for multi-criteria analysis was developed [31,32]. Based on this framework, the criteria shown in Table 1 were defined and used to evaluate software solutions or visual programming intended for children with special educational needs.

Table 1. Criteria for evaluating visual programming software for children with special educational needs.

No.	Criterion	Description	References
1	Visual accessibility	Level of interface clarity, contrast, and visual readability of elements	[27,28]
2	Cognitive complexity	Level of mental workload during system use	[27,31,33]
3	Pedagogical support	Availability of educational resources, tutorials, and system support during the learning process	[30,31],
4	Support for specific skill development	Development of computational thinking, algorithmic reasoning, and problem-solving skills	[33,34]
5	Interaction simplicity	Intuitiveness and ease of use of the visual environment	[33,35]
6	Multisensory learning	Use of combined visual and auditory learning channels	[31,34]
7	Adaptability and inclusiveness	Degree of adaptation to the individual educational needs of users	[27–29]
8	Voice/audio support	Availability of audio interaction, sound feedback, speech assistance, and multimedia audio capabilities	[31]
9	Technical accessibility	Stability, compatibility with different devices, and ease of technical implementation	[31,32]
10	Adaptability to the school environment	Integration into the teaching process and curriculum	[30]
11	Price of software solution	Cost sustainability and financial affordability of the software solution	[32,36]

A structured set of criteria was developed to evaluate visual programming software solutions intended for children with special educational needs. The criteria are systematized and presented in Table 1, ensuring clarity and providing a basis for further quantitative analysis within a multicriteria decision-making model.

3.5. Data Structure and Preprocessing

The data were organized into two basic structures to construct a hybrid MCDM framework using the ROC and PROMETHEE II methods. This approach enables the integration of respondents' subjective assessments with the objective characteristics of software solutions.

The first structure presents the criterion-ranking matrix based on respondents' responses. Using these rankings, the average criterion rankings were calculated and used as inputs for the ROC method [37]. The ROC method transforms ordinal rankings into

quantitative weights, with the weight of the i -th criterion determined as the average of the reciprocal ranks, according to the following expression:

$$w_i = \frac{1}{n} \sum_{j=i}^n \frac{1}{j} \quad (1)$$

where w_i is the weight of the i th criterion and n is the total number of criteria. The ROC method was chosen for its simplicity, stability, suitability for processing ordinal data, and the absence of additional subjective parameters in weight determination [38,39].

The questionnaire was designed to obtain ordinal rankings of predefined evaluation criteria for use within a multicriteria decision-making framework. The evaluation criteria were predefined based on a systematic review of relevant scientific and professional literature and were not intended to represent items of a single measurement scale. Instead, they were treated as independent decision criteria within the MCDM framework. Since the criteria represent independent evaluation dimensions rather than items of a psychometric scale, psychometric reliability and validity procedures, such as Cronbach's alpha and factor analysis, were not considered appropriate for the present study design. The ROC method was applied to transform respondents' ordinal rankings into quantitative criterion weights.

The second structure presents a decision matrix that includes the software alternatives Scratch, Tynker, MakeCode, Blockly, and Alice, selected based on empirical data from respondents and the relevant literature in educational technologies [40,41]. Although the present study considered five software alternatives identified through empirical findings, the proposed MCDM framework is not limited to these solutions and may be applied to a broader set of visual programming environments. The values for the alternatives were drawn from technical documentation and prior research and are expressed on a unified scale to ensure comparability across criteria.

The data were normalized using the min–max method to ensure comparability across criteria, given their differing measurement scales [42]. This method was chosen because it preserves relative relationships between values and is a standard step in MCDM data preprocessing [41]. The normalization further enables the consistent application of the PROMETHEE II method for ranking alternatives [43].

3.6. MCDM Framework

A hybrid MCDM framework combines the ROC method, which transforms ordinal ranks into numerical weights without additional subjective parameters, with the PROMETHEE II method, which provides a complete ranking of alternatives based on the net preference flow [7,43]. The ROC and PROMETHEE II methods were selected for their complementary roles in multicriteria decision-making. The ROC method offers a simple, stable transformation of criterion ranks into weights, whereas PROMETHEE II provides a complete ranking of alternatives based on preferential relations among criteria. This combination enables the consistent integration of respondents' subjective assessments with objective multicriteria analyses.

The criterion weights obtained using the ROC method served as input parameters for the PROMETHEE II analysis, ensuring consistent evaluation of alternatives [38,39]. The net preference flow for each alternative is defined as

$$\Phi(a) = \Phi^+(a) - \Phi^-(a) \quad (2)$$

where $\Phi^+(a)$ and $\Phi^-(a)$ are the positive and negative preference flows, respectively. The final ranking of alternatives is based on the net flow values, with higher values indicating a better position in the ranking and negative values indicating weaker relative performance [43].

A linear (V-shaped) preference function with a threshold of $p = 2$ was applied to all criteria because the input data were previously normalized and expressed on a uniform scale. This approach enabled differentiation among alternatives based on the magnitude of differences in criterion values while avoiding the need to define an additional indifference threshold (q). This approach contributes to the model's simplicity, preserves interpretability, and increases the method's sensitivity to variations among alternatives [44,45].

Although a common preference threshold ($p = 2$) was applied across all criteria to ensure methodological consistency and simplify model implementation, criterion-specific thresholds could provide a more detailed representation of preference structures. This may be particularly relevant when comparing objectively measurable criteria, such as the price of a software solution criterion, with more subjective criteria, such as pedagogical support. Future studies may investigate criterion-specific preference and indifference thresholds to further refine the decision-making model and assess their influence on ranking stability.

A sensitivity analysis was conducted by varying the criteria weights within $\pm 10\%$ and $+20\%$ of their nominal values to assess the stability of the ranking and identify the criteria with the greatest impact on the results. This approach validates the model's robustness and aligns with contemporary PROMETHEE and MCDM studies [39,43–45].

The ROC weights were calculated in Microsoft Excel for its convenience in processing ordinal data and implementing standard MCDM calculations. Visual PROMETHEE 1.4 Academic Edition software was used to apply the PROMETHEE II method and conduct a sensitivity analysis, enabling ranking of alternatives and assessing the stability of the results.

3.7. Limitations of the Study

A limitation of this research is that fewer than 10% of participants have experience with multiple visual programming software solutions. Consequently, respondents' direct experience could not provide a sufficiently reliable basis for the comparative evaluation of all analyzed alternatives. For this reason, the assessment of software alternatives was based on technical documentation, relevant scientific literature, and a comparative analysis of software functionalities. The survey was not used to evaluate software alternatives against the defined criteria; rather, it served to identify visual programming environments used in educational practice and to determine the relative importance of the evaluation criteria. Therefore, the final ranking results should be interpreted in light of this triangulated evaluation framework.

Another limitation relates to the heterogeneity of the target population. Children with special educational needs were considered as a broad target group, without distinguishing among specific categories of disabilities, such as cognitive, motor, sensory, or perceptual impairments. Although the selected criteria reflect general principles of inclusive education and accessibility, future research may examine the suitability of visual programming environments for specific groups of learners and investigate whether different disability categories require different evaluation criteria or weighting structures.

A further limitation concerns the study sample. The empirical research was conducted among teachers and professional associates working in schools that educate children with special educational needs in the Republic of Serbia. Participants were recruited using a convenience sampling approach, which may have influenced the composition of the sample. Consequently, the obtained criterion weights and ranking results should be interpreted in terms of the characteristics of the surveyed participants and should not be automatically generalized to other educational systems. Future research may validate and extend the proposed framework using larger and more geographically diverse samples, enabling comparisons across different countries and educational systems.

4. Results

The relative importance of the criteria was determined using the ROC method based on the ranking of 11 criteria by 66 respondents. The criteria were ranked on a scale from 1 (most important) to 11 (least important), after which the mean ranks were calculated, and an aggregate ranking of the criteria was established. Based on these values, the weights of the criteria were calculated and used in further analyses. Table 2 presents the calculated criteria weights obtained using the ROC method.

Table 2. Criteria weights obtained using the ROC method.

Criterion	Average Rank	Rank	ROC Weight
Visual accessibility	2.9545	1	0.2745
Cognitive complexity	3.9545	2	0.1836
Pedagogical support	5.4091	3	0.1382
Interaction simplicity	5.4545	4	0.1079
Support for specific skill development	5.6364	5	0.0851
Adaptability and inclusiveness	5.8182	6	0.0670
Multisensory learning	6.0909	7	0.0518
Technical accessibility	6.6818	8	0.0388
Price of software solution	7.6818	9	0.0275
Voice/audio support	8.0909	10	0.0174
Adaptability to the school environment	8.2273	11	0.0083

The results in Table 2 show that the criteria for visual accessibility (0.2745) and cognitive complexity (0.1836) have the highest weights. These are followed by pedagogical support, interaction simplicity, support for specific skill development, adaptability and inclusiveness, multisensory learning, and technical accessibility. The lowest weights were assigned to software solution cost, voice/audio support, and adaptability to the school environment. This weight distribution underscores the dominant importance of accessibility, usability, and support for the learning process in evaluating software solutions.

The criterion weights and the decision matrix, together with the defined preference functions, constitute the fundamental inputs of the multi-criteria model and enable quantitative comparison of alternatives within the MCDM approach [46]. The PROMETHEE II method is based on the outranking principle, in which alternatives are compared in pairs for each criterion. The results are aggregated into positive and negative flows, and the difference between them defines the net flow, which is used to rank the alternatives [47].

Table 3 presents a qualitative evaluation of the software alternatives against the defined criteria, which serves as the basis for constructing the decision matrix and precedes its numerical transformation. Qualitative assessments were standardized using a five-level ordinal scale to ensure consistency and comparability across criteria. The levels used (very high, high, medium, low, and very low) were based on an analysis of technical documentation, available scientific literature, and a comparative analysis of software solution functionalities.

Table 3. Qualitative comparative evaluation of visual programming software alternatives based on defined criteria.

Criterion	Scratch	MakeCode	Blockly	Tynker	Alice	References
Visual accessibility	very high	high	high	high	medium	[48–53]
Cognitive complexity	very low	medium	medium	medium	high	[54–56]
Pedagogical support	very high	high	medium	very high	high	[48–52,57]

Table 3. *Cont.*

Criterion	Scratch	MakeCode	Blockly	Tynker	Alice	References
Interaction simplicity	very high	high	high	high	low	[49,50,53,55]
Support for specific skill development	very high	high	medium	high	high	[54,56,57]
Adaptability and inclusiveness	very high	high	medium	high	medium	[51,57,58]
Multisensory learning	very high	high	medium	very high	high	[52,59]
Technical accessibility	very high	very high	medium	high	medium	[48–52]
Price of software solution	very low	very low	very low	low	medium	[48–52]
Voice/audio support	high	medium	low	high	medium	[45,48–52,57]
Adaptability to the school environment	very high	very high	medium	high	medium	[44,51]

The qualitative assessments were converted to a five-point ordinal Likert scale (1–5) [60], using the following mapping: very high = 5, high = 4, medium = 3, low = 2, and very low = 1. This approach enables the quantification of qualitative data and their use in multi-criteria analysis. In this scheme, a higher value (5) indicates the most favorable performance level, whereas lower values indicate a weaker degree of fulfillment of the observed criterion. For the criteria of cognitive complexity and software solution cost, an inverse preference was applied, so that lower values represented more favorable alternatives, thereby ensuring consistency in preference direction within the PROMETHEE II method [43,47].

Qualitative assessments were assigned through a comparative analysis of the software solutions' functionalities, drawing on relevant technical documentation and scientific literature. Using these standardized and transformed data, a decision matrix was constructed, as shown in Table 4, which serves as the input dataset for the multicriteria ranking of alternatives.

Table 4. Quantitative comparative evaluation of visual programming software alternatives based on defined criteria.

Criterion	Scratch	MakeCode	Blockly	Tynker	Alice
Visual accessibility	5	4	4	4	3
Cognitive complexity (inverse)	5	3	3	3	2
Pedagogical support	5	4	3	5	4
Interaction simplicity	5	4	4	4	2
Support for specific skill development	5	4	3	4	4
Adaptability and inclusiveness	5	4	3	4	3
Multisensory learning	5	4	3	5	4
Technical accessibility	5	5	3	4	3
Price of software solution (inverse)	5	5	5	4	3
Voice/audio support	4	3	2	4	3
Adaptability to the school environment	5	5	3	4	3

Scratch consistently achieves the highest scores across nearly all evaluated criteria, indicating its overall superiority among the software solutions analyzed. MakeCode ranks second, demonstrating strong performance in technical accessibility, software solution cost, and adaptability to the school environment, while maintaining stable results across other criteria. Tynker ranks third, showing consistently high values in pedagogical support and multisensory learning, although slightly lower performance on the software solution cost criterion influences its overall position. Blockly ranks fourth and exhibits moderate perfor-

mance across most criteria, with its strongest performance observed in the software solution cost criterion. Alice ranks last, with comparatively lower values for visual accessibility, cognitive complexity, and interaction simplicity than the other evaluated alternatives.

The resulting decision matrix served as input for the PROMETHEE II analysis. Within the model, a linear (V-shaped) preference function with a threshold $p = 2$ was applied, defined in accordance with the five-point Likert rating scale, thereby allowing differentiation between moderate and more pronounced differences among alternatives and enabling more precise discrimination with respect to the criteria considered [60–62].

The PROMETHEE II method enables calculation of positive (φ^+), negative (φ^-), and net preference flows (φ). Positive flow (φ^+) reflects how much an alternative surpasses others, whereas negative flow (φ^-) indicates how much others outweigh a given alternative. The net flow (φ) is the difference between the positive and negative flows and is used to rank alternatives [61–63]. The results are presented in Table 5.

Table 5. PROMETHEE II ranking and preference flows of software alternatives.

Rank	Alternative	φ^+	φ^-	φ
1	Scratch	0.4755	0.1939	0.2817
2	Tynker	0.2603	0.1050	0.1554
3	MakeCode	0.1899	0.1519	0.0380
4	Alice	0.1753	0.4049	−0.2295
5	Blockly	0.1072	0.3527	−0.2456

The values in Table 5 show that the Scratch alternative achieved the highest net preference flow ($\varphi = 0.2817$), ranking first. Tynker ranked second with a positive net flow ($\varphi = 0.1554$), indicating stable and competitive performance across the observed criteria. MakeCode achieved a lower but still positive value ($\varphi = 0.0380$), indicating a relatively balanced ratio of advantages to disadvantages. Alice and Blockly had negative net preference flow values, indicating that other alternatives on more criteria outperformed them. The lowest value was recorded by Blockly ($\varphi = -0.2456$), which ranked last.

Although the initial analysis of the decision matrix (Table 4) shows a slight advantage for the MakeCode alternative over the Tynker alternative, the application of the PROMETHEE II method yields a different ranking, with the Tynker alternative ranked second. This difference results from the influence of criterion weights and the model's preference structure, thereby confirming the importance of applying multi-criteria methods in decision-making and the need for a more detailed analysis of the relationships between alternatives. When interpreting the ranking results, it should be taken into account that respondents differed in their familiarity with the analyzed software solutions. However, the ranking was derived from a triangulated evaluation process based on technical documentation, scientific literature, and software functionality analysis rather than respondents' assessments of the software alternatives. For a clearer interpretation of the results, a graphical representation of the ranking of alternatives based on net preference flow values is provided (Figure 1).

The ranking graph (Figure 1) shows the distribution of alternatives according to the net preference flow values. Scratch held the highest position with a pronounced positive value, confirming its overall most favorable status compared to the other alternatives. Tynker also achieved a significant positive value and ranked second, whereas MakeCode, with a value close to zero, indicated an almost balanced relationship between advantages and disadvantages. In contrast, Blockly and Alice are located in the negative part of the scale, indicating that, in aggregate terms, they are outperformed by other alternatives based on the observed criteria. Blockly has the lowest value and thus occupies the last place in the

ranking. For additional insight into the relationships between alternatives and criteria, the results of the PROMETHEE II method are also presented on the GAIA plane, which enables visualization of multidimensional preferential relations [61,62] (Figure 2).

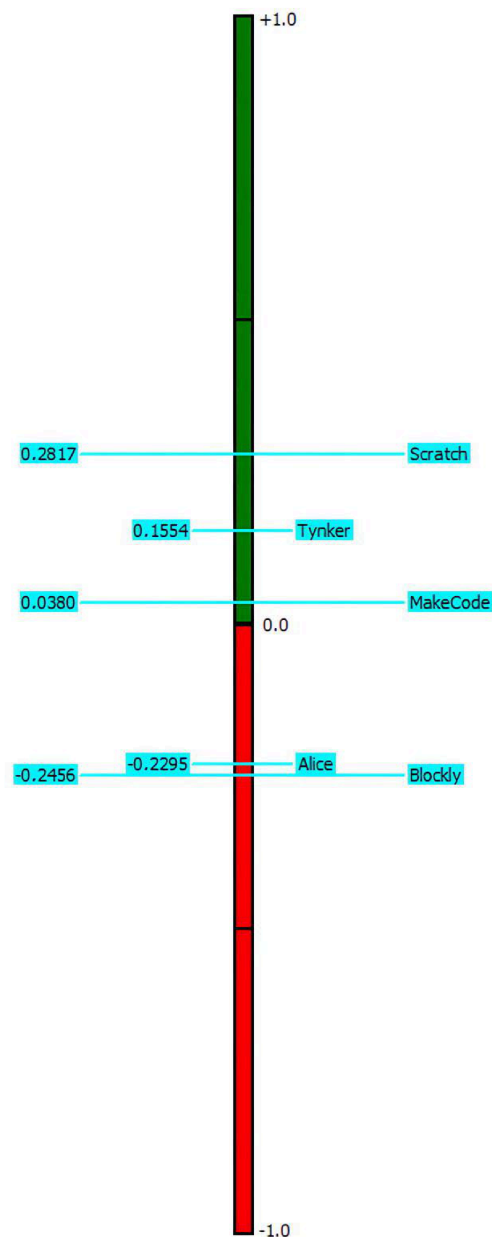


Figure 1. PROMETHEE II ranking of software alternatives based on net preference flow (ϕ).

The GAIA plane enables visual analysis of relationships between alternatives and criteria by projecting the multidimensional decision space into a two-dimensional display [62,63]. The GAIA plane shows that most criteria are aligned in a similar direction, indicating strong mutual alignment and low conflict within the model. Due to the high similarity and close alignment of several criterion vectors, some criteria overlap in the GAIA projection and are therefore not individually distinguishable in the graphical representation. Such overlap is expected when criteria exhibit a high degree of consistency and does not affect the interpretation of the results.

The criteria cognitive complexity and cost of the software solution stand out with a different vector orientation, indicating their conflicting relationship with criteria related to the inclusive, pedagogical, and functional characteristics of the software solutions. In

relation to the decision axis (π -axis), the alternative Scratch is closest to the dominant direction of the criteria, confirming its highest alignment with the defined preferences and corresponding to the results of the PROMETHEE II ranking. Tynker and MakeCode occupy intermediate positions, indicating partial alignment with the dominant decision direction. At the same time, Alice and Blockly are farther from the π -axis, indicating a lower level of overall alignment with the defined evaluation criteria. The spatial distribution of alternatives confirms differences in the performance of the analyzed software solutions and enables a clearer interpretation of their interrelationships within the defined decision model. The projection quality of 91.1% indicates very good representativeness of the GAIA plane and confirms the reliability of the visual interpretation of the results [62,63].



Figure 2. GAIA plane of software alternatives and evaluation criteria.

To assess the stability of the ranking and determine the influence of individual criteria on the results, a sensitivity analysis was conducted by varying criterion weights by $\pm 10\%$ and $+20\%$. The analysis was performed using the variable weights method (Walking Weights) in the Visual PROMETHEE tool, with particular emphasis on the key

criteria [44,45,47]. The results of the sensitivity analysis are presented in Table 6, and Table 7 provides a comparative overview of the baseline ranking and the rankings obtained across the analyzed scenarios.

Table 6. Sensitivity analysis of PROMETHEE II ranking under variations in criterion weights.

Scenario	Weight Modification	Scratch	Tynker	MakeCode	Alice	Blockly	Rank Reversal	Stability Level
Baseline scenario	ROC weights	1	2	3	4	5	No	Stable
Visual accessibility	+10%	1	2	3	5	4	Yes	Moderate sensitivity
	+20%	1	2	3	5	4	Yes	Moderate sensitivity
	−10%	1	2	3	4	5	No	Stable
Cognitive complexity	+10%	2	1	3	4	5	Yes	High sensitivity
	+20%	4	1	2	3	5	Yes	High sensitivity
	−10%	1	2	3	5	4	Yes	Moderate sensitivity
Pedagogical support	+10%	1	2	3	4	5	No	Stable
	+20%	1	2	3	4	5	No	Stable
	−10%	1	2	3	5	4	Yes	Moderate sensitivity
Interaction simplicity	+10%	1	2	3	5	4	Yes	Moderate sensitivity
	+20%	1	2	3	5	4	Yes	Moderate sensitivity
	−10%	1	2	3	4	5	No	Stable

Table 7. Comparison of baseline ranking and sensitivity analysis results.

Alternative	Baseline Rank	Range of Rank Variation	Sensitivity Interpretation
Scratch	1	1–4	Moderate sensitivity
Tynker	2	1–2	Low sensitivity
MakeCode	3	2–3	Low to moderate sensitivity
Alice	4	3–5	High sensitivity
Blockly	5	4–5	Low to moderate sensitivity

The sensitivity analysis focused on the four highest-ranked criteria: visual accessibility, cognitive complexity, pedagogical support, and interaction simplicity. These criteria were selected because they received the highest rankings from respondents and obtained the largest ROC weights. Together, they account for 70.42% of the total weight distribution and therefore exert the strongest influence on the final ranking results. Consequently, sensitivity analysis was focused on these criteria, as variations in lower-weighted criteria would be expected to have a substantially smaller impact on the overall ranking. For all criteria, weight changes of +10% and +20% were applied to assess the impact of positive deviations on the final ranking of alternatives. In comparison, negative variation was limited to −10% to maintain scenario comparability and a stable sensitivity analysis structure.

In the base scenario, the PROMETHEE II method ranked the alternatives as follows: Scratch, Tynker, MakeCode, Alice, and Blockly. The same order was used as the reference display in the tables presenting the sensitivity analysis results (Tables 6 and 7).

The results in Table 6 indicate a relatively stable and robust ranking across variations in criterion weights. The greatest stability was observed for the criteria visual accessibility and interactional simplicity under negative weight variation (−10%), and for the criterion

pedagogical support under positive variations (+10% and +20%), where the ranking of alternatives did not change relative to the baseline scenario. These results indicate that these criteria have a limited influence on the overall ranking structure.

On the other hand, the cognitive complexity criterion emerged as the most influential for model stability, since variations in its weight lead to the largest changes in the ranking of alternatives. To a lesser extent, the interactional simplicity criterion also showed changes, with shifts in positions among the lower-ranked alternatives.

The results of the sensitivity analysis indicate that the top three alternatives (Scratch, Tynker, and MakeCode) remain stable across most scenarios analyzed. In contrast, ranking changes primarily affect the last two alternatives (Alice and Blockly). These results confirm the high stability of the top-ranked alternatives, while the lower-ranked alternatives are more sensitive to changes in criterion weights.

Table 7 compares the baseline ranking of alternatives with the results across scenarios with varying criterion weights. To assess the stability of the ranking, a five-point sensitivity scale (1–5) was used, defined in this study as operationalizing the PROMETHEE-based sensitivity analysis results [45].

The results in Table 7 summarize the stability of alternative rankings across the sensitivity analysis and indicate varying levels of robustness among software solutions in response to changes in criterion weights. Scratch is the top-ranked alternative but shows sensitivity in scenarios with increased weighting of the cognitive complexity criterion, shifting as low as fourth place, indicating moderate sensitivity.

Tynker appears as a stable alternative with low sensitivity. Its position mainly alternates between first and second place, indicating balanced results in relation to the most influential criteria. MakeCode demonstrates low-to-moderate sensitivity and consistently ranks in the middle. Minor fluctuations between second and third place occur mainly when the weighting of the cognitive complexity criterion is more pronounced, indicating a relatively balanced profile.

Alice demonstrates high sensitivity, as her position changes significantly with shifts in the weights of the criteria. In scenarios with increased cognitive complexity and reduced weight for inclusive criteria, Alice shifts toward the middle of the ranking list, whereas in other scenarios, she remains in lower positions.

Blockly also demonstrates low to moderate sensitivity, but in most scenarios, it remains in last place. Although it occasionally switches positions with the alternative Alice, its overall ranking stays at the bottom, indicating limited competitiveness relative to the defined evaluation criteria.

The sensitivity analysis demonstrates that the ranking of alternatives remains relatively stable under the considered weight variations. Although cognitive complexity produces the most pronounced ranking changes, the three highest-ranked alternatives generally preserve their leading positions. These findings indicate that the proposed ROC–PROMETHEE II framework is robust to moderate preference shifts within the most influential evaluation criteria. Future research may investigate more extreme weight modifications and additional scenario configurations to further evaluate the robustness of the highest-ranked alternatives under stronger preference shifts.

5. Discussion

The research results showed that criteria related to accessibility and cognitive load reduction play a central role in evaluating and selecting software solutions for visual programming intended for children with special educational needs. The highest weights were assigned to visual accessibility and cognitive complexity, indicating that, when choosing software solutions, the greatest importance is placed on intuitive, clearly structured en-

vironments that facilitate content understanding and user interaction with the software. These results are consistent with research showing that block-based programming environments reduce cognitive load and make it easier for beginners and students with diverse educational needs to master fundamental programming concepts [35,64].

High-ranking criteria for pedagogical support and interactional simplicity confirm that the quality of educational software depends not only on technical capabilities but also on its ability to support the learning process, motivation, and students' autonomous activity. A clear interface structure, a gradual introduction to programming concepts, and visually guided interaction can significantly affect engagement and learning efficiency among children with special educational needs. The results are consistent with research in inclusive digital education, which emphasizes the importance of adaptive and interactive environments for students with diverse educational needs [65,66].

In the PROMETHEE II ranking, Scratch stood out as the highest-rated alternative, achieving the most favorable balance across the criteria analyzed. Its dominant position primarily stems from high scores on the criteria most influential in the final ranking, particularly visual accessibility, cognitive complexity, pedagogical support, and interactional simplicity. These results are consistent with other studies that recognize Scratch as one of the most accessible environments for introductory programming and for developing computational logic in children [67,68]. Its intuitive drag-and-drop approach, visual organization of programming elements, and wide availability of educational resources make it especially suitable for an inclusive educational environment [67].

Tynker ranked second and demonstrated stable performance across most criteria. Its strengths are particularly pronounced in multisensory learning and pedagogical support, positively affecting student motivation and engagement during the learning process. Make-Code displayed a balanced performance profile and a relatively high level of stability across various sensitivity analyses. Its strengths primarily relate to technical accessibility and compatibility with the educational environment, while somewhat lower scores in inclusiveness and pedagogical support limit its overall ranking. The stability of this alternative suggests it could be a reliable solution in educational settings where evaluation priorities vary by the needs of teachers, students, and institutions. In contrast, Alice and Blockly achieved lower results than the leading alternatives, especially in the criteria related to accessibility, inclusiveness, and cognitive simplicity. Although Alice enables more advanced visualizations and more complex programming concepts, the increased complexity of the interaction can be a limitation for students who require additional educational support.

The results of the GAIA analysis showed high consistency across most criteria, indicating a relatively coherent decision-making model and low conflict among the dominant evaluation criteria. The criteria of cognitive complexity and software solution cost stood out because their vectors pointed in different directions, indicating a conflicting relationship with the criteria related to the software solutions' pedagogical and inclusive features. This distribution confirms that lower costs or higher functional complexity do not necessarily lead to greater educational effectiveness, especially when working with children with special educational needs.

Sensitivity analysis showed that the proposed ROC-PROMETHEE II framework was stable across most scenarios. The largest changes in the ranking of alternatives occurred when the weight of the cognitive complexity criterion changed, confirming its significant impact on the final evaluation results. The stability of the top alternatives across scenarios with varying criterion weights confirms the reliability and practical applicability of the proposed hybrid MCDM framework for evaluating and selecting educational software solutions for children with special educational needs.

6. Conclusions

This study presents a hybrid multi-criteria decision-making (MCDM) framework for evaluating, ranking, and selecting visual programming software solutions for children with special educational needs. Drawing on the specificities of the educational context and the needs of the target population, a set of criteria tailored to the research domain was defined. It was observed that no unified, standardized framework for this type of evaluation exists in the current literature.

The research used a structured questionnaire administered to all schools in Serbia that educate children with special educational needs. The resulting dataset provided the empirical basis for determining criterion importance and constructing the weighting structure used within the decision model. The ROC method was used to determine criterion weights, and the PROMETHEE II method was used to rank the alternatives. Additional GAIA and sensitivity analyses were conducted to assess the stability and interpretability of the results.

Regarding RQ1, the results indicate that visual accessibility, cognitive complexity, pedagogical support, and interactional simplicity are the most significant criteria in decision-making. The results confirm the dominant role of educational-cognitive factors over technical and economic aspects.

Regarding RQ2, the ROC–PROMETHEE II model produced a clear ranking of the alternatives, with Scratch in first place, followed by Tynker and MakeCode, while Alice and Blockly ranked lower. The results indicate that software solutions with a simpler interface and stronger pedagogical support perform better under the defined criteria.

For RQ3, the sensitivity analysis and GAIA projection confirm the ranking's robustness to changes in criterion weights ($\pm 10\%$ and $+20\%$). Although the preference flows change numerically, the ranking structure remains stable, especially for the top three alternatives (Scratch, Tynker, and MakeCode). Most changes occur among the lower-ranked alternatives (Alice and Blockly), indicating greater sensitivity in these solutions. The analysis shows that the model is most sensitive to changes in the cognitive complexity criterion, while the other criteria have less impact on the final ranking.

The work's contribution is an integrated ROC–PROMETHEE II framework for multi-criteria evaluation, ranking, and selecting visual programming software solutions in an inclusive educational context, based on empirically collected data. The approach enables structured, transparent decision-making by combining objective determination of criterion weights with the PROMETHEE II method for ranking alternatives. Another contribution is the application of the GAIA projection and sensitivity analysis, which, in addition to the final ranking, assesses the stability of the results under changes in input parameters. In this way, the methodological framework enables not only the selection of the best alternative but also insight into the robustness of solutions across different decision-making scenarios.

The proposed framework has practical value, as it can support decision-making in educational institutions when selecting appropriate educational tools, thereby improving the quality of teaching and the inclusiveness of education. The study has several limitations, including the number of alternatives considered, the use of a predefined set of criteria, the limited direct experience of respondents with multiple visual programming environments, the heterogeneous nature of the target population, and the limited representativeness of the study sample. Future research may expand the set of criteria, include a larger number of software solutions, and apply additional multi-criteria methods for comparative analysis and further validation of the results' robustness. It may also incorporate qualitative data collected through interviews or focus groups with teachers, professional associates, and software developers, examine the practical application of the proposed framework in

educational settings, and explore the long-term effects of visual programming environments on the development of children with special educational needs.

Author Contributions: M.K., D.S. and L.K.; methodology, M.K., D.S. and L.K.; software, M.K.; investigation, M.K.; formal analysis, M.K.; writing—original draft preparation, M.K.; writing—review and editing, M.K., D.S. and L.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study because it involved an anonymous voluntary survey.

Informed Consent Statement: Informed consent was obtained from all participants involved in the study.

Data Availability Statement: The data are not publicly available due to privacy and ethical restrictions, as they include responses collected from educational institutions and survey participants.

Acknowledgments: The authors acknowledge the support of the participating schools and respondents involved in the data collection process.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GAIA	Geometrical Analysis for Interactive Aid
MCDM	Multi-Criteria Decision Making
PROMETHEE II	Preference Ranking Organization Method for Enrichment Evaluation
ROC	Rank Order Centroid

References

1. Sovilj-Nikić, S.; Ćorović, J. Examining the Possibility of Using Visual Programming Language Scratch in Preschool Education for Learning Arithmetic Operations. *Int. J. Cogn. Res. Sci. Eng. Educ. (IJCRSEE)* **2025**, *13*, 413–426. [\[CrossRef\]](#)
2. Yang, Z.; Blake-West, J.; Yang, D.; Bers, M. The Impact of a Block-Based Visual Programming Curriculum: Untangling Coding Skills and Computational Thinking. *Learn. Instr.* **2025**, *95*, 102041. [\[CrossRef\]](#)
3. Fiori, S.; Strong, G.; Dukes, J. How Are We Teaching Programming to Students with Intellectual Disabilities? A Systematic Review of the Literature. In Proceedings of the 25th Koli Calling International Conference on Computing Education Research, Koli, Finland, 11–14 November 2025; pp. 1–12. [\[CrossRef\]](#)
4. Stimoli, M.A.; Di Blasi, F.D.; Maccarrone, S.; Costanzo, A.A.; Occhipinti, P.; Buono, S. Teaching Basic Computer Programming to Young Adults with Intellectual Disability. *Front. Educ.* **2024**, *9*, 1484921. [\[CrossRef\]](#)
5. Ahmed, S.; Rahman, M.S.; Kaiser, M.S.; Hosen, A.S.M.S. Advancing Personalized and Inclusive Education for Students with Disability Through Artificial Intelligence: Perspectives, Challenges, and Opportunities. *Digital* **2025**, *5*, 11. [\[CrossRef\]](#)
6. Azhar, N.A.; Radzi, N.A.M.; Wan Ahmad, W.S.H.M. Multi-Criteria Decision Making: A Systematic Review. *Recent Adv. Electr. Electron. Eng.* **2021**, *14*, 779–801. [\[CrossRef\]](#)
7. Więckowski, J.; Sałabun, W.; Kizielewicz, B.; Bączkiewicz, A.; Shekhovtsov, A.; Paradowski, B.; Wątróbski, J. Recent Advances in Multi-Criteria Decision Analysis: A Comprehensive Review of Applications and Trends. *Int. J. Knowl.-Based Intell. Eng. Syst.* **2023**, *27*, 367–393. [\[CrossRef\]](#)
8. Basílio, M.P.; Pereira, V.; Costa, H.G.; Santos, M.; Ghosh, A. A Systematic Review of the Applications of Multi-Criteria Decision Aid Methods (1977–2022). *Electronics* **2022**, *11*, 1720. [\[CrossRef\]](#)
9. Shao, M.; Han, Z.; Sun, J.; Xiao, C.; Zhang, S.; Zhao, Y. A Review of Multi-Criteria Decision Making Applications for Renewable Energy Site Selection. *Renew. Energy* **2020**, *157*, 377–403. [\[CrossRef\]](#)
10. Topaloğlu, F. Development of a New Hybrid Method for Multi-Criteria Decision Making (MCDM) Approach: A Case Study for Facility Location Selection. *Oper. Res. Int. J.* **2024**, *24*, 60. [\[CrossRef\]](#)
11. Kocaman, H.; Asan, U. Integration Modes Between MCDM Methods and Machine Learning Algorithms: A Structured Approach for Framework Development. *Mathematics* **2025**, *14*, 33. [\[CrossRef\]](#)

12. Susmaga, R.; Szczech, I.; Brzezinski, D. Towards Explainable TOPSIS: Visual Insights into the Effects of Weights and Aggregations on Rankings. *Appl. Soft Comput.* **2024**, *153*, 111279. [CrossRef]
13. Troussas, C.; Krouska, A.; Mylonas, P.; Sgouropoulou, C. Personalized Instructional Strategy Adaptation Using TOPSIS: A Multi-Criteria Decision-Making Approach for Adaptive Learning Systems. *Information* **2025**, *16*, 409. [CrossRef]
14. Alshamsi, A.M.; El-Kassabi, H.; Serhani, M.A.; Bouhaddioui, C. A multi-criteria decision-making (MCDM) approach for data-driven distance learning recommendations. *Educ. Inf. Technol.* **2023**, *28*, 10421–10458. [CrossRef] [PubMed]
15. Basciani, F.; Di Pompeo, D.; Di Rocco, J.; Pierantonio, A. A Customizable Approach to Assess Software Quality through Multi-Criteria Decision Making. *arXiv* **2023**, arXiv:2301.12202. [CrossRef]
16. Shabrina, P.; Marwan, S.; Bennison, A.; Chi, M.; Price, T.; Barnes, T. A Multicriteria Evaluation for Data-Driven Programming Feedback Systems: Accuracy, Effectiveness, Fallibility, and Students' Response. *arXiv* **2022**, arXiv:2208.05326.
17. Albahri, A.S.; Hamid, R.A.; Alzubaidi, L.; Homod, R.Z.; Zidan, K.A.; Mubark, H.; Shayea, G.G.; Albahri, O.S.; Alamoodi, A.H. Prioritizing Complex Health Levels beyond Autism Triage Using Fuzzy Multi-Criteria Decision-Making. *Complex Intell. Syst.* **2024**, *10*, 6159–6188. [CrossRef]
18. Chen, J.; Ismail, A.S.; Xie, X. Comparative Evaluation of Play Environment Design Elements in Autism Schools Using AHP and GRA. *Sci. Rep.* **2025**, *15*, 43207. [CrossRef] [PubMed]
19. Kumar, U.; Banerji, H. Estimating Weightage and Prioritizing Major Domains of Inclusive Education Assessment Framework Using Analytical Hierarchy Process. *Int. J. Syst. Assur. Eng. Manag.* **2025**, *16*, 3440–3451. [CrossRef]
20. Cañete, R.; Picardo, A.; Trueba, P.; Torres, Y.; Peralta, E. A New Multi-Criteria Decision-Making Approach for the Design and Selection of Materials and Manufacturing Processes of Toys for Children with Autism. *Mater. Today Commun.* **2024**, *40*, 109709. [CrossRef]
21. Lin, Y. A Deep Learning-Based Visual Strategy Intervention Study for Children with Autism Spectrum Disorders. *Eur. Rev. Med. Pharmacol. Sci.* **2023**, *27*, 4914–4928. [CrossRef] [PubMed]
22. Chaudhry, U.B.; Afzal, S.; Wajid, A. Machine Learning Techniques for Identifying Self-Care Problems in Disabled Children. *Pak. J. Sci. Res.* **2023**, *3*, 158–168. Available online: <https://pjosr.com/index.php/pjosr/article/view/1041> (accessed on 12 May 2026). [CrossRef]
23. Mountapmbeme, A.; Ludi, S. Designing Accessible Block-Based Programming Environments for Persons with Visual Impairments. *Univ. Access Inf. Soc.* **2025**, *24*, 2351–2382. [CrossRef]
24. Zubair, M.S.; Brown, D.J.; Hughes-Roberts, T.; Bates, M. Designing Accessible Visual Programming Tools for Children with Autism Spectrum Condition. *Univ. Access Inf. Soc.* **2023**, *22*, 277–296. [CrossRef]
25. Institute for the Improvement of Education and Upbringing (Zavod za unapređivanje obrazovanja i vaspitanja). Educational Institutions in the Republic of Serbia. Available online: <https://www.zuov.gov.rs/starisajt/obrazovno-vaspitanje/ustanove/index.html> (accessed on 11 May 2026).
26. Statistical Office of the Republic of Serbia. Schools and Students at the Beginning of the School Year 2024/2025. Available online: <https://publikacije.stat.gov.rs/G2025/Pdf/G20252058.pdf> (accessed on 11 May 2026).
27. Love, H.R.; Horn, E. Definition, Context, Quality: Current Issues in Research Examining High-Quality Inclusive Education. *Top. Early Child. Spec. Educ.* **2021**, *40*, 204–216. [CrossRef]
28. UNESCO. *Inclusion and Education: All Means All*; UNESCO Publishing: Paris, France, 2020. Available online: <https://unesdoc.unesco.org/ark:/48223/pf0000373718> (accessed on 11 May 2026).
29. Al-Gerafi, M.A.M.; Goswami, S.S.; Sahoo, S.K.; Kumar, R.; Simic, V.; Bacanin, N.; Naveed, Q.N.; Lasisi, A. Promoting Inclusivity in Education amid the Post-COVID-19 Challenges: An Interval-Valued Fuzzy Model for Pedagogy Method Selection. *Int. J. Manag. Educ.* **2024**, *22*, 101018. [CrossRef]
30. European Commission. European Framework for the Digital Competence of Educators (DigCompEdu). Publications Office of the European Union. 2017. Available online: <https://publications.jrc.ec.europa.eu/repository/handle/JRC107466> (accessed on 11 May 2026).
31. Spector, J.M. *Handbook of Research on Educational Communications and Technology*, 4th ed.; Springer: New York, NY, USA, 2014.
32. Velasquez, M.; Hester, P.T. An Analysis of Multi-Criteria Decision Making Methods. 2013. Available online: <https://www.researchgate.net/publication/275960103> (accessed on 11 May 2026).
33. Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. Scratch: Programming for All. *Commun. ACM* **2009**, *52*, 60–67. [CrossRef]
34. Lye, S.Y.; Koh, J.H.L. Review on Teaching and Learning of Computational Thinking through Programming: What Is next for K-12? *Comput. Hum. Behav.* **2014**, *41*, 51–61. [CrossRef]
35. Price, T.W.; Barnes, T. Comparing Textual and Block Interfaces in a Novice Programming Environment. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research, Omaha, NE, USA, 9–13 August 2015*; ACM: New York, NY, USA, 2015; pp. 91–99. [CrossRef]

36. Wastiau, P.; Blamire, R.; Kearney, C.; Quittre, V.; Van De Gaer, E.; Monseur, C. The Use of ICT in Education: A Survey of Schools in Europe. *Eur. J. Educ.* **2013**, *48*, 11–27. [\[CrossRef\]](#)
37. Hatefi, M.A. An Improved Rank Order Centroid Method (IROC) for Criteria Weight Estimation: An Application in the Engine/Vehicle Selection Problem. *Informatica* **2023**, *34*, 249–270. [\[CrossRef\]](#)
38. Ayan, B.; Abacioğlu, S.; Basilio, M.P. A Comprehensive Review of the Novel Weighting Methods for Multi-Criteria Decision-Making. *Information* **2023**, *14*, 285. [\[CrossRef\]](#)
39. Sureeyatanapas, P. Comparison of Rank-Based Weighting Methods for Multi-Criteria Decision Making. *KKU Eng. J.* **2016**, *43*, 376–379. [\[CrossRef\]](#)
40. Ari, R.; Altinay, Z.; Altinay, F.; Dagli, G.; Ari, E. Sustainable Management and Policies: The Roles of Stakeholders in the Practice of Inclusive Education in Digital Transformation. *Electronics* **2022**, *11*, 585. [\[CrossRef\]](#)
41. Mardani, A.; Jusoh, A.; Md Nor, K.; Khalifah, Z.; Zakwan, N.; Valipour, A. Multiple Criteria Decision-Making Techniques and Their Applications—A Review of the Literature from 2000 to 2014. *Econ. Res.-Ekonom. Istraž.* **2015**, *28*, 516–571. [\[CrossRef\]](#)
42. Krishnan, A.R. Past Efforts in Determining Suitable Normalization Methods for Multi-Criteria Decision-Making: A Short Survey. *Front. Big Data* **2022**, *5*, 990699. [\[CrossRef\]](#) [\[PubMed\]](#)
43. Ishizaka, A.; Resce, G.; Mareschal, B. Visual Management of Performance with PROMETHEE Productivity Analysis. *Soft Comput.* **2018**, *22*, 7325–7338. [\[CrossRef\]](#)
44. Lellis Moreira, M.Â.; Simões Gomes, C.F.; Dos Santos, M.; Da Silva Júnior, A.C.; De Araújo Costa, I.P. Sensitivity Analysis by the PROMETHEE-GAIA Method: Algorithms Evaluation for COVID-19 Prediction. *Procedia Comput. Sci.* **2022**, *199*, 431–438. [\[CrossRef\]](#) [\[PubMed\]](#)
45. Genc, T. Sensitivity Analysis on PROMETHEE and TOPSIS Weights. *Int. J. Manag. Decis. Mak.* **2014**, *13*, 403. [\[CrossRef\]](#)
46. Kumar, R.; Pamucar, D. A Comprehensive and Systematic Review of Multi-Criteria Decision-Making (MCDM) Methods to Solve Decision-Making Problems: Two Decades from 2004 to 2024. *Spec. Decis. Mak. Appl.* **2025**, *2*, 177–196. [\[CrossRef\]](#)
47. Coquelet, B.; Dejaegere, G.; De Smet, Y. Evaluating the Promethee II Ranking Quality. *Algorithms* **2025**, *18*, 597. [\[CrossRef\]](#)
48. Scratch Foundation. About Scratch. Available online: <https://scratch.mit.edu/about> (accessed on 11 May 2026).
49. Microsoft. MakeCode Documentation. 2025. Available online: <https://microsoft.github.io/pxt-docs/> (accessed on 11 May 2026).
50. Google. Blockly Developer Documentation. 2025. Available online: <https://developers.google.com/blockly> (accessed on 11 May 2026).
51. Tynker. Educator Guide and Curriculum Overview. 2025. Available online: <https://www.tynker.com/support/Tynker%20User%20Guide%20for%20Educators.pdf> (accessed on 11 May 2026).
52. Carnegie Mellon University. Alice Programming Environment. 2025. Available online: <https://www.alice.org> (accessed on 11 May 2026).
53. Weintrop, D.; Wilensky, U. Using Commutative Assessments to Compare Conceptual Understanding in Blocks-Based and Text-Based Programs. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research, Omaha, NE, USA, 9–13 August 2015*; ACM: New York, NY, USA, 2015; pp. 101–110.
54. Bau, D.; Gray, J.; Kelleher, C.; Sheldon, J.; Turbak, F. Learnable Programming: Blocks and Beyond. *Commun. ACM* **2017**, *60*, 72–80. [\[CrossRef\]](#)
55. Wang, W.; Kwatra, A.; Skripchuk, J.; Gomes, N.; Milliken, A.; Martens, C.; Barnes, T.; Price, T. Novices’ Learning Barriers When Using Code Examples in Open-Ended Programming. In *Proceedings of 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, Virtual Event Germany, 26 June–1 July 2021*; ACM: New York, NY, USA, 2021; pp. 394–400.
56. Grover, S.; Pea, R. Computational Thinking in K–12: A Review of the State of the Field. *Educ. Res.* **2013**, *42*, 38–43. [\[CrossRef\]](#)
57. Adorni, G.; Piatti, A.; Bumbacher, E.; Negrini, L.; Mondada, F.; Assaf, D.; Mangili, F.; Gambardella, L. FADE-CTP: A Framework for the Analysis and Design of Educational Computational Thinking Problems. *Technol. Knowl. Learn.* **2025**, *30*, 1073–1148. [\[CrossRef\]](#) [\[PubMed\]](#)
58. Frauenberger, C.; Good, J.; Keay-Bright, W. Designing Technology for Children with Special Needs: Bridging Perspectives through Participatory Design. *CoDesign* **2011**, *7*, 1–28. [\[CrossRef\]](#)
59. Mayer, R. *Multimedia Learning*, 3rd ed.; Cambridge University Press: Cambridge, UK, 2020.
60. Joshi, A.; Kale, S.; Chandel, S.; Pal, D. Likert Scale: Explored and Explained. *Br. J. Appl. Sci. Technol.* **2015**, *7*, 396–403. [\[CrossRef\]](#)
61. Brans, J.-P.; De Smet, Y. PROMETHEE Methods. In *Multiple Criteria Decision Analysis*; Greco, S., Ehrgott, M., Figueira, J.R., Eds.; International Series in Operations Research & Management Science; Springer: New York, NY, USA, 2016; Volume 233, pp. 187–219.
62. Mareschal, B.; De Smet, Y. *Visual PROMETHEE User Manual*; PROMETHEE-GAIA: Brussels, Belgium, 2024. Available online: <https://www.promethee-gaia.net/VisualPROMETHEE> (accessed on 8 May 2026).
63. Sapkota, M.; Arora, M.; Malano, H.; Sharma, A.; Moglia, M. Integrated Evaluation of Hybrid Water Supply Systems Using a PROMETHEE–GAIA Approach. *Water* **2018**, *10*, 610. [\[CrossRef\]](#)

64. Weintrop, D.; Wilensky, U. To Block or Not to Block, That Is the Question: Students' Perceptions of Blocks-Based Programming. In *Proceedings of the 14th International Conference on Interaction Design and Children, Boston, MA, USA, 21–24 June 2015*; ACM: New York, NY, USA, 2015; pp. 199–208.
65. Al-Azawei, A.; Serenelli, F.; Lundqvist, K. Universal Design for Learning (UDL): A Content Analysis of Peer Reviewed Journals from 2012 to 2015. *JoSoTL* **2016**, *16*, 39–56. [[CrossRef](#)]
66. Bers, M.U. *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*, 2nd ed.; Routledge: Oxfordshire, UK, 2020.
67. Sáez-López, J.-M.; Román-González, M.; Vázquez-Cano, E. Visual Programming Languages Integrated across the Curriculum in Elementary School: A Two Year Case Study Using “Scratch” in Five Schools. *Comput. Educ.* **2016**, *97*, 129–141. [[CrossRef](#)]
68. Moreno-Leon, J.; Robles, G.; Roman-Gonzalez, M. Comparing Computational Thinking Development Assessment Scores with Software Complexity Metrics. In *Proceedings of the 2016 IEEE Global Engineering Education Conference (EDUCON)*; IEEE: Abu Dhabi, United Arab Emirates, 2016; pp. 1040–1045.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.