

Automated Randomization of Test Problems for Cheating Prevention

Chang J. Lee

Abstract—Use of mobile devices in classes is no longer an unusual practice these days. In an offline college chemistry class I taught recently, regular textbooks and pocket calculators or computer labs were forgone in favor of mobile devices and apps available in app stores. My role as an instructor was to guide the students to use the apps on mobile devices to solve problems relevant to chemistry. Students were encouraged to discuss and help each other in utilizing the apps to solve chemistry problems, and it was certain that they were highly engaged in learning with this approach. However, a downside of using mobile devices is that it can make cheating much easier. In the spirit of the class students were allowed to use them in examinations to solve problems, and it was found that cheating was more widespread among students than in traditional pen and paper examinations. In this paper I discuss a system developed with web technologies for mitigating student cheating on tests and report the outcome of implementation of the system.

Index Terms—Anti-cheating, Online Exam, M-Learning.

I. INTRODUCTION

The percentage of smartphone users is steadily growing worldwide [1]. Especially, the smartphone penetration among young adults is currently about 90% in the countries surveyed, according to references cited in a recent article [2]. A 2015 survey indicates that nine in ten college students use a laptop or smartphone on a regular basis, and half regularly use a tablet [3]. For school work, 87% college students use a laptop or other portable computers, 64% a smartphone, 40% a tablet, and 15% a hybrid every week. And interest in using tablets for school work continues to grow. The university I am with currently institutes a system for checking student attendance using a mobile application; and in classes I teach, nearly 100 percent of students bring smartphones to the classroom.

The computing power of today's smartphones surpass that IBM's 1997 Deep Blue supercomputer had just two decades ago. With myriads of ways to apply mobile devices (phones, tablets, laptops, and hybrids) to educational activities, it is not surprising to find increasing use of them in higher education [4]. I incorporated mobile devices as core teaching and learning tools in a class on numerical solution to problems relevant to chemistry. The problems go beyond the "textbook" exercises that may be solved by hand or pocket calculators. Thus, these problems could only be tackled by a series of instructions to a computer, using a programming language. All the students used smartphones, although other

mobile devices were also allowed as computing tools. The app stores (Apple App Store and Google Play, for example) had plenty of apps to learn coding and to compile and execute codes, so neither textbooks nor calculators nor computer labs were needed. I provided guided explanations of how to use the apps in terms of scientific computation. All the coding exercises from the first day of class were related to chemical problems and progressed from simple to more non-trivial ones as students gained more programming skills. Students were given the liberty to do all the work by themselves or collaborate with or seek some help from other students, since many researches report the efficacy of collaborative learning [5]-[7]. They enjoyed the flexible learning climate and were excited to build apps on their own smartphones to deal with complicated problems, which otherwise might have been left only for ambitious students.

The capabilities of mobile devices, on the other hand, also entail the possibilities of high-tech cheating [8]. In the aforementioned class I let students use the devices for solving and answering test questions during the in-class exams. Problem-solving steps were required as well as the final answers to get full credit. Nonetheless, it was found that cheating was much more prevalent, despite tight proctoring and the university's honor system, than when the devices in exams were disallowed.

Accurate assessment of students' mastery of the subject a course is offering is an integral part of education, so it is imperative to prevent and detect academic dishonesty. For my class mobile device usage was essential in solving test questions, but it was extremely difficult to prevent and detect cheating in the traditionally administered paper-and-pen exams. Thus, starting with the second exam, the tests were switched from offline to online. Corrigan-Gibbs and coworkers [9] argue that honor codes alone are ineffective in reducing cheating in online exams, and pre-exam warnings can reduce cheating by 50%. The percentage is still so high, in my opinion, that additional methods to further thwart cheating are necessary. As technology evolves, new ways to cheat emerge and novel methods to mitigate this adverse effect in technology-enhanced classes are in great need of development. There are diverse IT-based cheating detection methods and services ranging from remote proctoring [10] to post-exam quantitative analysis for cheating detection [11], [12]. Sophisticated online proctoring services has proliferated over the last five years, but the services put extra financial burden on both students and schools and there is concern about privacy intrusions on the part of test takers [13]. It would be far more educational to prevent academic dishonesty in the first place rather than detect and penalize students who cheated. To this end I developed a method based on only standard web technologies without requiring

Chang Jae Lee, Department of Nanochemistry, Sunmoon University, Asan, Korea

special hardware. In the following sections I discuss the details of the approach and demonstrate its efficacy in curbing student cheating during mobile and online exams.

II. ANTI-CHEATING MEASURES

A. Basics of randomization of exam problems

A well-known technique for discouraging students to cheat is to introduce certain degree of randomness into testing. Common examples include randomly assigning seats during the test administration [14], shuffling the order of the problems so that not everyone has the same exam sheet [15], create several versions of the same test [16], [17]. These methods do have some efficacy in preventing student cheating for usual in-classroom examinations, but they are totally useless for online and m-learning testing environments for which mobile device usage is a part of teaching or learning. Furthermore, studies show that perceived difficulty of an exam may be influenced by the order of the questions in the exam [18]. Implementing the third method in traditionally administered exams incur a large workload on the part of the instructor, and the degree of randomness is inherently limited and not enough if the class is large. It, therefore, is necessary to overcome these shortcomings, and that can be achieved using the latest web technologies.

For the class I taught, student assessments were made using an online exam system. To take the test, each student was required to log in with his/her roster number along with a password. Each student was given a problem set, uniquely generated by the system based on the roster number input by the student. The core ingredient of this system is a random number generator. Since the random number generator produces the same sequence of numbers each time it is run, a variable random seed for the generator needs to be provided to obtain different sequences. The random seeds for the system are constructed with each student's roster number, which is unique for a particular student, combined with the time stamp at which the student logs in to take the exam. There are several programming or scripting languages that can be used to create an online exam system. For security reasons, the making of exam questions, and automated grading in particular, need to be done on a server. The PHP language [19] was chosen for the server-side scripting, but the same logic can easily be adapted to other server-side scripting languages such as Node.js[20]—all freely available on the web. Similar randomization technique is used by Idris and coworkers [21] for homework problem generation and grading, but they used additionally MATLAB® [22], which is a proprietary software, and did not give any discussion on displaying dynamically generated mathematical equations on the web.

The exam system I developed consists of HTML files for registering and logging in, PHP files for exam creation, grading, and communication with a web-based database to record and retrieve students' scores. Only exam generation and grading are discussed in this paper. The following code snippet is a part of the HTML file for logging in to take exam1:

```
<form action="exam1.php" method="post">
```

```
target="_parent">
Roster No.: <input type="text" size = "12"
name = "roster" >
Password: <input type="password"
size = "12" name = "pwd" >
<br />
<input type = "submit" value = "Log in" />
</form>
```

Then in the **exam1.php** file a random number is generated as:

```
$roster_no = $_POST['roster'];
mt_srand($roster_no * time());
$rn = mt_rand(1,100);
```

The first line in the above code snippet reads the roster number a student entered in the login HTML file, and in the second line the information and the time this line in the script is executed are used to generate a random seed. Note that from PHP 4.2.0 on, the random number generator is automatically seeded and it is not needed to use the `mt_srand()` function. Nonetheless, I keep using this step to completely control random seeding. In the third line a random integer between 1 and 100 inclusive is generated using the built-in PHP function `mt_rand()`. Random numbers generated this way are used or modified to create randomized problems. Some exemplar problems in chemistry are given in the next subsection.

B. Random question examples

(B.1) Ideal gas law:

The ideal gas law is given by the equation $pV = nRT$. It has four variables p , V , n , T , and a constant R . The equation may be rewritten in five different ways: $p = nRT/V$, $V = nRT/p$, $n = pV/(RT)$, $T = pV/(nR)$, $R = pV/(nT)$. The ideal gas law can be formulated in more diverse forms, but for demonstration purposes only the five simple variations are considered. Then it is possible to construct five question types

- What is the value of p if V , n , R , T are given?
- What is the value of V if p , n , R , T are given?
- What is the value of n if p , V , R , T are given?
- What is the value of T if p , V , n , R are given?
- What is the value of R if p , n , R , T are given?

These questions may be constructed with the help of random numbers as the following algorithm shows.

Algorithm 1. Randomized question for ideal gas

```
Create arrays containing variables, names, and units:
vars = array(p, V, n, R, T)
varNames = array("p", "V", "n", "R", "T")
units = array("atm", "L", "mol", "atm·L/(mol·K)", "K")
rn = random(1,5): a random integer between 1 and 5
inclusive to select a question type.
if rn = n then
    for k = 0 to 4, k ≠ n do
        rn_k = random(min vars[k], max vars[k])
        vars[k] ← rn_k
    end for
```

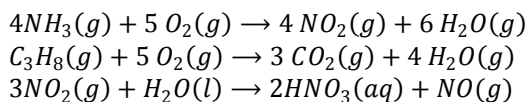
```

end if
print "What is the value of " + varNames[n-1] + " in " +
units[n-1] + ", if "
for k=0 to 4, k≠n
    print varNames[k] = vars[k] + units[k] + ", "
    (or "?" if at the end of the question sentence)
end for
    
```

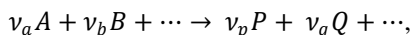
In Algorithm 1, arrays are assumed to be zero-based and the plus sign in the **print** statement denotes string concatenation. In addition to the five randomized question types, values of the variables that are to be given in the problem are randomly changed, using separate random numbers. Thus, this procedure creates an infinite number of questions, and no two students have a chance to have an identical question.

(B.2) Stoichiometric calculations:

Stoichiometry is concerned with the relationships among amounts of substances in chemical and physical changes. There are numerous chemical equations of similar complexity. A few examples are



A general chemical reaction involving N substances may be written as



where $\nu_a, \nu_b, \dots, \nu_N$ are stoichiometric coefficients, and $A, B, P, Q \dots$ denote chemical compounds.

Typical problem goes something like: "How much compound B is needed to completely react with m grams of compound A ?" or "How much compound P is produced, if n moles of compound A completely react with compound B ?" Thus, randomization may be applied to selecting (1) one from a pool of chemical equations; (2) a pair of compounds in a chemical equation, one as an input (amount is given) and the other as an output (amount to be determined); (3) input and output units.

Thus, we are able to construct (no. of chemical equations) \times (no. of partial permutations $P(N,2)$ of compounds) \times (no. of units) different questions. For example, with the above three equations with four compounds and two units grams and moles alone, $3 \times P(4,2) \times 2 = 72$ different questions can be created. In addition, the numerical values of the coefficients can be randomly varied. Thus, there is no chance for students to have identical exam problems.

The randomization algorithm for stoichiometry problem is not much different from Algorithm 1. In this case, arrays for the chemical equations, compounds, and units are declared; and random numbers pick one element (or two for the compounds) out from each of the arrays, and the quantity of the selected compound as the input is also randomly changed.

(B.3) Atomic term symbol:

In this example, I show that randomization can be applied to problems that involve no numerical calculations. Atomic term symbols are non-ambiguous designation of electronic states of the atom. If the total spin is S and total orbital

angular momentum is L , the total angular momentum J has the values restricted to $J = |L-S|, |L-S|+1, \dots, L+S-1, L+S$. The letters S, P, D, F, G, \dots correspond to $L=0,1,2,3,4, \dots$. The term symbol for a given S, L , and J is $^{2S+1}L_J$, where L denotes the letter corresponding to the numerical value of L . One may think of many question types for these angular momenta. For example, a couple of relevant questions might be

- "Given L and S , what are possible J 's and term(level) symbols?"
- "What are L, S , and J values for a given term(level)?"

In the above question types both L and S values can be randomized freely, but they should be nonnegative and for physical reasons should not be too large. The total spin S can be either whole or half integers. The value of J can also be randomized, subject to the restriction mentioned above. The algorithm for randomizing angular momenta to build test questions is given below:

Algorithm 2. Randomized questions for angular momenta

```

terms = array("S", "P", "D", "F", "G", ...)
L = random(0, L_max)
S = random(0, S_max) + [random(1,2)-1]/2.0
print "(1) Given L and S, what are possible J's and
term(level) symbols?"
J = random(1, L+S-|L-S|+1) + |L-S|-1
L ← terms[L]
Derive term symbol 2S+1LJ
print "(2) Given 2S+1LJ, what are L, S, and J values?"
    
```

The total number of questions is $(L_{\max}+1) \times (2S_{\max}+1)$ for the first question type, and this number should be multiplied by (no. of J values) for the second question type. To give an idea about the number of questions that can be written, consider, only the two questions types with the **terms** array of size 5 ($L_{\max} = 4$), and $S_{\max} = 3/2$. There are $5 \times 4 = 20$ variations of the first question type and there are 43 cases of J , which gives 860 variations of the second question type!

C. Displaying complex mathematical expressions in exam questions

The examples in the previous subsection include rather simple mathematical expressions. These simple cases can be rendered on a web page using HTML tags such as $\langle \text{sub} \rangle, \langle \text{sup} \rangle$ for subscripts and $\langle \text{sup} \rangle, \langle \text{sub} \rangle$ for superscripts, and HTML character entities and symbols such as $\&\#8594$; or $\&\rarr$; for right arrows. But for more complicated mathematical expressions these tags and symbols alone are inadequate. Moreover, these expressions need to be dynamically generated because randomization algorithm is employed to construct exam questions. As stated earlier, the core language to build randomized online exams is PHP. Unfortunately, PHP lacks a mathematical expression rendering API, and a viable solution is to embed Mathjax [22] for processing mathematical expressions in a php file that is used to build test problems.

Suppose we wish to create the following problems and display them on a web page:

1. In a chemical reaction one mole of A produces

n moles of P as given by the equation

$$A \xrightleftharpoons[k_b]{k_f} n P.$$

If $k_f = x$, $k_b = y$, $[A]_0 = a$ moles dm^{-3} , calculate the time taken to produce b moles dm^{-3} of P .

2. For a van der Waals gas the following relation hold:

$$\Delta C_m = C_{p,m} - C_{v,m} = \lambda R, \quad \frac{1}{\lambda} = \frac{(2 V_r - 1)^2}{4 V_r^3 T_r}.$$

Calculate ΔC_m for hydrogen molecule at $c^\circ\text{C}$ and d atm.

Our task is to dynamically change variables n , x , y , a , b , c and d in the problems with PHP and render the equations to a web page with the help of Mathjax. To do this we create a PHP file with PHP commands to randomize variables, and in the same PHP file embed HTML codes that communicate with Mathjax to display the final problems.

Let us illustrate this procedure using a simple example asking students to evaluate the integral $\int_{\pi/a}^{\pi/b} \cos x \, dx$, given randomized integers a and b in the range $[1, 10]$ and $[0,9]$, respectively. Below a code snippet in a PHP file **Exam.php** to display the example test is listed:

Exam.php

```
<?php
...
$a = mt_rand(1,10);
$b = mt_rand(0,$a-1);
...
?>

<!DOCTYPE html>
<html>
<head>
<script type='text/javascript' async
src='some link for MathJax.js'>
</script>
</head>
<body>
...
Evaluate the integral \[ \int_{\pi/a}^{\pi/b} \cos x \, dx \]. ...
</body>
</html>
```

In the above code, random numbers are generated and assigned to variables in the first PHP part, and these numbers are exposed in the HTML part using the PHP **echo** commands. Mathematical expressions are typeset with LaTeX commands and are processed by the Mathjax javascript engine, which is included in the HTML head section. Figure 1 shows the example test displayed on a web page. The variables $a=3$ and $b=4$ are randomly generated.

Exam No. 1

1. Evaluate the integral

$$\int_{\pi/4}^{\pi/3} \cos x \, dx$$

Your answer 1:

Submit answer

Fig. 1. Example test shown on a web browser, demonstrating mathematical expression rendering mixed with randomly generated variables. A textbox and a button to input and submit student's answer are also shown.

D. Automatic grading

Since variables in the questions are generated randomly with great diversity, it is virtually impossible to manually grade answers students submitted. After the **Submit answer** button in Fig. 1 is pressed, the student's answer is handed over to a new PHP file for automated grading. This requires storing of information during an exam session, so we start with the appropriately-named **session_start()** PHP session command. The correct answer is constructed in the **Exam.php** file as well. It could be done in the grading PHP file, but that would involve transferring many (randomized) variables between the two PHP files, which is better avoided. Thus, the first PHP block is modified as:

Exam.php

```
<?php
session_start();
// Random seeding for each student goes here.
$a = mt_rand(1,10);
$b = mt_rand(0,$a-1);
$answer1 = sin(M_PI/$b) - sin(M_PI/$a);
$_SESSION['ans_1'] = $answer1;
```

In the above code the correct answer is stored in the variable **\$answer1**, and **M_PI** is $\pi = 3.141592653\dots$. The last line is for setting a global session variable, which may be accessed in the other PHP file for grading.

The textbox and the button below the problem statement in Fig. 1 are added in the HTML part of **Exam.php** as elements of a form with the attributes **<form method="post" action="Grade.php">**. The textbox has the attributes **type="number" name="urans1"**.

The posted answer is handled by the **Grade.php** for grading, and the bare-bones code is listed below:

Grade.php

```
...
session_start();
$ans1_submitted = $_POST['urans1'];
$correct_ans1 = $_SESSION['ans_1'];
$score = 0.;
$tol = 0.01;
```



```
if (abs(($ans1_submitted-$correct_ans1)
/$correct_ans1)<= $tol){
    $score += 100;
}
else
    echo "Wrong answer. <br/>";
echo "<hr/>";
echo "Total score: $score";
...
```

In this code the student's answer posted and the correct answer that **Grade.php** accessed as a session variable are compared, and if their difference is within some tolerance, **\$tol**, the student's score is incremented. This method can be easily extended if there are more questions in an exam, and partial credits can also be incorporated without much complication. Figure 2 shows the (full) code in action on a web page.

Problem 1

Your answer 1 is: 0.5
 Correct answer is: 0.158918622598
 You got the wrong answer for **Problem 1**.

Your total score is: 0

Bye!

Fig. 2. Grading and feedback for the example test as displayed on a web browser.

III. CHEATING PREVENTION EFFICACY

As mentioned in Subsection II.A, students in the class I taught were encouraged to use mobile devices during lectures and for exercises. These devices were allowed even in in-class exams. But collaboration and information sharing were disallowed in the exams, and in the first exam only an honor system was implemented as a means of dispelling academic dishonesty. The exam was an open book one and problems were identical for all students. They were asked to write down on the answer sheet not only the final answers but also the computational procedures for solving the problems. While grading the exam papers, I noticed that many students simply copied or slightly modified the answers of a student who excelled in the class. It appeared that the particular student texted or emailed his answers to fellow students using his mobile phone.

From the second exam and later, measures for cheating prevention based on the random generation of test questions discussed in the previous section were introduced. For the small class of 10 or so students, it was not difficult to build a pool of test questions for each problem as described in subsection II.B, with question types exceeding the number of students, each question being of similar complexity or difficulty. The exams were posted online, although to be taken in class; and after logging in, each student was randomly assigned a series of questions from each of the pools. Students still used mobile devices for numerical

calculations and submitting answers online for automated grading but had to turn in hand-written exam papers, documenting the steps for arriving their answers. Students were given ample time (three hours) for each exam but were forewarned that they would receive different problem sets. The result of the anti-cheating measures was successful: As usual, the student who excelled in the classroom finished the exams earlier than the rest of the students. So, he had extra time for helping any other student if he desired so, but it was apparent he did not. There was a wide distribution of students' online scores as opposed to the first exam for which many students had the same score; and the written exam papers, upon close scrutiny, revealed no sign of cheating.

IV. CONCLUSIONS

I have described in this paper a system for curtailing cheating in online exams in detail. The system is based on automatic generation of exam problems using randomization algorithms. The randomization can be made arbitrarily infinite, and with the automatic grading feature this approach may be extended even to massive open online courses. This approach has been implemented in two different classes, one face-to-face and the other fully online. All the tests were open book and the questions were delivered via the Internet. As discussed in the previous section, I believe there had been no instances of cheating during in-class tests after employing the randomization. In the other class, online tests were non-proctored whatsoever, but the students were instructed that collaboration or group study *was* allowed. For this online course an anonymous end-of-semester questionnaire was conducted, which contained a question whether a student collaborated with or sought help from other students. Twenty five out of 26 students who returned the questionnaire replied they resorted to neither collaboration nor outside help in taking tests. Consequently, automated vastly random generation of test problems does a good job at counteracting academic dishonesty, and therefore should be a powerful, cost-effective, privacy non-invasive alternative or additional anti-cheating method.

REFERENCES

- [1] Available online at: <https://www.statista.com/statistics/201183/>.
- [2] A. Berenguer *et al.*, "Are Smartphones Ubiquitous?" *IEEE Consumer Electronics Magazine*, vol. 2, Jan. 2017, pp. 104 - 110; and references cited therein.
- [3] Available online at: <https://www.pearsoned.com/wp-content/uploads/2015-Pearson-Student-Mobile-Device-Survey-College.pdf>.
- [4] *Handbook of Mobile Teaching and Learning*, Y. Zhang, Ed., Berlin: Springer, 2015.
- [5] V. Vega and Y. Terada, "Research Supports Collaborative Learning," *Edutopia*, December 5, 2012. Available online at: <https://www.edutopia.org/stw-collaborative-learning-research>.
- [6] I. Arroyo *et al.*, "Collaboration Improves Student Interest in Online Tutoring," in *Artificial Intelligence in Education, 18th International Conference, AIED 2017, Wuhan, China, June 28 - July 1, 2017 Proceedings*, E. André *et al.* (Eds.), Springer, 2017.
- [7] D. Francescato *et al.*, "Evaluation of the efficacy of collaborative learning in face-to-face and computer-supported university contexts," *Computers in Human Behavior*, vol. 22, no. 2, 2006, pp. 163-176.
- [8] Available online at: <https://www.chronicle.com/article/Cheating-Goes-High-Tech/132093>.
- [9] H. Corrigan-Gibbs, N. Gupta, C. Northcutt, E. Cutrell, and W. Thies, "Deterring Cheating in Online Environments," *ACM Transactions on Computer-Human Interaction*, vol. 22, no. 6, Article 28, 2015.

- [10] D. Woldeab, T. Lindsay, and T. Brothen, "Under the Watchful Eye of Online Proctoring," in *Innovative Learning and Teaching: Experiments Across the Disciplines*, I. D. Alexander and R. K. Poch(Eds.), *University of Minnesota Libraries Publishing*, Minneapolis, MN, USA, 2017, chap. 15.
- [11] K. A. D'Souza and D. V. Siegfeldt, "A Conceptual Framework for Detecting Cheating in Online and Take-Home Exams,"*Decision Sciences Journal of Innovative Education*, vol. 15 no. 4, 2017.
- [12] *Handbook of Quantitative Methods for Detecting Cheating on Tests*, Cizek and J. A. Wollack (Eds.), New York:Routledge, 2017.
- [13] Available online at:
<https://www.insidehighered.com/digital-learning/article/2017/05/10/online-exam-proctoring-catches-cheaters-raises-concerns>.
- [14] A. K. Clark and N. M. Kinston, "A Brief History of Research on Test Fraud Detection and Prevention," in *Test Fraud: Statistical Detection and Methodology (Routledge Research in Education)*, N. M. Kinston and A. K. Clark (Eds.), New York: Routledge; 1st ed., 2014, pp. 4-7.
- [15] S. Smith, "Controlling Cheating in Online Courses: A Primer,"©2013 Pearson.Available online at:
<https://www.pearsonmylabandmastering.com/global/assets/pdf/controlling-cheating-in-online-courses.pdf>.
- [16] N. Chiesl, "Pragmatic methods to reduce dishonesty in web-based courses,"*Quarterly Review of Distance Education*, vol. 8, no. 3, 2007, pp. 203–211.
- [17] D. W. Bedford, J. R. Gregg, and M. S. Clinton, "Preventing online cheating with technology: A pilot study of remote proctor and an update of its use,"*J. Higher Education Theory and Practice*, vol. 11, no. 2, 2011,pp. 41–58.
- [18] T. F. Pettijohn II and M. F. Sacco, "Multiple-choice exam question order influences on student performance, completion time, and perceptions,"*J. Instructional Psychology*, vol. 34, no. 3, 2007, pp. 142-149.
- [19] Available online at: <http://php.net/>.
- [20] Available online at: <https://nodejs.org/en/>.
- [21] M. A. Idris, T. M. I. Mahlia, N. I. Om, I. Hussein, and M. R. Jamil, "Effectiveness of Computerized Homework Implementation in Thermodynamics: Correlation Between Doing the Homework and the Grade," in *7th International Conference on University Learning and Teaching(InCULT 2014) Proceedings*, C. Y. Fook, G. K. S. S. Narasuman, L. L. Fong, S. B. A. Rahman(Eds.), Singapore:Springer, 2016, pp. 627-637.
- [22] Available online at: <https://www.mathjax.org>.