Revue-IRS

Revue Internationale de la Recherche Scientifique
(Revue-IRS)
ISSN : 2958-8413
Vol. 2, No. 2, Avril 2024

USE OF PEDAGOGICAL TOOLS FOR TEACHING ALGORITHMS AND CODING IN FORM 1 AND FORM 2 OF SECONDARY SCHOOLS IN BUKAVU

Bally KASAMBI BALIMWENGU
Étudiant-Chercheur en Didactique de l'Informatique à l'École Doctorale de l'Institut Supérieur Pédagogique de Bukavu
Unité de Recherche en Technologie de l'Information et de la Communication (URETIC)
République Démocratique du Congo

Paulin BAPOLISI BAHUGA
Enseignant-Chercheur à l'École Doctorale de l'Institut Supérieur Pédagogique de Bukavu
République Démocratique du Congo

Deogratias MBILIZI MWISIMBWA
Enseignant-Chercheur à l'Institut Supérieur Pédagogique de Bukavu
Unité de Recherche en Technologie de l'Information et de la Communication (URETIC)
République Démocratique du Congo

Pascal AKILIMALI BAMALEMBUKO
Doctorant à l'École Doctorale de l'Institut Supérieur Pédagogique de Bukavu
Unité de Recherche en Technologie de l'Information et de la Communication (URETIC)
République Démocratique du Congo

NESTOR MUHIMA
Étudiant-Chercheur en Didactique de Psycho-Pédagogie à l'École Doctorale de l'Institut Supérieur Pédagogique de Bukavu
République Démocratique du Congo

Judith SIFA BAGULA
Étudiante-Chercheure en Didactique de l'Informatique à l'École Doctorale de l'Institut Supérieur Pédagogique de Bukavu
Unité de Recherche en Technologie de l'Information et de la Communication (URETIC)
République Démocratique du Congo

Rodrigue KAZILI MARTHIN
Enseignant-Chercheur à l'Institut Supérieur Pédagogique de Kamituga

Abstract: The DR Congo's Ministry of Primary, Secondary and Technical Education has embarked on a new, gradual educational reform. It introduces elements of algorithms and coding into the final cycle of basic education, and brings new features to Information and Communication Technology courses. This new program is supported by a guide and prescribes the situation-based approach as a methodological guideline to be used. Nevertheless, this reform lacks accompanying measures, especially in terms of the choice of pedagogical and didactic tools to support the methodological aspect envisaged by the supervisory ministry.

To this end, this study is based on qualitative data collected from teachers of Information and Communication Technology courses in the first and second years of science on the choice of environments (teaching tools) used to teach the concepts of algorithms and coding. SPSS statistical software was used for data analysis. The results showed that the choice of these pedagogical and/or didactic tools for algorithm development is a function of their ease of use by learners, their clear and concise feedback to learners; the choice of programming languages is a function firstly of their ease of comprehension by learners, secondly because they are recommended by the Ministry and thirdly because they have a large community and offer a variety of resources.

Keywords: Teaching tools; Algorithms, Algorithms, Coding, Bukavu.
Résumé : Le ministère de l'enseignement primaire, secondaire et technique de la République démocratique du Congo s'est lancé dans une nouvelle réforme graduelle de l'éducation. Elle introduit des éléments d'algorithmes et de codage dans le cycle final de l'éducation de base et apporte des nouveautés dans les cours de sciences. Ce nouveau programme est soutenu par un guide et prescrit l'approche par situations comme un guide méthodologique à utiliser. Néanmoins, cette réforme manque de mesures d'accompagnement, notamment en ce qui concerne le choix des outils pédagogiques et didactiques pour soutenir l'aspect méthodologique envisagé par le ministère de tutelle.
A cet effet, cette étude se base sur des données qualitatives recueillies auprès des enseignants du cours de Technologie de l'Information et de la Communication de la première et deuxième années scientifique sur le choix des environnements (outils pédagogiques) utilisés dans l'enseignement des concepts d'algorithmique et de codage. Le logiciel statistique SPSS a été utilisé pour l'analyse des données. Les résultats ont montré que le choix de ces outils pédagogiques et/ou didactiques pour le développement d'algorithmes est fonction de leur facilité d'utilisation par les apprenants, de leur retour d'information clair et concis aux apprenants ; le choix des langages de programmation est fonction premièrement de leur facilité de compréhension par les apprenants, deuxièmement parce qu'ils sont recommandés par le ministère et troisièmement parce qu'ils disposent d'une large communauté et offrent une variété de ressources.
Mots-clés : Outils pédagogiques ; Algorithmes, Algorithmes, Codage, Bukavu.

Digital Object Identifier (DOI): https://doi.org/10.5281/zenodo.10980044

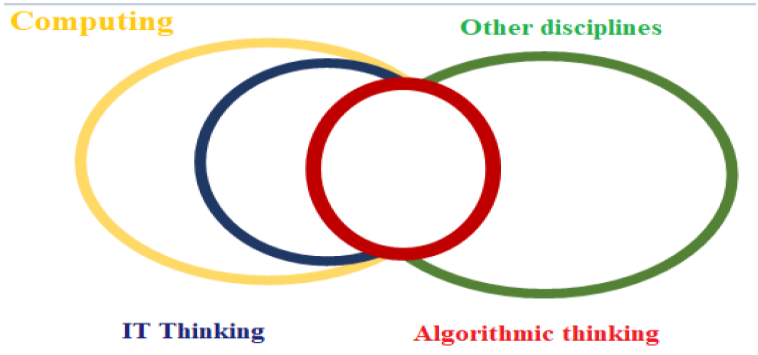
1. Introduction

The DR Congo's Ministry of Primary, Secondary and Technical Education has embarked on a progressive new educational reform of the education system by introducing a new curriculum in the 7th year of basic education in 2018 (EPST, 2018a) and in the 8th year of basic education in 2019 (EPST, 2019a). In the science section, these reforms led to the merging of the Biology-Chemistry and Maths-Physics sections into a single section called Science (EPST, 2019b). These new educational programmes introduce a number of new features: firstly, the course formerly known as Informatics is now called Information and Communication Technology (ICT). Secondly, each educational programme is accompanied by a guide (EPST, 2018b;EPSP, 2019a; EPSP, 2019b).
These reforms, although progressive, require accompanying measures, especially in the choice of tools to be used and in the design of learning situations (Chakri and Riouch, 2021). Very often, a number of teachers come up against problems in choosing tools to support the transmission of knowledge, and still others find it difficult to contextualize situations for teaching algorithms and coding. This is something we experienced during our supervision of the courses, and if the situations are not contextualized and the teaching tools are not well chosen, the learners will find it difficult to grasp the concepts of algorithms and coding that can help them solve certain daily problems (EPSP, 2019b).
According to the curriculum, the ICT teacher must contextualize the subjects learned in chemistry, physics and/or mathematics in order to show learners how to develop algorithms and implement them in a programming language in order to create computer programmes capable of solving mathematical equations. For example, based on an environmental study problem, the ICT teacher, at the request of the physical sciences teacher, asks the students to write an algorithm [and implement it in Python/C] that calculates the molecular mass based on the periodic table of elements.
Algorithms are a complex subject for both teachers and students. According to Faouzia and Hanoune (2007), on the teacher's side, because they have to find the right methods for getting learners who are just starting out to

assimilate fairly abstract concepts. Studies have shown that writing a good programme requires the prior development of a good algorithm.
For the sake of improving the quality of the Congolese education system as described by D.M. Ntambwe and G.M. Bambanota (2018), and in order to achieve the sustainable development goals advocated by the United Nations in the Democratic Republic of Congo (United Nations), it was useful for us to explore the tools for developing algorithms in the teaching of algorithmic concepts in Science first and second forms in the city of Bukavu.
This study problem focuses on using suitable environments for developing and implementing algorithms in accordance with the new educational programme and the Congolese context. We therefore wish to answer the following question: Do teachers of Information and Communication Technologies (ICT) use appropriate teaching environments and tools for developing and implementing algorithms?
This research is guided by the following hypothesis: ICT teachers in Science Forms 1 and 2 would not be using adequate (appropriate) environments (software) and pedagogical tools for developing and implementing algorithms in their teaching to enable students to solve certain daily problems. This study is of particular interest because it falls within the context of research aimed at evaluating, improving and enhancing quality teaching.

2. REVIEW OF LITERATURE

The concept of "algorithm" dates back to the tenth century through the work of the mathematician Al KWARISMI (Arsac, 1991; Ralahady, 2022).
There is a range of definitions of the concept "algorithm". According to ZEGOUR D. E.(2020), an algorithm is the result of a logical approach to solving a problem.
MODESTE (2012) points out that an algorithm is a problem-solving procedure that applies to a family of instances of the problem, so it has a beginning and an end, and the response (result) resulting from its execution is called the output of the algorithm; he is further qualified by Beffara et al. (2023) in these terms: "algorithmics is a finite sequence of instructions for solving a problem".
In our opinion, the best definition of algorithm is taken from the Transmath Cycle 4 textbook (Nathan, 2016), we quote:
- An algorithm describes the logical process of a program. It shows the structure of the program and its variables.
- Once developed, the algorithm is coded in a programme language.
The *Cahier d'algorithmique et de programmation Cycle 4*(Beffara et al., 2017), corroborates the previous definitions and writes:
- An algorithm is a sequence of instructions to be applied within a logical framework to solve a problem and quickly obtain a result. It is written by hand or using software in a language that everyone can understand.
- It is therefore used to prepare the writing of a computer programme.
From the above, we can deduce that writing algorithms requires 2 elements: a certain logic to be followed and the environments (paper and pen, specific software) for its development. So, in relation to logic, the study of algorithms is called "algorithmics" (Blanvillain, 2021).
There is a close link between computer science, computational thinking and algorithms (Beffara *et al.*, 2017;Marquet, 2022). Figure 1 below shows the skills developed in computational and algorithmic thinking.
Figure 1: Interaction between computing, computational thinking and algorithmic.



Source : (Denis, 2020) cited by Marquet (Marquet, 2022)

In old times, teaching computer science and specifically programming consisted of describing flowcharts as a graphic representation of the structure of a program. Given the complexity of algorithms and programs, this proved to be confusing and had to be put aside (Arsac, 1991). Nowadays, algorithms are written in pseudo-codes and translated into software (environments) for developing algorithms. A pseudo-code is a combination of

mathematical languages and programming language principles that allow algorithms to be expressed without evoking the particularities of a given programming language (Modeste, 2012).
There are software environments [such as LOGO, the very first to exist in the 1980s (Lagrange and Rogalski, 2017)] that enable, or at least facilitate, the teaching and learning of algorithms today, some of which are visual "cases of Scratch" and others that are not "Algobox example". These environments provide progressive assistance to learners with difficulties; in other words, they help them to design and implement algorithms.

What about programming?
According to computer jargon, programming is the art of debugging a file; in other words, it is the art of writing computer (software) programs.
EDSGER Dijkstra used to say, "Testing a program can be used to show that it contains errors, never that it is correct" (Arsac, 1991).
From this definition, we can see that programming is also the art of correcting errors in a program.

What is the difference between coding and programming?
Two distinctions are made by a Karsenti article (Karsenti, 2020). Firstly, the coder has no formal knowledge of computing, and is often considered a neophyte (beginner/novice), whereas the programmer is a professional in the field. Secondly, it's the playful, academic aspect of the term "coder", whereas for the programmer it's generally the lucrative aspect.
So coding is the term used much more in the school and academic context. It is less formal and more fun than program, which can be a term that encompasses all aspects of software creation (specifications, design, implementation, testing, updating).
Karsenti (2020) proposes 12 reasons for learning to code at school, an initiative taken up by the Ministry (EPST) in charge of education in the DRC, which offers algorithms and coding in Python or C from the Final Cycle of Base Education (FCBE) up to the 4th year of science.
To code, you need to choose a programming language, depending on the specialist field, which enables you to define a set of instructions to be carried out by the computer when executing a program (Nguyen, 2005). In a way, it is a set of rules and conventions for writing programs.
There are many different programming languages, each with its own particularities; some are better suited to the development of desktop applications, others for websites, mobile applications (Android/iOS), etc.
In its new program (in use today), the Congolese Ministry of Education has not justified the choice of using Python or C, although we believe that this is due to the importance of Python in today's world (Eidelman, 2020); France also recommends it for high schools and colleges (Meyer and Modeste, 2022).
What do we need to code or program?
Just as blackboards and chalk (notebooks/pens) remain important tools in teaching, there are many tools to use in coding, but some of the most basic in teaching include:

- **Code editors:** These are simple programs that allow you to enter and modify standard code. They are not specific to a given language: Sublime text, Notepad++, Vim, Atom, ...
- **Integrated Development Environments (IDEs):** An IDE is a more comprehensive program offering a range of programming tools, some of which even include compilers, debuggers and interpreters for certain languages (Jeanjean, P., 2022). More often than not, they are specific to certain languages, although they offer the possibility of integrating plugins from other languages, e.g. NetBeans.
- **Compilers:** A compiler is a program that converts source code into machine code. Machine code is a set of instructions that the processor can understand and execute (Delannoy, 2016)**.**
- **Interpreters:** An interpreter is a program that executes the source code directly. Interpreters are not as fast as compilers, but they are easier to debug.

The following is a non-exhaustive list of IDEs:

Table 1: List of IDEs and Code Editors
Source: Our own production

IDE	Supported programming languages
Anaconda	Python
Code::Blocks	C, C++, Python et Fortran
DeVC++	C et C++
Eclipse	Java, C++, C#, PHP, Ruby, Python, JavaScript, TypeScript, HTML, CSS
NetBeans	Java, HTML5, CSS3, JavaScript, PHP, Python, Ruby, SQL, XML, JSON, C/C++, Fortran
Notepad++	C++, C#, Java, PHP, Python, JavaScript, HTML, CSS
PyCharm	Python, Java, PHP, JavaScript, C/C++, Ruby, R, SQL, HTML, CSS, XML, JSON
Sublime Text	Python, JavaScript, HTML, CSS, PHP, Ruby, Go, Perl, Rust, VB.NET.
Visual Studio Code	JavaScript, TypeScript, HTML, CSS, Python, Java, C++, C#, PHP, Ruby, VB.NET

More details: In the table below, we contextualized the list of IDEs in relation to the Python and C languages.

Choosing an implementation language for teaching algorithmic

More often than not, in education, the choice of an implementation language is dictated by the ministry responsible. In Belgium, the choice is made because it is considered to provide training, or because a faculty wants it in the name of its own interests, or because the world of work demands it (Vandeputa and Henryb, 2018). However, this choice may also be made for :

- **Its ease of learning**

Especially the clarity of the syntaxes that describe the set of rules that determine how the code is written. Some syntaxes are easier to learn than others.

But also, the Semantics of a programming language is the very meaning of the code (Delahaye, Jaume and Prevosto, 2005). Some semantics are easier to understand than others, like the Greek alphabet in Languages.

- **The greatness of its community**

A developer community is a group of people sharing an interest in software development, it can be online or offline, and they can be based on a specific programming language, framework, platform or area of interest (BERKACHY, 2017). Examples include Python, C, Java (the largest in the world), ect.

- **A wide range of resources**

The availability of resources can also influence teachers' choice of language. These resources include books, websites and tutorials, etc. (Vandeputa and Henryb, 2018).

3. METHODOLOGICAL APPROACHES

3.1.STUDY POPULATION AND SURVEY SAMPLE

According to the DRC's educational organisation, the city of Bukavu is located in the educational province of South-Kivu 1 (Ilundu, 2019). The latter is run by the Provincial Directorate, the Principal Provincial Inspectorate and the National Directorate for the Control, Preparation of Payroll and Control of Teacher Numbers and School Administrative Staff (DINACOPE).

According to the school geolocation report, South Kivu province has around 2,175 secondary schools, including 396 (18.2%) in the city of Bukavu.

During our field visits, we officially identified 35 schools that organize the science section in the educational province of South Kivu 1 in Bukavu. We worked with a sample of 33 teachers in 4 educational systems: Protestant Conventioned Schools, Catholic Conventioned Schools, Official Schools and Approved Private Schools.

3.2. DATA COLLECTION and PROCESSING

In this research, we used the quasi-experimental method supported by certain techniques. Our variables were measured by Pearson's Chi-square to identify or test the relationship of independence (Milondo, 2016) and

Spearman's Chi-square for qualitative variables, Cramer's V was useful for comparing the intensity of the link between our variables under study.
The documentary technique enabled us to delve into the legal texts (the educational program, guide in support of the program, ministerial decrees, National Education Framework Law, Constitution of the Republic, etc.) that govern education but also to consult virtual libraries on the Internet. The interview technique was useful for interviewing the informants: Statistics officer, teachers, headmasters and Principals) in the South Kivu 1 education sector.
Data collection was based on a survey using a questionnaire written in French, direct observations and interviews with teachers in the first and second forms of science during the 2022-2023 school year. The questionnaires were delivered to teachers at the beginning of the second half of the school year and collected in respect to each teacher's appointment, which made it last for about 3 months, i.e. from February to April 2023. At the end of the school year, we went back to some schools to collect the learners' notebooks in order to compare the contents with the teachers' answers.
We used statistical analysis tools (Excel and SPSS) to analyze and process the data in order to draw up summary reports and data graphs to help us interpret the results.

4. RESULTS

Table 3: Socio-economic and demographic characteristics of the sample.

Variables		Frequency	Percentage
Gender	M	30	90.91%
	F	3	9.09%
Age group	Less than de 25 years	1	3.03%
	Aged 25 to 34 years	20	60.61%
	Aged 35 à 44 years	12	36.36%
Last degree	Bac+5	30	90.91%
	Bac+3	3	9.09%
Graduated at ISP	Yes	30	90.91%
	No	3	9.09%
Teacher qualification	Yes	29	87.88%
	No	4	12.12%
Marital Status	Single	10	30.30%
	Married	22	66.67%
	Divorced	1	3.03%
Seniority	1-4 Years	11	33.33%
	5-8 years	17	51.52%
	9-12 years	1	3.03%
	More than 12 years	4	12.12%
State paid	Yes	10	30.30%
	No	23	69.70%
Management system	Official school	3	9.09%
	Protestant School	9	27.27%
	Catholic school	9	27.27%
	Private school	12	36.36%

Source : Our results
With regard to the variables gender and age, men accounted for 90.91% of respondents, while women represented 9.09%; 60.61% were aged between 25 and 34, implying that the majority of teachers were still young; 36% of teachers were aged between 35 and 44 and 3.03% of respondents were under 25.
In terms of education level (latest qualification), 90.91% of teachers had a Bac+5 degree and 9.09% had a Bac+3. 90.63% of respondents had graduated from a Teacher Training College and 9.3% from other universities.

After processing the variable qualification, we found that 87.88% of teachers are qualified and 12.12% are not. With regard to marital status, 66.67% of teachers are married; 30.30% were still single and 3.03% are divorced. In terms of seniority, 51.51% have been teaching for 5 to 8 years; 33.33% have been teaching for 1 to 4 years; 12.12% have been teaching for more than 12 years, whereas 3.03% have been teaching for 9 to 12 years.
We surveyed 36.36% of public schools, 27.27% of Protestant schools with official agreement, 27.27% of Catholic schools with official agreement, and 9.09% of teachers in official schools. We found that 69.70% are State paid yet, and only 30.30% of teachers are State paid.

Number of periods per week on the ICT course

Table 4: Periods foreseen for the ICT course per week

Is the ICT number of periods per week enough?	Frequency	Percentage
YES	9	27.3%
No	24	72.7%
Total	33	100.0%

Source: Our results in SPSS

With regard to the number of periods in the ICT course, the Ministry provides for 1 hour per week. Looking at the results in the table above, 72.7% of teachers consider that one period a week is insufficient and 27.3% consider that the only is enough.

Teaching methods

Table 5: Teaching methods for algorithms and coding

How do you approach the teaching of algorithms and coding?	Frequency	Percentage
Theory only	9	27.3%
Practice only	0	0.0%
Theory in class → Practice in Lab	16	48/5%
Practice in Lab → Theory in class	0	0/0%
Theory and Practice	8	24/2%
Total	33	100.0%

Source : our SPSS results

The above table shows that 48.5% of teachers prefer to start with theory in the classroom and then practice in the laboratory; 27.3% of teachers limit themselves to theory alone, which may be due to a number of reasons, such as lack of logistical resources (lack of laboratories, unequipped laboratories, breakdown theory or outright resistance to change, etc.) and others (24.2%) prefer to do theory and practice directly.

Best strategy for teaching algorithms and coding

Table 6: Best practice

What is the best strategy ?	Frequency	Percentage
Theory only	0	0.0%
Practice only	0	0.0%
Theory in class → Practice in Lab	29	87.9%
Practice in Lab → Theory in class	0	0.0%
Theory and Practice	3	9.1%
Abstain	1	3.0%
Total	33	100.0%

Source : Our results in SPSS

In relation to best teaching strategy, 87.9% felt that the best strategy was to strategy is to do the theory in class and the practice in the laboratory, while laboratory, while 9.1% of the teachers thought that the best strategy was to is to do the theory and practice directly.

Environments used for teaching algorithms and coding

Table 7: Environments used to develop algorithms

What software environments are used to develop the algorithm?	Frequency	Percentage
Abstain	3	9.1%
Pratiquer l'algorithme	8	24.2%
LARP	5	15.2%
Algobox	10	30.3%
MyPascal	7	21.2%
Other	0	0.0%
Total	33	100.0%

Source: Our results in SPSS

With regard to the use of environments for developing algorithms, 30.3% of teachers used the Algobox environment, 24.2% of teachers used the Pratiquer l'algorithme software, 21.2% used the MyPascal environment, 15.2% used the LARP software, and 9.1% of teachers did not give their opinion on the choice of software for developing algorithms.

Justification of the environments used

Table 8: Justification for the choice of algorithmic development environments

Why do you choose these environments?	Frequency	Percentage
Free, open-source software	7	21.2%
Easy to use	24	72.7%
Near human language	2	6.1%
Total	33	100.0%

Source: Our results in SPSS

This is a summary of the qualitative data collected in the field: 72.7% use an environment because it is easy for learners to use; 21.2% of the teachers use it because it is free and open source, others (6.1% of the teachers) opt for it because its syntax is close to human language, for reasons of anonymity, we represent the author as OSTIN, for him, his choice is for Algobox because it is a pedagogical and didactic tool at the same time, its functionalities are easy to handle even for novices since the instructions are entered in French and prepared in advance on an interface (software window).

Languages used to implement algorithms

Table 9: Language used to implement the algorithms in the scientist classes.

What language do you use to implement your algorithms?	Frequency	Percentage
Abstain	2	6.1%
Python	9	27.3%
C	2	6.1%
C#	0	0.0%
Java	1	3.0%
VBA	7	21.2%
Q-Basic	9	27.3%
Pascal	3	9.1%
Total	33	100.0%

Source : Our results in SPSS

Pearson chi-square: 0.007; Symmetrical measure (Cramer's V): 0.014; Asymptotic significance (Bilateral): 0.935

In terms of the programming language used, 27.3% of teachers implemented the algorithms in Python; 27.3% of teachers implemented the algorithms in Q-Basic; 21.2% of teachers implemented the algorithms in Visual Basic for Application; 9.1% of teachers implemented the algorithms in Pascal; 6.1% of teachers implemented the algorithms in C; only 3% of teachers implemented the algorithms in Java; and 6.1% of teachers did not give their opinion on the choice of programming language.
These results also show that only 33.4% of teachers use the programming languages recommended by the Ministry. The Chi2 test of independence between gender and programming language was significant ($p=0.007<0.05$), which means that gender has a positive impact on the use of programming languages recommended by the Ministry.

Justification of the languages used to implement the algorithms

Table 10: Justification for the choice of programming language environments

What are your preferences for these languages?	Frequency	Percentage
Recommended by the Ministry	7	21.2%
Easy to understand and learn	15	45.5%
Closely related to human language	5	15.2%
Large community/variety of resources	6	18.2%
Total	33	100.0%

Source: Our surveys
With regard to the choice of using programming languages in teaching, 45.5% use it because they think it is easy for learners to understand and master; 21.2% of teachers use a programming language because it is recommended by the Ministry; a further 18.2% use it because it has a large community of researchers/programmers who offer resources and aids for teaching and learning this language, and 15.2% of teachers believe that this is due to the fact that this programming language is close to human language.

Use of integrated development environments

Table 11: Integrated development environments used.

What IDEs are used in teaching algorithmics and coding?	Frequency	Percentage
VS Codes for C	4	12.1%
Netbeans for Java	2	6.1%
Pycharm for Python	2	6.1%
Anaconda/Spyder for Python	8	24.2%
Sublime Text for C	3	9.1%
Code::Blocks for C	4	12.1%
Other	10	30.3%
Total	33	100.0%

Source: Our SPSS results
Pearson chi-square: 0.015; Symmetric measure (Cramer's V): 0.022; Asymptotic significance (Bilateral): 0.903
With regard to integrated development environments, other software not grouped here is used by 30.3% of teachers in these environments or editors include Notepad++, DevC++ for C; 24.2% of teachers prefer to use the Anaconda/Spyder IDE for implementing algorithms in the Python language; 12.1% of teachers use Visual Studio Codes for the C programming language; Code: :Blocks is used by 12.1% of teachers; 6.1% of teachers use Netbeans for Java; 6.1% of teachers use the Pycharm environment for Python implementation.
We had classified the integrated development environments in terms of the program languages chosen and the ease with which learners could use them (pedagogical and didactic tools) (Marra et al., 2004), [46] and found that only 36.4% of teachers used appropriate environments (pedagogical and didactic) and 63.6% used environments that were not appropriate for the learning process. The Chi2 test of independence is significant between mechanization and integrated development environments ($p=0.015<0.05$).

Some tools used to assess the concepts of algorithmics and coding

Table 12: Tools used for the assessment

What tools are used to assess knowledge?	Frequency	Percentage
Oral questioning	2	5.1%
Written test	6	17.1%
Practical test	5	16.2%
Classwork	3	8.5%
Homework	6	17.9%
Written examination	4	12.8%
Oral examination	0	0.0%
Practical examination	4	12.0%
Physical presence	1	2.6%
Active paricipation	3	7.7%
Total	33	100,0%

Source: Our field surveys

With regard to the instruments used to assess notions of algorithms and coding, 17.9% opted for homework; 17.1% used written questions; 16.2% used practical questions; 12.8% of teachers preferred practical examinations; 12% of teachers used practical examinations; 8.5% used in-class homework; 7.7% assessed using participatory methods; 5.1% of teachers used oral questions; 2.6% opted for physical attendance.

Assessment of knowledge of algorithms and coding

Table 13: Assessment methods

How would you rate learner's knowledge of algorithms and coding?	Frequency	Percentage
By multiple choice questions	2	6.3%
By open questions	10	31.3%
By practical work	19	56.3%
By matching	0	0.0%
By alternative questions	2	6.3%
Total	33	100.0%

Source: Our field surveys

These instruments contain practical work (56.3%); open questions (31.3%); multiple-choice questions (6.3%) and alternative questions (6.3%).

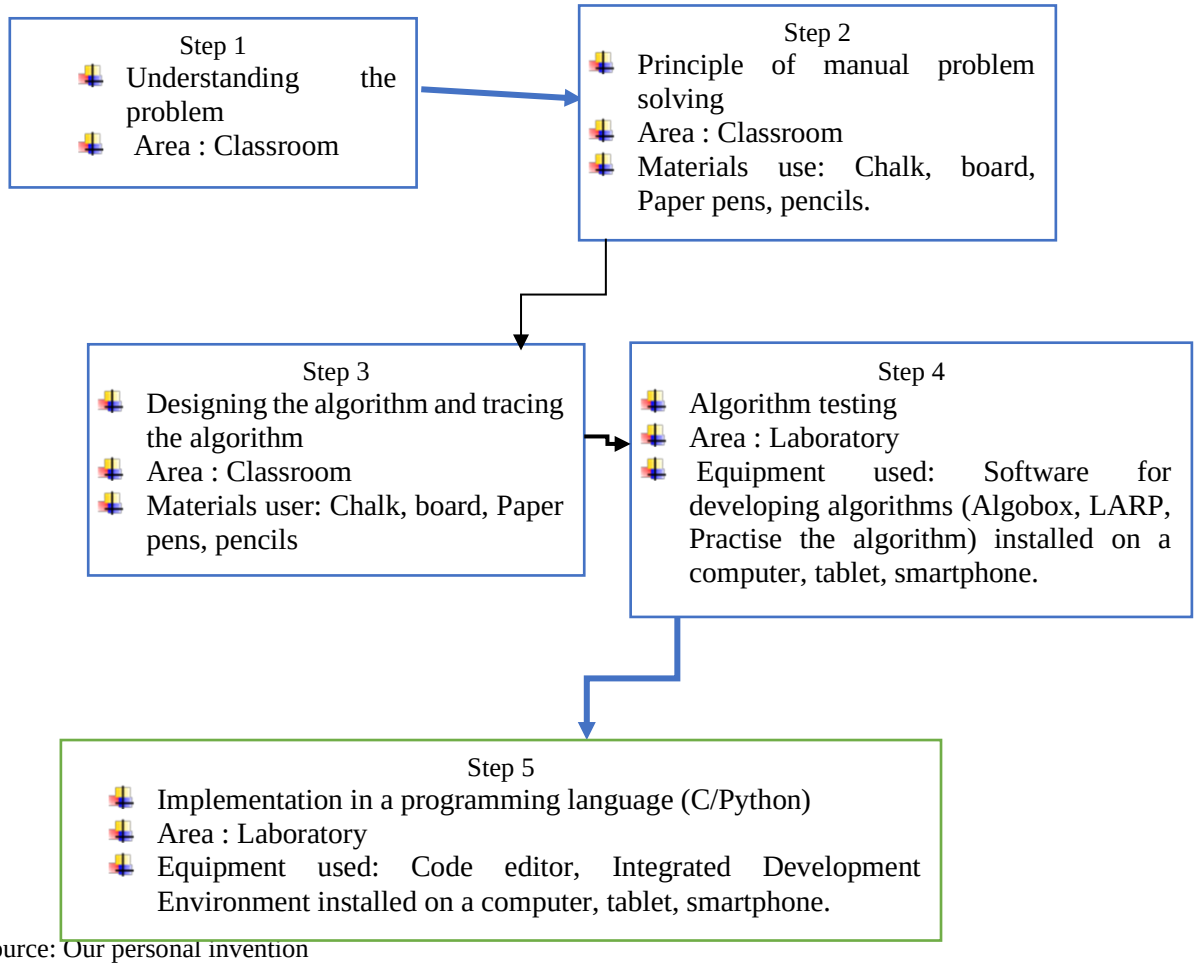
5. DISCUSSION

In line with the socio-economic and demographic characteristics of the teachers (Table 3), we encourage school authorities and managers who are willing to recruit qualified staff to do so, and we also ask for their involvement to obtain the payment of teachers by the State. This is a key factor in the motivation and performance of their work. A study by B. K. BALIMWENGU et al (2023) showed that mechanization had a significant impact on the applicability of the educational program.

According to the teachers, the weekly package allocated to the ICT course is insufficient to teach algorithms and implement them in a programming language. Tables 4 and 5 show that during the 50 minutes allocated by the Ministry, the teacher must do both theory and practice; yet, in terms of pedagogical experience, good theory in the classroom prepares for good practice in the laboratory. When the focus is on doing everything [theory and practice directly] in the laboratory, learners prefer to handle the tool rather than follow the theories that should accompany the practice. However, according to our experience, it is impossible, even hypothetical, to get learners create good algorithms and implement them in 50 minutes, especially as they are still new to programming. Therefore, we suggest that teachers organize fixing sessions outside the scheduled time, in practice, doing the theory in class at the scheduled time and asking for an hour (7th or 8th periods) to code the algorithm produced, without forgetting that it is also recommended to test the algorithm (in view of its complexity) in appropriate software for developing algorithms. We agree with the teachers that the best strategy is to do the theory in the classroom and the practical work in the laboratory.

Proposal of a model for developing and implementing algorithms:

Figure 2: Proposed strategic model for algorithmic design and implementation



Source: Our personal invention

About this model, we suggest to follow the iterations sequentially in Figure 2 to make it easier for learners to learn the algorithms in the following order of occurrence: Understanding the problem → Manual solution principle → Elaboration of the algorithm or Pseudo-code → Implementation in C/Python. For simple algorithms, certain steps can be skipped (e.g. calculating molecular or atomic mass, see MTIC3.8), but for complex or compound algorithms, they need to be tested in software before being implemented in a programming language. As these concepts are complex, a specialist in the field is expected to teach these subjects.

With regard to the justifications concerning the use of algorithmic software, the majority of teachers claim that the use of a teaching tool [algorithmic software] is a factor in its ease of use for learners, which implies that the icons (Marra et al, 2004), menus and commands are clearly identifiable, that the environment is concise and does not present an excessive amount of information or functionality that could distract the learner, that it offers clear and concise feedback to learners (Eriksson, Baykal and Torgersson, 2022), etc.

In the light of all these characteristics, Algobox is right to occupy the top position (cf. Table 8) among the environments used for developing algorithms in the first and second year of science in schools in the town of Bukavu; it is free, open, intuitive, concise, flexible and retroactive. The Ministry of Secondary Education has not designated an environment to be used for developing algorithms, as that has been the case in France (Meyer and Modeste, 2022), (Ovono, Hérold and Ginestié, 2014), (Ponsonnet, 2011), (Ralahady, 2022). We can therefore refute our hypothesis that teachers do not use appropriate environments for developing algorithms.

With regard to the programming languages to be used for coding in the science section, the Ministry plans to implement the algorithms using Python or C. These two languages are therefore the appropriate ones mentioned in our hypotheses for implementing the algorithms in accordance with the Congolese education program. Looking at our results (Table 9), we find that only 33.4% of teachers use the programming languages recommended by the Ministry, even though these are considered useful choices for the current era, such as Python, which is a versatile programming language that can be used for a wide variety of tasks: web and mobile application development, machine learning, artificial intelligence; it is also effective for developing fast and efficient applications. Learners who master the principles of programming using Python will have the necessary background to tackle higher education in both computer science and other sciences, as computer science is one of the cross-curricular subjects.

However, other programming languages are also necessary, but in the Congolese context, where secondary education is concerned, it is vital to follow the Ministry's guidelines. One important announcement corroborates the Ministry's justification for Python, that of Microsoft, which stated in August 2023 that it planned to integrate Python into higher versions of the Office package (Microsoft, 2023), a more reason to be interested in this language.

With regard to the integrated development environments and editors to be used for implementing algorithms in Python or C, we have found that this choice is made according to the wishes of each teacher. However, we suggest choosing simple environments that directly integrate the compiler so that learners do not have to perform many actions to compile or run their programs. For C language implementation, Dev-C++ would be a better choice, since learners can compile and run their programs directly with a single click, rather than using editors such as Sublime Text or Notepad++, which would require additional actions under the command prompt. For Python implementation, although cumbersome, the Anaconda/Spyder environment would be better, since it offers the simplicity of input, execution of Python code and display in the same interface.

In accordance with the assessment tools and instruments used, the teachers use practical tools, which we consider to be very important for the mastery of algorithmic concepts, as they are aimed at the development of the learners' psychomotor skills.

We therefore suggest that decision-makers at the various levels (Division, Coordination, Management Committee) support learning by recruiting specialists in the fields, because this new educational program requires qualified specialists [and/or] graduates in computer science, and that the government and other educational partners support schools with logistical resources (computers, textbooks) so that they can provide quality teaching. The government should also make an effort to include teachers in the payroll, more so those who practice this beautiful profession out of a sense of vocation, which should, in no way, be precluded by financial motivation.

6. CONCLUSION

This study focused on an approach aimed at shedding light on the justifications of environments (teaching tools) used in the teaching of algorithmic and coding concepts by first and second year science teachers in the city of Bukavu. The teachers' opinions were analyzed using SPSS statistical software and Excel spreadsheets. The results showed that the use of these pedagogical and/or didactic tools for developing algorithms depends on their ease of use by learners, their clear and concise feedback to learners. Moreover, the use of programming languages depends firstly on their ease of comprehension by learners, secondly on the fact that they are recommended by the Ministry, and thirdly because they had a large community and offered a variety of resources. However, statistical tests have shown that payment by the State has a positive influence on the use of integrated development environments and/or suitable educational publishers.

So, with regard to the integrated development environments and editors to be used for implementing algorithms in Python or C, we suggest choosing simple environments that directly integrate the compiler so that learners do not have to perform many actions to compile or run their programs; for implementation in Python, although cumbersome, the Anaconda/Spyder environment would be better because it offers the simplicity of input, execution of Python code and display in the same interface, or Dev-C++ for implementation in the C language.

We do not claim to have covered all the concepts involved in teaching algorithms and coding in the science section, so in future research we intend to propose textbooks for teachers and students to provide practical support for the teaching-learning process; and/or propose a platform for exchanging and popularizing contextualized algorithmic elements related to the educational program in use in DR Congo.

REFERENCES

Arsac, J. (1991) ‘Algorithmique et langages de programmation’, *Bulletin de l’EPI (Enseignement Public et Informatique)*, (64), pp. 115–124.

B. K. BALIMWENGU *et al.* (2023) ‘Teaching algorithmic and coding in the first year of science in Bukavu: An exploratory study’, *International Journal of Innovation Scientific Research and Review*, 5(10). Available at: https://hal.science/hal-04264925v1 (Accessed: 8 November 2023).

Beffara, E. *et al.* (2017) *Algorithmique et programmation au cycle 4*. CII Lycée. Available at: https://hal.science/hal-03961063/ (Accessed: 18 October 2023).

Beffara, E. (2023) ‘L’algorithme : pourquoi et comment le définir pour l’enseigner’. Available at: https://hal.science/hal-04112182/ (Accessed: 18 October 2023).

BERKACHY, R. (2017) ‘Nouvelle méthode d’auto-apprentissage du langage de programmation R en utilisant le Notebook Jupyter’. Available at:https://www.unifr.ch/didactic/fr/assets/public/Travaux_fin_etudes/berkachy_diplome.pdf.

Blanvillain, C. (2021) ‘Apprendre à penser les algorithmes’, in *Atelier «Apprendre la Pensée Informatique de la Maternelle à l’Université», dans le cadre de la conférence Environnements Informatiques pour l’Apprentissage Humain (EIAH)*, pp. 1–12. Available at: https://hal.science/hal-03241685/ (Accessed: 18 October 2023).

Chakri, L. and Riouch, M.L. (2021) 'Apports des TIC dans l'enseignement et l'apprentissage des mathématiques : Scénarisation pédagogique et pratiques de l'enseignement à distance', in *ITM Web of Conferences*. EDP Sciences, p. 03012. Available at: https://www.itm-conferences.org/articles/itmconf/abs/2021/04/itmconf_cifem2020_03012/itmconf_cifem2020_03012.html (Accessed: 15 October 2023).
Delahaye, D., Jaume, M. and Prevosto, V. (2005) 'Coq, un outil pour l'enseignement', *Technique et Science Informatiques*, 24(9), pp. 1139–1160. Available at: https://www.researchgate.net/profile/David-Delahaye/publication/237425163_Une_experience_avec_les_etudiants_du_DESS_Developpement_de_logiciels_surs/links/00b4953546ea4bc005000000/Une-experience-avec-les-etudiants-du-DESS-Developpement-de-logiciels-surs.pdf (Accessed: 17 October 2023).
Delannoy, C. (2016) *Programmer en langage C: Cours et exercices corrigés*. Editions Eyrolles. Available at: https://www.fnac.com/a9824782/Claude-Delannoy-Programmer-en-langage-C-5e-edition.
Denis, B. (2020) 'Erasmus+ PIAF, un projet au service de la formation des encadrants d'activités visant développement de la Pensée Informatique et Algorithmique dès 5 ans [séminaire]', *Université de Liège, Université de Lorraine, Universität des Saarlande, Université du Luxembourg*, 81. Available at: https://piaf.loria.fr/wp-content/uploads/2020/02/Pr%C3%A9sentation-du-projet_29-janvier-2020_-Brigitte-Denis.pdf (Accessed: 18 October 2023).
D.M. NTAMBUE and G. M. BAMBANOTA (2018) 'Impact de la qualité de l'enseignement sur les performances des étudiants de l'institut supérieur pédagogique de Bukavu', *International Journal of Innovation and Scientific Research*, 36(1), pp. 79–88.
Eidelman, A. (2020) 'Python Data Science Handbook', *Statistique et Société*, 8(2), pp. 45–47.
EPSP (2019a) 'Guide en appui au Programme Éducatif du Domaine d'Apprentissage des Sciences Classe de deuxième année Scientifique', in. KINSHASA: Direction des Programmes Scolaires et Matériel Didactique. Available at: https://peqpesu.com/component/edocman/mepst-guide-pe4-sptic/viewdocument?Itemid= (Accessed: 12 February 2021).
EPSP (2019b) 'Guide en appui au Programme Éducatif du Domaine d'Apprentissage des Sciences Classe de première année Scientifique', in. KINSHASA: Direction des Programmes Scolaires et Matériel Didactique. Available at: https://peqpesu.com/component/edocman/mepst-guide-pe3-sptic/viewdocument?Itemid= (Accessed: 12 February 2021).
EPST (2018a) *Arrêté ministériel N° MINEPSP/CABMIN/1973/2018 du 26/06/2018 portant validation et généralisation des programmes éducatifs (PE) du Domaine d'Apprentissage des Sciences (DAS) pour la classe de 7e année de l'éducation de base (EB)*. Available at: https://peqpesu.com/component/edocman/arrete-ministeriel-n-minepsp-cabmin-1973-2018-du-26-06-2018-portant-validation-et-generalisation-des-programmes-educatifs-du-domaine-d-apprentissages-des-sciences-pour-la-classe-de-7e-annee/viewdocument?Itemid= (Accessed: 11 February 2022).
EPST (2018b) 'Guide en appui au Programme Éducatif du Domaine d'Apprentissage des Sciences Classe de 8ème année de Éducation de Base', in. KINSHASA: Direction des Programmes Scolaires et Matériel Didactique. Available at: https://peqpesu.com/component/edocman/sciences-physiques-technologie-et-technologies-de-l-information-et-de-la-communication/viewdocument?Itemid= (Accessed: 11 September 2021).
EPST (2019a) *Arrêté ministériel N° MINEPSP/CABMIN/599/2019 du 03/07/2019 portant validation et généralisation des programmes éducatifs du DAS pour la classe de 8e année de l'EB*. Available at: https://peqpesu.com/component/edocman/arrete-ministeriel-n-minepsp-cabmin-599-2019-du-03-07-2019-portant-validation-et-generalisation-des-programmes-educatifs-du-domaine-d-apprentissage-des-sciences-pour-la-classe-de-8e-annee-de-l-education-de-base/viewdocument?Itemid= (Accessed: 2 February 2022).
EPST (2019b) *Arrêté N° MINEPSP/CABMIN 600/2019 du 03/07/2019 portant mise en place de la section unique des humanités scientifiques*. Available at: https://peqpesu.com/component/edocman/arrete-generalisation-1e-hsc/viewdocument?Itemid= (Accessed: 1 February 2022).
Eriksson, E., Baykal, G.E. and Torgersson, O. (2022) 'The Role of Learning Theory in Child-Computer Interaction - A Semi-Systematic Literature Review', in *Interaction Design and Children*. *IDC '22: Interaction Design and Children*, Braga Portugal: ACM, pp. 50–68. Available at: https://doi.org/10.1145/3501712.3529728.
Faouzia, B. and Mostafa, H. (2007) 'Utilisation des NTICs pour l'apprentissage et l'autoévaluation de l'algorithmique', in *4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, pp. 25–29. Available at: https://www.researchgate.net/profile/Mostafa-Hanoune/publication/268003435_Utilisation_des_NTICs_pour_l%27apprentissage_et_l%27autoevaluation_de_l%27algorithmique/links/5819ecfa08aeffb294130644/Utilisation-des-NTICs-pour-lapprentissage-et-lautoevaluation-de-lalgorithmique.pdf (Accessed: 15 October 2023).
Ilundu, W.W. (2019) *Impact de la formation continue des enseignants des classes de premières années secondaires de la ville de Bukavu sur l'enseignement de la structure interne et de la maintenance des ordinateurs*. Université

Pédagogique Nationale de Kinshasa. Available at: https://theses.hal.science/tel-02618723/ (Accessed: 4 March 2023).
Jeanjean, P. (2022) 'IDE as Code: reifying language...' Available at: https://scholar.google.fr/scholar?hl=fr&as_sdt=0%2C5&q=Jeanjean%2C+P.+%282022%29.+IDE+as+Code%3A+reifying+language+protocols+as+first-class+citizens.+https%3A%2F%2Fwww.theses.fr%2F2022REN1S033+&btnG= (Accessed: 19 October 2023).
Karsenti, T. (2020) 'raisons d'apprendre à coder à l'école', *Éducation Canada, Mai* [Preprint]. Available at: https://www.desjardins.com/ressources/pdf/d25-12-raisons-apprendre-coder-ecole-f.pdf?resVer=1568637378000 (Accessed: 1 February 2023).
Lagrange, J.-B. and Rogalski, J. (2017) 'Savoirs, concepts et situations dans les premiers apprentissages en programmation et en algorithmique', *Annales de Didactique et de Sciences Cognitives. Revue internationale de didactique des mathématiques*, (22), pp. 119–158.
Marquet, P. (2022) 'Analyse de l'impact de la mise en œeuvre de scénarios pédagogiques et d'outils d'étayage visant le développement de la pensée informatique et algorithmique dans l'enseignement fondamental'. Available at: https://matheo.uliege.be/handle/2268.2/16157 (Accessed: 15 October 2023).
Marra, R.M. *et al.* (2004) 'Validating the technology learning cycle in the context of faculty adoption of integrated uses of technology in a teacher education curriculum', *Int. J. Learn. Technol.*, 1(1), pp. 63–83.
Meyer, A. and Modeste, S. (2022) 'Rôle d'un logiciel dans la transposition didactique du concept d'algorithme: le cas du logiciel AlgoBox en France et des programmes du lycée entre 2009 et 2019', *Cahiers d'histoire du Cnam*, 15(1), p. pp-97.
Microsoft (2023) *Combining the power of Python and the flexibility of Excel.*, *TECHCOMMUNITY.MICROSOFT.COM*. Available at: https://techcommunity.microsoft.com/t5/excel-blog/announcing-python-in-excel-combining-the-power-of-python-and-the/bc-p/3910459 (Accessed: 19 October 2023).
Milondo, A.M. (2016) *Méthodes quantitatives et recherche scientifique en sciences sociales: Aspects théoriques et méthodologiques sur le traitement des données*. Éditions universitaires européennes. Available at: https://books.google.cd/books/about/M%C3%A9thodes_quantitatives_et_recherche_sci.html?id=HzH5jwEACAAJ&redir_esc=y (Accessed: 2 April 2023).
Modeste, S. (2012) *Enseigner l'algorithme pour quoi? Quelles nouvelles questions pour les mathématiques? Quels apports pour l'apprentissage de la preuve?* PhD Thesis. Université de Grenoble. Available at: https://theses.hal.science/tel-00783294/ (Accessed: 18 October 2023).
Nathan (2016) *Transmath Cycle 4 (2016)*. Available at: https://transmath-college.nathan.fr/transmath-c4 (Accessed: 18 October 2023).
Nguyen, C.T. (2005) *Etude didactique de l'introduction d'éléments d'algorithmique et de programmation dans l'enseignement mathématique secondaire à l'aide de la calculatrice*. PhD Thesis. Université Joseph-Fourier-Grenoble I. Available at: https://theses.hal.science/tel-00011500/ (Accessed: 15 October 2023).
Ovono, M.-S., Hérold, J.-F. and Ginestié, J. (2014) 'Introduction d'un logiciel de simulation d'algorithmes dans le processus enseignement apprentissage de l'algorithmique chez les apprenants débutants de l'ENSET de Libreville', in *4° Colloque du RAIFFET*. Available at: https://hal.science/hal-01780925/ (Accessed: 18 October 2023).
Ponsonnet, L. (2011) *Initiation à l'algorithmique et à la programmation pour le lycée : programmation sur calculatrices Texas Instruments et Casio, programmation avec les logiciels Algobox et Xcas*. Ellipses.
Ralahady, B.B. (2022) *Contribution à l'étude de l'Éducation algorithmique chez les apprenants du secondaire à l'aide d'analyses statistiques implicatives selon MGK*. PhD Thesis. Université d'Antananarivo (Madagascar). Available at: https://hal.science/tel-03698075/ (Accessed: 18 October 2023).
Vandeputa, É. and Henryb, J. (2018) 'Apprendre à programmer Comment les enseignants justifient-ils le choix d'un outil didactique ?', *De 0 à 1 ou l'heure de l'informatique à l'école*, p. 325.
ZEGOUR. D. E. (2020) *Apprendre et enseigner l'algorithmique*. Available at: www.software-partners.co.uk (Accessed: 18 October 2023).