

Automated Assessment with Multiple-choice Questions using Weighted Answers

Francisco de Assis Zampiroli ^{a,*}, Valério Ramos Batista ^b, Carla Rodriguez ^c,
Rafaela Vilela da Rocha ^d and Denise Goya ^e

*Centro de Matemática, Computação e Cognição, Universidade Federal do ABC (UFABC),
09210-580, Santo André, SP, Brazil*

Keywords: Automated Assessment, Multiple Choice Questions, Parametrized Quizzes.


Abstract: A resource that has been used increasingly in order to assess people is the evaluation through multiple-choice questions. However, in many cases some test alternatives are wrong just because of a detail and scoring nought for them can be counter-pedagogical. Because of that, we propose an adaptation of the open-source system MCTest, which considers weighted test alternatives. The automatic correction is carried out by a spreadsheet that stores the students' responses and compares them with the individual answer keys of the corresponding test issues. Applicable to exams either in hardcopy or online, this study was validated to a total of 607 students from three different courses: Networks & Communications, Nature of Information, and Compilers.


1 INTRODUCTION


In the case of multiple-choice questions, it is expected that teachers and professors engage in extra endeavour to elaborate ones that fairly assess their students' competencies and skills. There are widely accepted methods to evaluate and classify a large number of candidates, for instance the Item Response Theory (Aybek and Demirtasli, 2017). As an example, let us consider the Brazilian National High School Exam (ENEM), elaborated by Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). In January 2021 ENEM had almost 5.8 million students enrolled for the classroom tests, but the absence rate was 55.3% mostly due to the Coronavirus pandemic. The reader can see enem.inep.gov.br for details, but here we highlight that for the first time applicants could sit this exam online in some venues. This was possible for 93,079 of the candidates but precisely they contributed 70% to absence. Anyway, INEP foresees that 100% of the tests will be online al-


ready in 2026. In fact, it is following the same trend of many others, e.g. the TOEFL language exam, which is now online (ets.org), and such a trend boosts more sophisticated studies devoted to the elaboration of applicable questions.


In (Burton, 2001) the author presents a study on improvements for the reliability of multiple-choice questions through deterring examinees from just guessing the right answer. The paper states that pure guessing can be discouraged by fractional marks attributed to wrong answers, namely 'negative marking' or 'penalty scoring'. However, the final performance can be damaged by the examinee's uncertainty in case they have solved a question just partially. The very author cites some works that debate such penalties, but he focuses on achieving percentage values of unreliability of a test by studying three scenarios: Q, where the only random element is the drawing of some items from Question Banks (QB) in which scope and difficulty are equally levelled, and the final mark is exactly the number of correct answers; G, in which the only random element is the drawing of the alternatives; QG, which uses both random elements. All questions must be answered. For Q and QG one must have QB with at least five times the number of questions in the exam. In his model, the author considers an exam with sixty questions and four alternatives per question. By taking the average knowledge

^a  <https://orcid.org/0000-0002-7707-1793>

^b  <https://orcid.org/0000-0002-8761-2450>

^c  <https://orcid.org/0000-0002-1522-3130>

^d  <https://orcid.org/0000-0003-4573-3016>

^e  <https://orcid.org/0000-0003-0852-6456>

*Grant #2018/23561-1, São Paulo Research Foundation (FAPESP).

of 50% the mean scores in cases Q, G and QG were 30, 37.5 and 37.5, respectively. He concludes that a 60-question four-choice test is rather unreliable, and one of the main reasons is that G and QG allow guessing, which is not the case of Q.

The authors in (Oliveira Neto and Nascimento, 2012) adapted the Learning Management System (LMS) Moodle to make formative assessment during the teaching-learning process with high quality feedback for a distance learning course of 40h per week in Mathematical Finance. These evaluations can better direct the student's performance if the feedback is quick and precise at pointing out their difficulties. Moreover, the feedback can guide the teacher about the adopted teaching process, and so the students' understanding can be reinforced regarding some topics that have not been assimilated yet. By analysing the students' answers in previous classes, the authors have improved the QB with additional rules to tests, error messages and links to either theoretical topics or extra exercises.

In the elaboration of multiple-choice questions, it is also important to consider suitable wrong options among the alternatives, also called *distractors*. Unsuitable distractors enable the examinee to guess the correct answer by discard, as discussed in (Moser et al., 2012), where the authors present a text processing algorithm for automatic selection of distractors. A more recent work is (Susanti et al., 2018), but devoted to automatic production of distractors for the English vocabulary. In (Ali and Ruit, 2015) the authors present an empirical study on flawed alternatives and low distractor functioning. They conclude that removal or replacement of such defective distractors, together with increasing the cognitive level, improve detection of high- and low-ability examinees.

Our present work introduces an automatic generator and corrector devoted to exams that consist of multiple-choice questions with weighted alternatives. It is adapted from the open-source system MCTest available on GitHub. For such exams MCTest stores the correction in a CSV-file and emails it to the professor. This file contains each student's responses compared with the individual answer key of the exam issue received by that student. Common programs like Excel and LibreOffice open the file in a spreadsheet with built-in formulas that give each student's final mark according to the weights, as we shall detail in this paper.

As a related work we cite (Presedo et al., 2015), in which the authors use Moodle to create multiple-choice questions with weighted alternatives. Their system also enables the user to give an exam in hard-copy but with neither the student's id nor variations

of the exam. Moreover, it requires the plugin *Offline Quiz* (moodle.org/plugins/mod_offlinequiz). Moodle enables *Calculated question type*, that we call parametric question in MCTest, in which the statement and the alternatives accept *wildcards* but in Moodle only for simple mathematical operations. By contrast, MCTest enables nominal exams, numerous variations and wildcards that accept complex formulas written in Python and its libraries. Details on parametric questions with MCTest can be found in (Zampirolli et al., 2021; Zampirolli et al., 2020; Zampirolli et al., 2019).

The paper is organized as follows. Section 2 describes the adapted MCTest for an automated assessment with multiple-choice questions using weighted answers; Section 3 shows the obtained results and discusses them; finally, Section 4 presents our main conclusions and opportunities for future work.

2 USING ADAPTED MCTest: MATERIALS AND STEPS

This work applies the open-source Information and Communication Technology (ICT) MCTest available on GitHub (<https://github.com/fzampirolli/mctest>). We have implemented MCTest in order to enable weighting answers of multiple-choice questions. In this section, we explain how to create exams that include such questions with weighted answers.

2.1 Creating Multiple-choice Questions

After downloading MCTest from GitHub, the system administrator must install it on a server. Before creating a question, they have to include Institution, Course, Discipline and also associate a professor as Discipline Coordinator. This one can create discipline Topics and also add more professors. See vision.ufabc.edu.br for details. Afterwards, any of them can add a Class and also questions associated to a Topic thereof. An example would be setting [ED]<template-figure> at Choose Topic in Figure 1. Namely, this topic belongs to a discipline called ED, a mnemonic to Example Discipline. In that figure we have Short Description: template-fig-tiger-en, which is optional but makes it easier to locate questions in Question Banks (QB), as we shall explain in Subsection 2.2. The field Group is also optional for the user to define a group of questions, so that in each exam MCTest will always draw only one question from that group. The most relevant field is Description, where we can insert paragraphs in \LaTeX and also combine them with a Python code, as explained later in another example.

Question Update

Create-PDF	Compile-Colab	Save-Json
See this question in PDF format	Copy-Paste the description of question for test in Colab Google	It will save all your questions to a file in json format

Choose Topic [Ex]<template-figure>

Short Description TEMPLATE-01: FIGURE-en

Group group 1

Description

What is the best answer for the animal name that appears in the following figure:

```
\begin{figure}[h!]\centering
\write18{wget -c https://cdn.pixabay.com/photo/2017/02/04/22/37/panther-2038656_960_720.png -O ./tmp/fz_pi_fig01.png}
\fbbox{\includegraphics[scale=0.25]{fz_pi_fig01}}
\end{figure}
```

Type Multiple-Choice Question

Difficult Very easy level question

Bloom Taxonomy remember: recognizing, recalling

Parametric No

Who Created fzampirolli@ufabc.edu.br

Last Update 2021-02-11

Figure 1: Cutout of the MCTest subpage to create/update a question. In this first step we describe the statement and optionally some parameters.

Notice that in Figure 1 we are invoking a graphic, namely `mctest_fig01.png` available at `shorturl.at/opFM5`. This image was stored in a server where MCTest is installed, and we use the path `./tmp/mctest_fig01.png` to retrieve it with the following command:

```
\write18{wget -c shorturl.at/opFM5 -O
./tmp/mctest_fig01.png}
```

The \LaTeX compiler retrieves this image and includes it in the question.

In Figure 1 one sees `Type: Multiple-Choice Question (QM)`, which could be toggled to `Dissertation Question (QT)` in case of written response, for instance if the student must write a program code. In the field `Difficult` the user must choose one among five levels of difficulty to that question, whereas `Bloom Taxonomy` is optional, but has six levels (Krathwohl et al., 1956; Anderson and Krathwohl, 2001). As default the field `Parametric` is toggled to `No`, but if the user changes it to `Yes` they will be able to define wildcards to which MCTest can attribute values either at random or as a result of a mathematical operation (Zampirolli et al., 2019). Finally, we have the self-explanatory fields `Who`

`Created` and `Last Update`, which are optional.

Figure 2 complements MCTest's subpage of updating a question. There we see the fields of answers and optionally feedback. New answers can be included right after filling out the last field `Answer Text` and clicking on `Submit`. Already existing answers can be marked with `Delete`, and they will be discarded with `Submit`. MCTest draws a random order for the alternatives each time it produces an issue. See the red and blue numbers after the symbol # in Figure 3. But in Figure 2 the user must always put the correct answer in the first field.

The answers in Figure 2 can be weighted according to what the professor considers from 0% to 100% correct. This process will be detailed in Section 3, but here we already mention that values outside the interval $[0, 100]$ are possible, for instance negative weights that penalize wrong answers. However, we only know this choice in case of competitions for a job opportunity, a prize, etc. Since teaching is the main objective of an educational institution, then we have never applied negative weights, as corroborated in (Burton, 2001).

Answer Text:

Answer Feedback:

Delete:

Answer Text:

Answer Feedback:

Delete:

Answer Text:

Answer Feedback:

Delete:

Answer Text:

Answer Feedback:

Delete:

Answer Text:

Answer Feedback:

Delete:

Figure 2: Cutout of the MCTest subpage to create/update a question. In this second step the user writes each answer and decides on its feedback.

MCTest

Topic: template-figure
 Group: group 1
 Short Description: TEMPLATE-01: FIGURE-en
 Type: QM
 Difficulty: 1
 Bloom taxonomy: remember
 Last update: 2021-02-11
 Who created: fzampirolli@ufabc.edu.br

#071 1. What is the best answer for the animal name that appears in the following figure:



A.->Tiger B.->Jaguar C.#0Panther D.->Cat

Figure 3: Cutout of MCTest to visualize the layout of a question after clicking on Create-PDF (see Figure 1). The green number is the question ID in the QB, and here the right answer is C (located with #0). This and the red numbers give the order in which MCTest drew the answers. Invoking again Create-PDF will result in another random order of the answers.

2.2 Creating Exams

Once the user has made or updated QB according to the steps explained in Subsection 2.1, next they can create an exam by choosing classes and questions. Figure 4 shows MCTest's subpage Exam for

this purpose. There one must fill out Name of the exam and Choose Classrooms to apply it. In order to choose questions the user can resort to a searching interface. As an example, in the field Search: we included the token “-en” to see all the questions that have it in their description. Here the field Par. was toggled to sort them by showing first the non-parametric questions. We also have the option to create PDF, similarly to our explanation of Figure 1, but here MCTest will show the whole exam as depicted in Figure 5. In this figure one sees other questions drawn by MCTest and the third one is parametric, described as `template-sum matrix-en`. The correct answer is computed by a Python code that goes together with the \LaTeX statement, but located between `[[code: and]]` for MCTest to compile them correctly. The right answer is computed with `[[code:0+1+2+3+1+2+3+4+2+3+4+5]]`. See vision.ufabc.edu.br for details about combining \LaTeX and Python for parametric questions and for creating exams.

The professor gets a PDF, like in Figure 5, and can also print it if approved. As a matter of fact, the PDF has many pages that reproduce what we show in Figure 5, but each page has a different order of the questions and of their respective answers. Of course, these are the different issues, one for each student, who will have to fill out the answer card of the exam header shown in Figure 5. Afterwards, the professor must digitize all exam first pages into a single PDF and send it to MCTest by clicking on Upload-PDF of the Exam subpage. Then MCTest will email a CSV-file to the professor. This file contains the correction of each exam, as we shall explain in Section 3.

If students are allowed to sit the exam online, then instead of printing the PDF the professor can email it to them. Of course, each student will receive only their corresponding issue, not the whole PDF. In this case each one must fill out an online form, and the automatic correction will happen similarly, as we shall explain in Section 3. Also, in this case, MCTest emails the aforementioned CSV-file to the professor.

3 EXPERIENCE REPORT: METHOD, RESULTS AND DISCUSSION

MCTest has been used by professors at our institution since 2011. Earlier versions of MCTest did not work with weighted alternatives, which were first incorporated by the system in 2017. In this paper, we present an experience report of a professor that started using

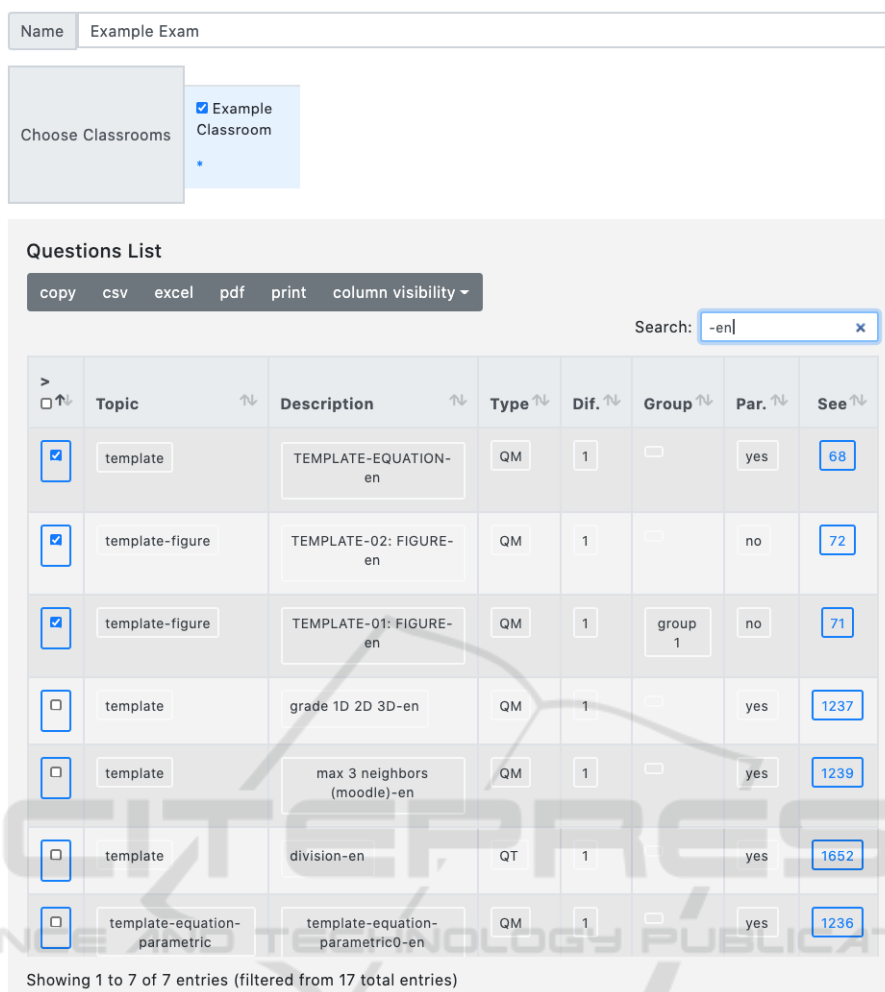


Figure 4: Cutout of the MCTest subpage to create/update an exam. In this first step one chooses name, classes and questions.

MCTest in 2016 and motivated us to implement this option in our system.

This professor started lecturing Computer Science at our university in 2012. Until 2016 the largest classes consisted of five to six dozens of students, and at the time he was responsible for two of such classes in the course Networks & Communications, the one in the morning and the other in the evening. He was consulted about applying MCTest to his classes, so that the correction of the exams could be carried out automatically, except for the written response part, called QT in MCTest. This would reduce the correction time, but MCTest needed QB for that course, so he would have to elaborate multiple-choice questions, called QM. The recommended format was ten QM, one long question of QT, each part devoted to half of the total mark, and this he called model 1.

Of course, it is much simpler to prepare an exam with four to five written response questions, which he called model 2, but for circa 60 to 70 students the

correction time is considerable. Since both options would make him spend an equivalent time, he decided to let the students vote the model for the first exam. However, in order to discourage guessing alternatives, in both models questions were to be of middle to high difficulty levels, but students could use a handwritten summary on a single paper in model 2.

Surprisingly, morning and evening classes voted for different models, 2 and 1, respectively. But these results were repeated in the subsequent votings for the second exam, and also in another course he lectured both in 2016 and 2017: Nature of Information. Classes were then growing to circa 80 students and model 1 was becoming his preference, specially because one can easily change and increase already existing QB. By asking the students for their reasons to reject either model, the evening classes claimed they did not have time to prepare a handwritten summary, and the morning classes complained that QM is unfair because a partially correct answer deserves a frac-

	Example Institute Example Course Discipline: Example Discipline Classroom: Example Classroom Exam: Example Exam	Room: sa Date: 11-02-2021	
	Sig.: _____ Student: Student Two ID/RA: 2		

#135 - 2021-02-11 - 20:13:58

	1	2	3
A	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
D	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Instructions:

- (a) Turn off the cell phone;
- (b) Take this exam necessarily at this link <https://forms.gle/L8bdaTgStGs9WUPW6>.

Multiple Choice Questions:

1. Consider the Cantor set C whose construction is depicted in figure. About $\#C$ we can say that:



 - A. $\#N < \#C < \#R$
 - B. $\#C = \#N$
 - C. $\#C = \#R$
 - D. It's impossible to determine $\#C$
2. What is the result of the equation $\sqrt{3^2 + 5^2}$
 - A. 5.830951894845301
 - B. 9.8309518948453
 - C. 10.8309518948453
 - D. 4.830951894845301
3. What is the best answer for the animal name that appears in the following figure:



 - A. Jaguar
 - B. Tiger
 - C. Panther
 - D. Cat

Figure 5: Cutout of the MCTest subpage to create/update an exam. In this second step the user checks whether the PDF meets their expectations.

tional score. He then decided to propose model 1 with 4 alternatives per question and the following weights: 100% (the correct one), 0% (the absurd one), and 20% (either of the partially correct ones). This because if someone made “shots in the dark” when answering the QM the expected score would still remain quite low, namely 3.5 (compared with 2.5 if non-weighted). This improvement in model 1 made it acceptable for

the morning classes, so that he started applying it even for other courses, e.g. Compilers in 2019.

As commented in Subsection 2.2, MCTest stores the correction in a CSV-file and emails it to the professor, and Figure 6 shows an example thereof. Column A gives the PDF page of the scanned filled-out answer card, B is the exam issue, C is the number of alternatives per question (MCTest allows only the

Page	ID	Answer	Question	Fail	Grade	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	2	4	10	0	8	D/C	D	A	A	A	C	D	A	C/A	A	11261203	11303210	11330213	11310132	11240231	11323102	11283120	11290321	11250231	11270312
2	5	4	10	1	9	C	B	D	B	B	B	A	A	0/B	D	11293102	11322013	11311320	11271023	11282031	11253012	11240321	11330213	11263021	11303210
3	4	4	10	0	8	C/D	B	C	C	B/D	C	C	A	C	C	11251320	11322013	11291203	11273102	11301320	11331302	11261302	11310321	11242301	11281203
4	1	4	10	0	3	A/C	D	C/D	C/B	C/B	C	A	C/B	A/D	B/D	11322301	11282130	11251320	11331023	11241032	11263102	11300213	11273012	11293120	11311230
5	2	4	10	0	10	C	D	A	A	A	C	D	A	A	A	11261320	11303210	11330213	11310132	11240231	11323102	11283120	11290321	11250231	11270312
6	4	4	10	0	9	D	B	C	C	D	B/C	C	A	C	C	11251320	11322013	11291203	11273102	11301320	11331302	11261302	11310321	11242301	11281203
7	1	4	10	0	10	C	D	D	B	B	C	A	B	D	D	11322301	11282130	11251320	11331023	11241032	11263102	11300213	11273012	11293120	11311230
8	2	4	10	0	10	C	D	A	A	A	C	D	A	A	A	11261203	11303210	11330213	11310132	11240231	11323102	11283120	11290321	11250231	11270312
9	5	4	10	0	8	D/C	B	A/D	B	B	B	A	A	B	D	11293102	11322013	11311320	11271023	11282031	11253012	11240321	11330213	11263021	11303210
10	3	4	10	0	9	C	C	D	B	B	D	A	D	B/A	C	11323201	11253201	11263120	11243021	11283012	11272310	11310321	11301320	11290321	11332301
11	4	4	10	2	8	D	B	0/C	C	D	C	C	A	0/C	C	11251320	11322013	11291203	11273102	11301320	11331302	11261302	11310321	11242301	11281203
12	3	4	10	0	9	C	C	D	B	B	D	C/A	D	A	C	11323201	11253201	11263120	11243021	11283012	11272310	11310321	11301320	11290321	11332301
13	3	4	10	0	9	C	C	D	B	B	D	C/A	D	A	C	11323201	11253201	11263120	11243021	11283012	11272310	11310321	11301320	11290321	11332301

Figure 6: LibreOffice visualization of the CSV-file sent by MCTest.

	A	B	C	D
	0	1	2	3
Weight	1	0,2	0,2	0

Grade	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	
8	0	1	1	1	1	1	1	1	1	0	1	D/C	D	A	A	A	C	D	A	C/A	A	1203	3210	0213	0132	0231	3102	3120	0321	0231	0312
9	1	1	1	1	1	1	1	1	1	0	1	C	B	D	B	B	B	A	A	0/B	D	3102	2013	1320	1023	2031	3012	0321	0213	3021	3210
8,2	0,2	1	1	1	0	1	1	1	1	1	1	C/D	B	C	C	B/D	C	C	A	C	C	1320	2013	1203	3102	1320	1302	1302	0321	2301	1203
4	0,2	1	0,2	0,2	0	1	1	0,2	0	0,2	A/C	D	C/D	C/B	C/B	C	A	C/B	A/D	B/D	2301	2130	1320	1023	1032	3102	0213	3012	3120	1230	
10	1	1	1	1	1	1	1	1	1	1	1	C	D	A	A	A	C	D	A	A	A	1203	3210	0213	0132	0231	3102	3120	0321	0231	0312
9	1	1	1	1	1	0	1	1	1	1	1	D	B	C	C	D	B/C	C	A	C	C	1320	2013	1203	3102	1320	1302	1302	0321	2301	1203
10	1	1	1	1	1	1	1	1	1	1	1	C	D	D	B	B	C	A	B	D	D	2301	2130	1320	1023	1032	3102	0213	3012	3120	1230
10	1	1	1	1	1	1	1	1	1	1	1	C	D	A	A	A	C	D	A	A	A	1203	3210	0213	0132	0231	3102	3120	0321	0231	0312
8,4	0,2	1	0,2	1	1	1	1	1	1	1	1	D/C	B	A/D	B	B	B	A	A	B	D	3102	2013	1320	1023	2031	3012	0321	0213	3021	3210
9	1	1	1	1	1	1	1	1	1	0	1	C	C	D	B	B	D	A	D	B/A	C	3201	3201	3120	3021	3012	2310	0321	1320	0321	2301
8	1	1	0	1	1	1	1	1	0	1	0	D	B	0/C	C	D	C	C	A	0/C	C	1320	2013	1203	3102	1320	1302	1302	0321	2301	1203
9,2	1	1	1	1	1	1	1	0,2	1	1	1	C	C	D	B	B	D	C/A	D	A	C	3201	3201	3120	3021	3012	2310	0321	1320	0321	2301
9,2	1	1	1	1	1	1	1	1	0,2	1	1	C	C	D	B	B	D	A	D	D/A	C	3201	3201	3120	3021	3012	2310	0321	1320	0321	2301

Figure 7: Spreadsheet with weighted alternatives derived from MCTest’s CSV-file in Figure 6.

same number for the whole QM), D is the QM size, E shows the number of corrupted questions (0 = non-corrupted, i.e. the student chose one item per question), and F the non-weighted score. Columns G to P show the answers in red and also the correct one in case of mismatch (“chosen”/“key”).

Columns Q to Z in green show information on both the question ID in the QB and the order in which alternatives were drawn for that student. In this example, we had four alternatives per question, hence the last four digits show their order. For instance, cell Q2 shows 11261203, where 1126 stands for the question ID and 1203 means the drawn order. By looking at Figure 2 this order indicates that A=1, B=2, C=0, D=3 are the 2nd, 3rd, 1st and 4th alternative, respectively.

Hence one can add a new tab to the spreadsheet, and with simple formulas compute the final mark by attributing weights to the alternatives, as depicted in Figure 7. In this example the right alternative weighs 1, the partially-right ones weigh 0.2 each, and the absurd one weighs 0. The yellow field in Figure 7 shows the score of each question, where any row corresponds to a single student’s answer card.

According to the professor’s account, simple solutions as this one proposed in our work increase the students’ acceptance of QM. For the users not very familiar with spreadsheet formulas, we have left the template of Figure 7 at: <http://vision.ufabc.edu.br/static/GRADE.ods>

Anyway, it is important to consider what we briefly exposed at the Introduction. For this purpose, the professor conceded one of his questions:

In the course we have seen that a graph $G = (V, E)$ can be planar though one needs to redraw it, as in Figure 8(a), which is equivalent to a regular pentagon with vertices $1 - \dots - 5 - 1$. By considering G in Figure 8(b) and $S = V - E + F$, where F is the number of faces, we have:

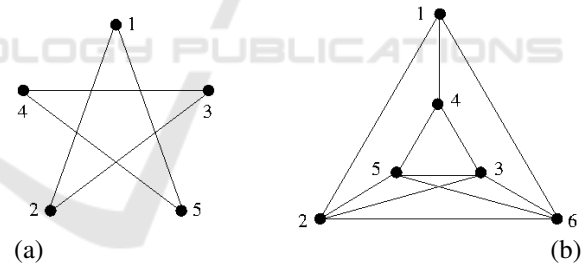


Figure 8: A question with weighted alternatives used in the experience report.

These were the alternatives (but the order varied by issue): A. G is planar because $S = 2 * 3$; B. $S = 2$ but G is planar not for this reason $* 2$; C. G is not planar because $S = 3 \neq 0$; D. G is not planar although $S = 2 * 1$.

4 CONCLUSIONS AND FUTURE WORKS

We have just presented a system and experience report with 607 students in three courses between 2017 and 2019, based on the open-source MCTest available on

GitHub. With MCTest the professor can prepare an exam either in hardcopy or online, and this second modality is useful in case the class cannot gather in a classroom. Here each student receives their PDF issue by email and sends the answers through Google Form. In either modality the professor gets all non-weighted corrections in a spreadsheet, which can be adapted to compute marks by weighting the answers, as explained in Section 3.

In future works we are going to investigate the students' acceptance of weighted QM by considering more participants and questionnaires. Our purpose is to find out which styles of exam can simplify the professor's work, be acceptable by the classes and, at the same time, result in a fair assessment of the students' learning. To the best of our knowledge the literature still lacks such a study.

REFERENCES

- Ali, S. and Ruit, K. (2015). The impact of item flaws, testing at low cognitive level, and low distractor functioning on multiple-choice question quality. *Perspectives on medical education*, 4(5):244–251.
- Anderson, L. and Krathwohl, D. (2001). A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives.
- Aybek, E. and Demirtasli, R. (2017). Computerized adaptive test (cat) applications and item response theory models for polytomous items. *International Journal of Research in Education and Science*, 3:475–487.
- Burton, R. (2001). Quantifying the effects of chance in multiple choice and true/false tests: question selection and guessing of answers. *Assessment & Evaluation in Higher Education*, 26(1):41–50.
- Krathwohl, D., Bloom, B., and Masia, B. (1956). *Taxonomy of educational objectives: The classification of educational goals*. David McKay Company, Incorporated.
- Moser, J., Gütl, C., and Liu, W. (2012). Refined distractor generation with lsa and stylometry for automated multiple choice question generation. In *Australasian Joint Conference on Artificial Intelligence*, pages 95–106. Springer.
- Oliveira Neto, J. and Nascimento, E. (2012). Intelligent tutoring system for distance education. *JISTEM-Journal of Information Systems and Technology Management*, 9(1):109–122.
- Presedo, C., Armendariz, A., López-Cuadrado, J., and Pérez, T. (2015). Calibración de ítems vía expertos utilizando moodle. *Revista Ibero-americana de Educação*, 69(1):117–132.
- Susanti, Y., Tokunaga, T., Nishikawa, H., and Obari, H. (2018). Automatic distractor generation for multiple-choice english vocabulary questions. *Research and Practice in Technology Enhanced Learning*, 13(1):15.
- Zampirolli, F., Borovina Josko, J., Venero, M., Kobayashi, G., Fraga, F., Goya, D., and Savegnago, H. (2021). An experience of automated assessment in a large-scale introduction programming course. *Computer Applications in Engineering Education*.
- Zampirolli, F., Pisani, P., Borovina Josko, J., Venero, M., Kobayashi, G., Fraga, F., Goya, D., and Savegnago, H. (2020). Parameterized and automated assessment on an introductory programming course. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1573–1582. SBC.
- Zampirolli, F., Teubl, F., and Batista, V. (2019). Online generator and corrector of parametric questions in hard copy useful for the elaboration of thousands of individualized exams. In *CSEdu*, pages 352–359.