How Much Randomization is Needed to Deter Collaborative Cheating on Asynchronous Exams?

Binglin Chen, Matthew West, Craig Zilles
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{chen386, mwest, zilles}@illinois.edu

ABSTRACT

This paper investigates randomization on asynchronous exams as a defense against collaborative cheating. Asynchronous exams are those for which students take the exam at different times, potentially across a multi-day exam period. Collaborative cheating occurs when one student (the information producer) takes the exam early and passes information about the exam to other students (the information consumers) that are taking the exam later. Using a dataset of computerized exam and homework problems in a single course with 425 students, we identified 5.5% of students (on average) as information consumers by their disproportionate studying of problems that were on the exam. These information consumers ("cheaters") had a significant advantage (13 percentage points on average) when every student was given the same exam problem (even when the parameters are randomized for each student), but that advantage dropped to almost negligible levels (2–3 percentage points) when students were given a random problem from a pool of two or four problems. We conclude that randomization with pools of four (or even three) problems, which also contain randomized parameters, is an effective mitigation for collaborative cheating. Our analysis suggests that this mitigation is in part explained by cheating students having less complete information about larger pools.

ACM Classification Keywords

K.3.1 Computers and Education: Computer Uses in Education

Author Keywords

asynchronous exams; problem randomization; collaborative cheating; computerized testing.

INTRODUCTION

Exams are one of the most widely used methods for student assessment in college education, especially in introductory courses. However, at many universities these courses are large (e.g., 200+ students), and running exams for them can be very demanding [8, 6, 15]. One alternative to paper-and-pencil exams that has the potential to improve the exam experience for both faculty and students is computer-based testing [3, 10]. The major benefits of computer-based testing are that it greatly reduces the overhead of running exams and broadens the kinds of problems that can be automatically graded [15]. To handle computer-based exams for large enrollment classes with a small testing center, running the exams *asynchronously* (i.e., allowing students to choose their exam time within a given exam period) has been proposed as a solution [5, 16]. This asynchronous strategy also gracefully accommodates student exam time conflicts.

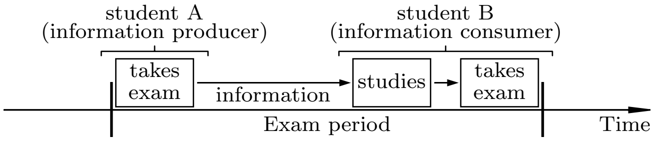

Figure 1. A typical collaborative cheating process where information producers take an exam before information consumers and pass the information about the exam to the information consumers.

One major drawback of running exams asynchronously is its potential to facilitate collaborative cheating. In fact, previous studies have found that "learning what is on a test from someone who has already taken it" is the most frequently reported cheating method among college students who have admitted to cheating [7, 13]. These results from face-to-face (i.e., non-online) classes whose exams may not even be asynchronous suggest that collaborative cheating is a serious issue that asynchronous computerized exams need to address. In contrast, a previous study [4] examined a dataset of asynchronous computerized exams and found that on average students who take exams later tend to get lower scores. This result suggests that over the entire student population as a whole the effect of collaborative cheating is overwhelmed by other factors. However, it did not address how advantageous collaborative cheating is for those students who did cheat or how we can effectively mitigate collaborative cheating.

In this paper, we define collaborative cheating specifically as a cheating activity where information about exam content is passed from a student who has taken the exam to a student who has not yet taken the exam. We label such students as *information producers* and *information consumers*, respectively. As depicted in Figure 1, the producers take an exam earlier than the consumers and pass the information about the exam to the consumers. A consumer could receive information from multiple producers, and a consumer could also be a producer

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

for a later consumer. With the information, the consumers study accordingly and then take the exam at a later time, thus gaining an unfair advantage over the rest of the class. This paper will focus on information consumers since they are the subgroup of cheating students directly benefiting from collaborative cheating and they are also the subset that we are able to identify in our dataset. We will refer to them as *cheaters* for simplicity throughout the rest of the paper.

This study is possible because of a dataset of computerized homeworks and exams that has three special fortuitous features. The first special feature is that a subset of the homework problems were also present on subsequent exams. We make use of this feature to look for students who disproportionately study the homework problems that are on the exam, indicating that they have likely received information about the exam problems. The second special feature is that each student only had a subset of these problems on their exams, where this subset was chosen randomly from problem pools of varying sizes. This variation in pool size allowed us to see how the amount of randomization affected student practice and performance. The third special feature is that there were other problems which only appeared on the exams, which we use as a control variable for a difference-in-differences analysis to estimate true effect sizes. Since problems are shared between exams and homeworks, this scenario is likely the "best condition" for collaborative cheating as students can study these problems if they have information about which ones are on the exam.

This paper is organized as follows. In the **Data Description** section, we discuss the course context of this study and the data we analyzed. In **Cheater Classification**, we describe a simple method to identify students who are information consumers by observing students' homework behaviors. In **Analysis of Exam Problem Scores**, we show that cheaters' advantage can be greatly reduced by introducing randomization of problem selection in addition to random problem parameterizations. In **Analysis of Problem Coverage**, we define a coverage metric and show that randomized problem selection makes it harder for cheaters to gain a large coverage advantage, which partially explains why randomized problem selection works. We finish with **Limitations** and **Discussion and Conclusions**.

DATA DESCRIPTION

The data was collected at a large public research university during the Spring 2017 semester. The particular course which we studied is an introductory undergraduate mechanical engineering course of 425 students. The course managed all 14 homework assignments via the PrairieLearn system. All 13 exams in the course were asynchronous computerized exams administrated via the same PrairieLearn system and held in the Computer-Based Testing Facility (CBTF). With IRB approval, we obtained all 363,847 homework records outside the CBTF as well as all 56,486 exam records in the CBTF for the class.

Computer-Based Testing Facility

The CBTF [15] is a computer lab with 85 seats for students and another 4 seats in a reduced distraction environment for students registered with the disability resource center. Each of the computers is outfitted with a privacy screen that prevents test takers from reading the screens of neighboring computers, and the network and file systems are strictly controlled. The facility is open and proctored 10–12 hours a day, 7 days a week to accommodate two to four thousand exams per week [16]. Students are not permitted to take written notes, photos, or other records into or out of the exam room. At their scheduled exam time, students have their identity checked by a proctor and are randomly assigned to a computer to deter cheating during an exam seating.

Exams within the CBTF are typically administrated as follows [16]: Classes assign a 3–5 day period for the students to take an exam depending on the class size; longer exam periods are used during finals week. Students are free to reserve any time during the exam period, provided that there are slots available at that time. Sign-ups for exams typically begin two weeks before the exam period begins. Generally, the exam periods of exams from different classes overlap each other, and the CBTF is almost always running a number of distinct exams concurrently.

Homework data from PrairieLearn

PrairieLearn [14] is an online problem posing system that permits the specification of *problem generator*s, each of which is capable of generating randomly parameterized *problem instance*s. Problem generators are typically written to generate problem instances with different numeric values or other small changes so that the correct answer is different. This allows students to practice any particular type of problem indefinitely with immediate feedback on the correctness of their attempts.

Each homework assigned in PrairieLearn consists of a set of problem generators. Students need to answer problem instances generated by a problem generator to earn points. Each problem instance can only be attempted once. A new problem instance will be generated after each attempt, regardless of the correctness of the submitted answer. For the course under study, each homework provided a large set of problems, but students could get full points on the homework by answering only a subset (often around half) of the provided problems. The course allowed students to access any homework and practice with the problem generators after homework deadlines.

Each of the homework records has the form **(problem id, student id, date)**. The problem id is a unique identifier for each problem generator. The student id is a unique identifier for a student. The date is a timestamp of when the student attempted a problem instance generated by the problem generator. For this analysis, it isn't important what the scores on practice records are, thus score is not a part of each record.

The homework data collected has 14 homework assignments and 98 unique problem generators. On average, each homework has 7 problem generators, and none of the problem generators was repeated in multiple homework assignments.

Exam data from PrairieLearn

Each exam in PrairieLearn consists of a set of *problem slot*s where each slot has a corresponding *pool* of problem generators. For each student, PrairieLearn will randomly select a generator for each slot and then randomly generate a problem

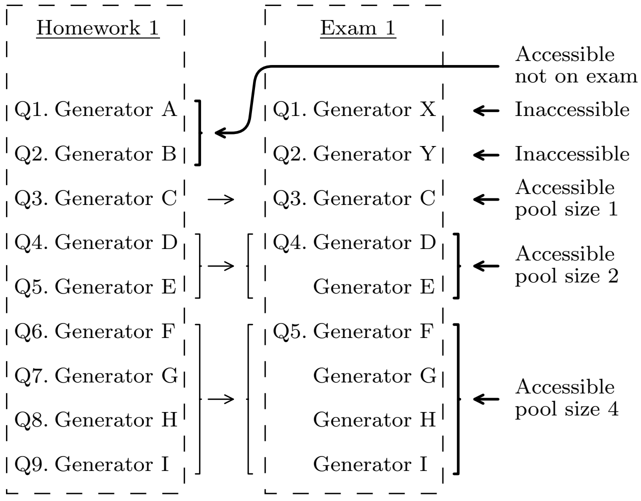

Figure 2. Five categories of problem generators with respect to Exam 1.

instance with each selected generator. Once generated, the set of problem instances is fixed, and students have to answer these instances to earn points. The use of problem generators on exams makes cheating harder, as students get random problem instances whose answers are unlikely the same as that of other students. The course allowed students to have multiple attempts at each problem instance with a partial-point schedule controlled on a per problem slot basis. For example, a schedule of $[100\%, 70\%, 50\%, 0\%]$ for a problem generator means a student will get 100% credit on that problem instance if the first attempt is correct. The student will get 70% of the points if the first attempt is wrong but the second attempt is correct and so forth.

The course exams are a mixture of problem generators from preceding homeworks and problem generators that students have not previously seen. Specifically, for each exam, we can categorize all of the problem generators on the exam and previous homeworks assignments on the same topic into the following five *categories*: (1) inaccessible, (2) accessible pool size 1, (3) accessible pool size 2, (4) accessible pool size 4 and (5) accessible not-on-exam. Specifically, *inaccessible* problem generators cannot be accessed outside the exam and will appear in every student's exam. *Accessible* generators can be accessed outside the exam, meaning that these generators appeared in the corresponding set of homework assignments. Pool size k means the generator belongs to a problem slot whose pool contains k problem generators. In an accessible pool size of 4, one problem generator of the 4 would be used to create a problem instance on a student's exam. The different pool sizes essentially introduce different level of randomization in problem selection. On-exam and not-on-exam indicate whether the problem generator is on the exam. Since the first four categories of problem generators are on the exam while only the last one is not, we omitted "on-exam" for the first four categories for simplicity. An illustrative example of the five categories is shown in Figure 2. This particular arrangement of problem generator selection was never made public, though throughout the semester students could have learned that some problem generators are shared between homeworks and exams, and that everyone gets slightly different set of problem generators.

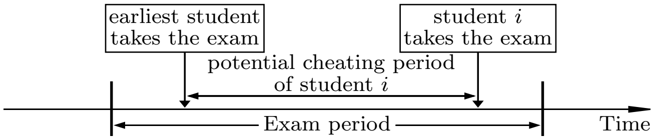
earliest student takes the exam
student i takes the exam
potential cheating period of student i
Exam period
Time

Figure 3. The potential cheating period of student i on an exam.

Each of the exam records has the form **(exam id, problem id, category, student id, score, date)**. The problem id, student id, and date are the same as for the homework data. The exam id is a unique identifier for each exam. The category is one of the five categories defined above; since these records are about exam problems, the accessible not-on-exam category will not appear in any of these records. The score is a real number ranging from 0 to 100, indicating the percentage of points the student got based on the partial-point schedule of the problem slot.

The exam data collected for the course has a total of 13 exams which students could take and 98 unique problem generators. 57 out of the 98 problem generators did appear in homework. On average, each exam uses 11 problem generators and some of the problem generators were repeated in multiple exams. Out of all exams, there were 54 problem generators categorized as inaccessible, 41 categorized as accessible pool size 1 (41 problem slots of pool size 1), 16 categorized as accessible pool size 2 (8 problem slots of pool size 2), 32 categorized as accessible pool size 4 (8 problem slots of pool size 4), and 195 categorized as accessible not-on-exam. Note that a few problem generators were re-used in different roles on different exams, so the total of the previous numbers is more than the total number of problem generators.

CHEATER CLASSIFICATION

By the nature of collaborative cheating, cheaters must use the information about the exam that they have obtained if they are to get an advantage. If some of the exam problem generators were accessible through the homework system, it is natural for the cheaters to focus on practicing these generators in PrairieLearn before their exams. Thus they are likely to disproportionately practice problem generators which are on the exams as compared with the rest of the class.

Specifically, for exam x and student i who has taken the exam, we only consider practice records that are within student i's *potential cheating period* of exam x. We define the potential cheating period of student i on exam x as the time period between when the earliest student started exam x and when student i started exam x, as depicted in Figure 3. Essentially, this is the time period when information of exam x might be available to student i and student i can practice strategically based on this information. We denote the number of student i's practice records during student i's potential cheating period of exam x as $n_{x,i}$ and the number of practice records among the $n_{x,i}$ practice records whose problem generator was part of exam x as $k_{x,i}$.

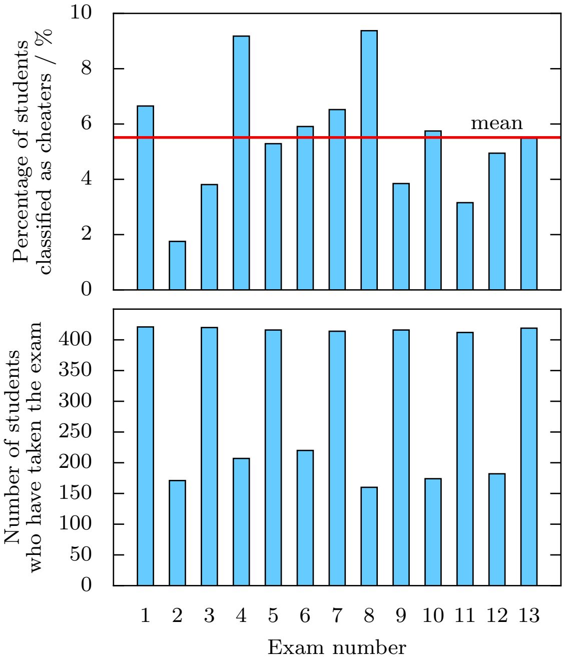

Figure 4. The upper plot shows the percentage of students classified as cheaters for each exam. The horizontal line represents the mean, which is 5.5%. The lower plot shows the number of students who have taken each exam. Exams with even numbers are second chance exams.

Using the plain fraction $k_{x,i}/n_{x,i}$ to classify student i as a cheater on exam x is problematic since a student with a fraction of 100% but who has only attempted a single problem instance is likely due to chance, whereas a student with the same fraction who has attempted 1,000 problem instances is very likely to be cheating. Thus we used binomial distributions rather than plain fractions to classify students as cheaters on a per exam basis. To do this we first computed the average fraction of attempted problem instances that were generated by problem generators that were on exam x as follows:

$$p_x = \frac{\sum_i k_{x,i}}{\sum_i n_{x,i}}, \tag{1}$$

where p_x is the *average probability that a problem generator practiced by a student is on the exam* for exam x; both sums are over students who have taken exam x.

With p_x computed, we assume that a non-cheating student i will have a number of practice attempts that were on on-exam problem generators ($k_{x,i}$) distributed according to a binomial distribution with number $n_{x,i}$ and success probability p_x. Thus we define a random variable $K_{x,i}$ that follows this binomial distribution for exam x and student i. If a student's $k_{x,i}$ greatly exceeds what is predicted by this distribution, that is, if $P(K_{x,i} \geq k_{x,i})$ is less than some threshold, we classify student i as a *cheater* for exam x, otherwise we classify student i as a non-cheater for exam x. The threshold used in this study is 0.0001, which we would expect to give one false positive per 10,000 students and our population is only 425 students. We will discuss potential issues with this method in the Limitations section. Note that this method will not classify students who are only information producers as cheaters. This is actually desired for the following analysis since information producers would not likely benefit from collaborative cheating for exams that they took earlier than their collaborators.

With the above method, we classified students as a cheater or non-cheater on a per exam basis and computed the percentage of students that were classified as cheaters for each exam[1]. We plotted the percentage of cheaters for each exam and the number of students who have taken each exam in Figure 4. Even-numbered exams are second chance exams that students can take to partially replace their grades on the preceding first chance exam (which is odd numbered). Exam 13 is the final exam. As the figure shows, none of the exams has more than 10% of the students classified as cheaters. The mean is 5.5% (95% CI [4.2%, 6.8%]), which is shown as the horizontal solid line in the upper plot of the figure.

ANALYSIS OF EXAM PROBLEM SCORES

With the cheaters and non-cheaters labeled by the method in the previous section, we wanted to study what advantage cheaters have on exam problem scores. For this analysis, we will study the four categories of problem generators that are on the exam.

Naive method and result

One naive way to study the impact of randomized problem selection on exam problem scores is to compare the mean score of cheaters to that of non-cheaters. To do this, we divided exam records into eight groups based on the category of each record and whether the student associated with the record is classified as a cheater on that exam. We then computed the mean score for each group and plotted the results with 95% confidence intervals in Figure 5.

Most notably, Figure 5 shows cheaters performing about 13 percentage points better than non-cheaters when every student was given the same (i.e., pool size 1) accessible problem generator. For larger (accessible) pools this difference is shown to drop to within 2 percentage points. This suggests that increasing pool size makes collaborative cheating less effective. Finally, it shows cheaters performing about 2 percentage points worse than non-cheaters on inaccessible problem generators.

However, to estimate the true score advantage of cheaters, we shouldn't just look at the raw differences at each pool size. This would tend to underestimate the advantage of cheating, because the cheaters seem to be slightly weaker on average as indicated by their lower mean performance on inaccessible problem generators. We attempt to correct for this in the next section.

Difference in differences method

To obtain a more precise estimate of cheaters' score advantage and confidence interval, we employed the difference in differences method [1, 2]. Difference in differences is a statistical method frequently used to study the differential effect of an experimental condition on two groups that differ in one important attribute. It first takes the measurements of two groups

[1] Only students who have taken the exam were included in the calculation for each exam, since some students missed some of the exams.

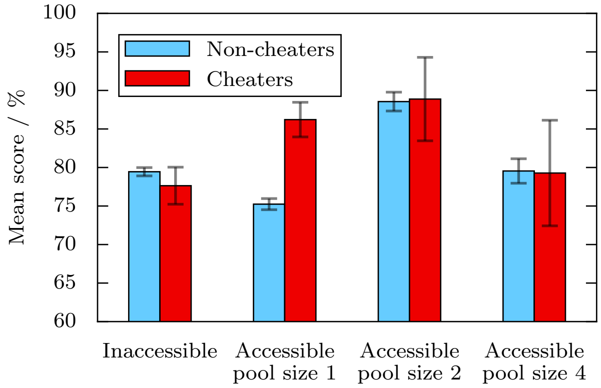

Figure 5. Mean score of exam records in each category for cheaters and non-cheaters. The error bars correspond to 95% confidence intervals of the means.

under a control condition. It then takes measurements of the two groups under an experimental condition. The difference in differences effect size δ is then calculated as

$$\begin{aligned}\delta = (&\text{avg measure of group A under experimental condition}\\ &-\text{avg measure of group A under control condition})\\ -(&\text{avg measure of group B under experimental condition}\\ &-\text{avg measure of group B under control condition}).\end{aligned} \tag{2}$$

An illustrative example is shown in Figure 6. Under the control condition, group A's mean measure of interest is α while group B's mean measure of interest is β higher than that of group A. Under the experimental condition, group A's mean measure of interest increases by γ while group B's mean measure of interest increases by δ in addition to γ. This δ is what the difference in differences method is trying to estimate.

The difference in differences method is often framed as a linear regression with ordinary least squares over all of the measurements as follows:

$$m = \alpha + \beta b + \gamma e + \delta be, \tag{3}$$

where m, b, e are observed values from each measurement, defined as follows:

- m: the measure of interest of the measurement,
- b: 1 if the measurement is associated with group B, 0 otherwise,
- e: 1 if the measurement is under experimental condition, 0 otherwise,

and $\alpha, \beta, \gamma, \delta$ are the coefficients that we want to estimate, with their meanings depicted in Figure 6. The major benefit of framing difference in differences as linear regression is that confidence intervals of the coefficients can be obtained conveniently from the regression.

In our case, the two groups are cheaters and non-cheaters and the measure of interest is their score. We treat inaccessible problem generators as a control condition, since it is less likely to be affected by collaborative cheating. We treat accessible problem generators with a specific pool size as an experimental condition. We will discuss the appropriateness of the difference in differences method for this analysis as well as this particular setup of control and experimental conditions in the Limitations section. The linear regression we study for difference in differences can thus be specified as follows:

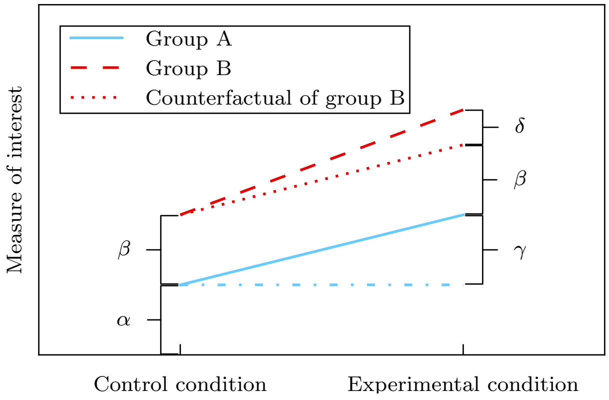

Figure 6. An example of difference in differences. The difference in differences method will capture the value δ, which is the differential effect between group A and group B under the experimental condition.

$$z = \alpha + \beta c + \gamma_1 s_1 + \gamma_2 s_2 + \gamma_4 s_4 + \delta_1 cs_1 + \delta_2 cs_2 + \delta_4 cs_4, \tag{4}$$

where z, c, s_1, s_2, s_4 are observed values from each exam record, which has the format (exam id, problem id, category, student id, score, date) as described in the Data Description section. These observed values are defined as follows:

- z: the score of the record,
- c: 1 if the student associated with the record is classified as a cheater on the exam, 0 otherwise,
- s_k: 1 if the problem generator associated with the record is accessible pool size k, 0 otherwise,

and $\alpha, \beta, \gamma_1, \gamma_2, \gamma_4, \delta_1, \delta_2, \delta_4$ are the coefficients that we want to compute, which can be interpreted as follows:

- α: the mean score of non-cheaters on inaccessible problem generators,
- β: the mean score difference between cheaters and non-cheaters on inaccessible problem generators,
- γ_k: the mean score difference between inaccessible problem generators and accessible pool size k generators for non-cheaters,
- δ_k: the additional mean score difference that cheaters have between inaccessible problem generators and accessible pool size k generators.

Difference in differences result and discussion

Since we mainly care about cheaters' mean score advantage over non-cheaters on the three different pool sizes, δ_1, δ_2 and δ_4 are the coefficients that we will mainly focus on. The detailed results of all of the coefficients and a visualization are available in the Appendix for interested readers. We plotted δ_1, δ_2 and δ_4 with 95% confidence intervals in Figure 7. As the figure shows, cheaters have an advantage of about 13 percentage points over non-cheaters on accessible pool size 1 problem generators. However, this advantage rapidly diminishes to around 3 percentage points when pool size is increased to 2, and it further diminishes to 2 percentage points at pool

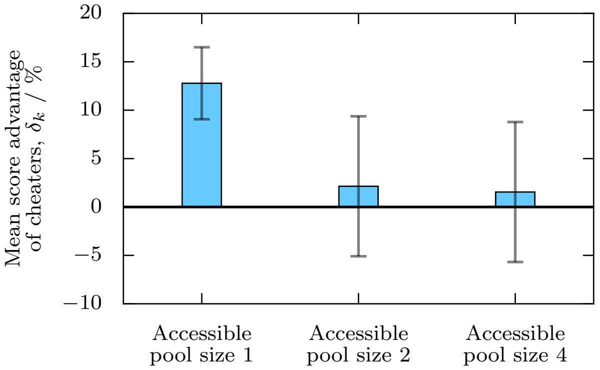

Figure 7. Regression coefficients that can be interpreted as cheaters' mean score advantage over non-cheaters on accessible pool size 1, 2 and 4 problem generators. The error bars correspond to 95% confidence intervals of the regression coefficients.

size 4. One subtlety to notice is that the mean score of non-cheaters on accessible pool size 2 problem generators is as high as 89 percentage points. This is due to random chance since the instructors of the course did not try to balance the difficulty of problem generators across different pools. The high mean score for pool size 2 suggests that the potential advantage of collaborative cheating is limited due to the ceiling effect [12], thus 3 percentage points is likely an underestimate of cheaters' advantage on pool size 2. We will discuss this issue in more detail in the Limitations section.

The above result suggests that increasing randomness in problem selection substantially helps to neutralize the advantage of collaborative cheating. Two hypotheses as to why it worked are: (1) larger pool sizes make it harder for cheaters to gather information about exam problems, and (2) cheaters have difficulty taking advantage of knowledge about the problems in larger pools, perhaps because it is hard to memorize solutions to more problems even if they maintain the same information advantage. We will explore hypothesis (1) in the next section.

ANALYSIS OF PROBLEM COVERAGE

To study the hypothesis that larger pool sizes make it harder to obtain complete information about exams, we introduce the notion of *coverage* as a measure of the amount of information a student might know. For exam x and student i, we compute student i's coverage over a category of problem generators q as follows:

$$\text{coverage}(x,i,q) = \frac{\ell_{x,i}}{|q|}, \tag{5}$$

where $|q|$ is the number of unique problem generators in q and $\ell_{x,i}$ is the number of unique problem generators among q that student i has practiced during student i's potential cheating period for exam x. For example, a student who achieves 50% coverage on the set of accessible pool size 4 problem generators for an exam has practiced half of these generators at least once during the student's potential cheating period of the exam. Note that this definition of coverage is somewhat orthogonal to our cheater classification: a student that only studies exam questions will be classified as a cheater even if the student only has partial information, and a student that studies every question equally (i.e., 100% coverage) will be classified as a non-cheater.

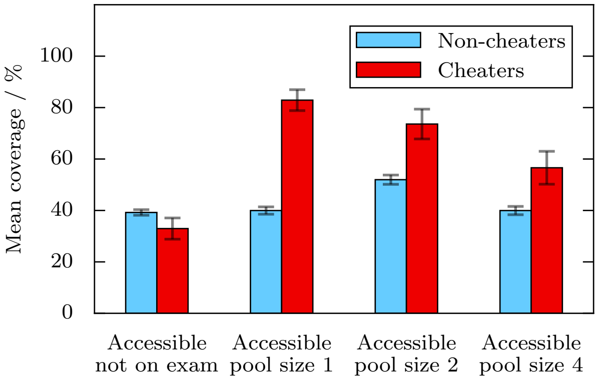

Figure 8. Mean coverage of each problem generator category for cheaters and non-cheaters. The error bars correspond to 95% confidence intervals of the means.

With the above definition of coverage, we derive a record of the form **(exam id, category, student id, coverage)** for every exam, student and category of problem generator. The coverage is a real number ranging from 0 to 100, representing the coverage percentage. The exam id, category, and student id are the same as defined previously. Since students cannot practice inaccessible problem generators, we will only study the last four categories of problem generators defined in the Exam data from the PrairieLearn section.

Naive method and result

With the same cheater and non-cheater labels used in the previous analysis, we computed the mean coverage of all the coverage records of each category of problem generator for cheaters and non-cheaters and plotted the results with 95% confidence intervals in Figure 8. As the figure shows, cheaters have lower mean coverage on accessible problem generators that are not on the exam, and they have higher mean coverage for accessible on-exam problem generators of every pool size. However, the difference in means between cheaters and non-cheaters decreases as pool size increases, suggesting that it is harder for cheaters to gather information about accessible on-exam problem generators with larger pool sizes.

As in the previous analysis of problem score advantages, this naive method probably underestimates the effect size because cheaters have a lower baseline coverage on accessible not-on-exam problem generators. We attempt to correct for this in the next section using a similar difference-in-differences analysis.

Difference in differences method

To obtain a good estimate of cheaters' coverage advantage and confidence interval, we again employed the difference in difference method. We treat accessible not-on-exam problem generators as a control condition and accessible problem generators with a specific pool size as an experimental condition. We will discuss limitations of this setup in the Limitations section. The linear regression we study thus be specified as follows:

$$\tilde{z} = \tilde{\alpha} + \tilde{\beta}\tilde{c} + \tilde{\gamma}_1\tilde{s}_1 + \tilde{\gamma}_2\tilde{s}_2 + \tilde{\gamma}_4\tilde{s}_4 + \tilde{\delta}_1\tilde{c}\tilde{s}_1 + \tilde{\delta}_2\tilde{c}\tilde{s}_2 + \tilde{\delta}_4\tilde{c}\tilde{s}_4, \tag{6}$$

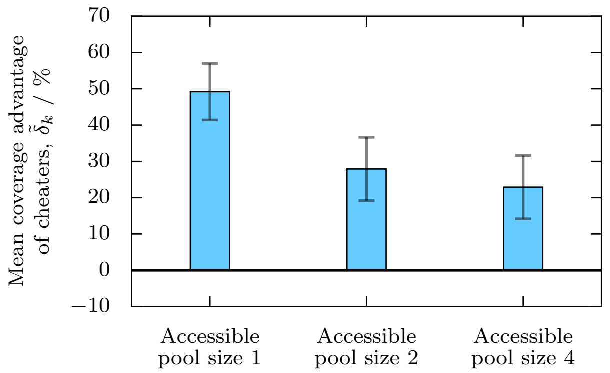

Figure 9. Regression coefficients that can be interpreted as cheaters' mean coverage advantage over non-cheaters on accessible pool size 1, 2 and 4 problem generators. The error bars correspond to 95 % confidence intervals of the regression coefficients.

where $\tilde{z}, \tilde{c}, \tilde{s}_1, \tilde{s}_2, \tilde{s}_4$ are observed values from each coverage record, defined as follows:

- $\tilde{z}$: the coverage of the record,
- $\tilde{c}$: 1 if the student associated with the record is classified as cheater on the exam, 0 otherwise,
- $\tilde{s}_k$: 1 if the category of problem generators associated with the record is accessible pool size k, 0 otherwise,

and $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{\gamma}_4, \tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_4$ are the coefficients that we want to compute, which can be interpreted as follows:

- $\tilde{\alpha}$: the mean coverage of non-cheaters on accessible not-on-exam problem generators,
- $\tilde{\beta}$: the mean coverage difference between cheaters and non-cheaters on accessible not-on-exam problem generators,
- $\tilde{\gamma}_k$: the mean coverage difference between accessible not-on-exam problem generators and accessible pool size k generators for non-cheaters,
- $\tilde{\delta}_k$: the additional mean coverage difference that cheaters have between accessible not-on-exam problem generators and accessible pool size k generators.

Difference in differences result and discussion

We will focus on $\tilde{\delta}_1$, $\tilde{\delta}_2$ and $\tilde{\delta}_4$ since we want to know how pool size affects cheaters' coverage advantage over non-cheaters. The detailed results of all of the coefficients and a visualization are available in the Appendix for interested readers. We plotted $\tilde{\delta}_1, \tilde{\delta}_2$ and $\tilde{\delta}_4$ with 95% confidence intervals in Figure 9. As the figure shows, cheaters' coverage of pool size 1 problem generators is 43 percentage points higher than that of non-cheaters. This advantage diminishes to around 22 percentage points when pool size is increased to 2 and it further diminishes to about 17 percentage points at pool size 4, although this is still significantly positive.

This suggests that the large reduction in cheaters' score advantage at the bigger pool sizes is partially due to the fact that cheaters have less complete information about exam problems drawn from larger pools (hypothesis 1), but it also seems that larger pools make it harder for cheaters to effectively utilize this advantage (hypothesis 2), perhaps because it is harder to memorize the increased number of problems.

LIMITATIONS

As the study in this paper is quasi-experimental using a pre-existing dataset, it has a number of limitations. We discuss seven specific limitations below. The first five are experimental limitations, and the final two relate to the generalizability of our results.

The first limitation relates to our method of cheater identification. We assumed that non-cheaters chose practice problem instances at random, resulting in a binomial distribution for the number of on-exam problem instances chosen. In fact, problem instance choices were probably somewhat correlated as students may have attempted several instances for the same problem generator in a row. To mitigate this, we chose a conservative cutoff of 0.0001 for identifying likely cheaters (i.e., $P(K_{x,i} \geq k_{x,i}) < 0.0001$). To test whether this is an important limitation we re-ran our analysis with a range of cutoff values from 0.000001 to 0.05. In all cases we found the same trend where increasing pool size reduces cheaters' exam score advantage and coverage advantage, though the exact numbers differed somewhat. From this we conclude that our analysis is not very sensitive to the details of our cheater identification scheme.

The second limitation draws from the issue introduced by the ceiling effect [12]. As observed in the analysis of exam problems score, the mean score of non-cheaters on accessible pool size 2 problem generators is about 89 percentage points, higher than for the other pool sizes. This high mean score limits the advantage cheaters could gain, because there is a score ceiling at 100%, and it is likely that we are thus underestimating the advantage of cheating with a pool size of 2. This limitation could potentially be addressed using nonlinear regression models designed to handle ceilings, such as the Tobit model [11], but we chose not to do so here to keep the analysis method relatively simple.

The third limitation is the appropriateness of the difference in differences method. The difference in differences method requires a parallel trend assumption, which states that if the attribute difference between the two groups has no impact on the measure under both the control and experimental conditions, then the result of the two groups between control condition and experimental condition should be parallel [1, 2]. While one could design an experiment to test the parallel trend assumption, we believe that it is likely to be a reasonable approximation because both cheaters and non-cheaters are broadly similar as evidenced by their similar behavior on control problems (e.g., scores on inaccessible exam problems and coverage on accessible-not-on-exam problems).

The fourth limitation relates to the extent that the inaccessible problem generators can be used as a control in the difference in differences analysis of score advantage. It is likely that cheaters also received some information about these problems, which may have helped them even though they could not see or practice the problems directly. This would lead to our analysis underestimating the effect size. However, it is also likely that cheaters under-prepared on problems that were not on the exam, leading them to be less prepared for the inaccessible problems and causing our analysis to overestimate the effect size. These issues could potentially be addressed by a followup study which adds more randomization to the inaccessible on-exam problem generators (choosing them from large pools) to provide a better control. Alternatively, a student-ability control could potentially be used as in Chen et al. [4]. Within the context of our current dataset, however, we can only note that this limitation introduces some additional uncertainty into the effect size estimates shown in Figure 7.

The fifth limitation involves the use of the accessible not-on-exam problem generators as a control for the difference in differences analysis of coverage. Because we identify cheaters as students who over-practice on-exam problems, and hence under-practice not-on-exam problems, our cheater coverage values for not-on-exam problems are likely too low. This means that we are probably overestimating the effect sizes shown in Figure 9. This limitation seems to be inherent to our cheater identification method, so we simply caution that the real cheater coverage advantage may be less than our estimates.

The sixth limitation is that our analysis focused on problem generators that are shared between exams and homeworks, and it is unclear how this would generalize to exam problems that are kept secret. However, these shared problems are likely to be the worst-case scenario for preventing collaborative cheating since it is easier for information producers to describe the problems they have seen on the exams, and it is easier for information consumers to practice the problems once they have obtained the information. This suggests that we are probably overestimating the score advantage.

The seventh limitation is that the asynchronous exams we examined are proctored and open only for a short period of time, and thus it is unclear how applicable our results are when asynchronous exams are not proctored or are perpetually open.

DISCUSSION AND CONCLUSION

In this paper we examined a dataset consisting of student homework records and asynchronous computerized exam records where some problems are shared between homeworks and exams. An asynchronous exam means that it was taken by students at different times over a multi-day period, opening up the possibility of collaborative cheating by earlier students giving information about the exam problems to later students.

We identified students who cheated collaboratively by observing whether they disproportionately studied homework problems that were also on the exam, after the earliest student had taken the exam. We found that 5.5% of students were classified as cheaters on average. With the cheaters identified, we studied how different degrees of randomized problem selection impacted collaborative cheating.

We found that increasing the size of the pool from which problem generators are selected lessened the score advantage of collaborative cheating. In particular, at a pool size of 4, we found the mean score advantage of collaborative cheating to be less than 2 percentage points (statistically indistinguishable from zero), down from an advantage of 13 percentage points for pool size 1 where every student had the same problem generator.

We hypothesize that the reduction in effectiveness of collaborative cheating with increasing random problem selection is partly due to cheaters' lack of complete information as the pool size increases. We showed that cheaters did have less information: their coverage advantage drops as pool size increases. However, the score advantage drops more than the coverage advantage, suggesting that it is also harder for cheaters to take advantage of studying with larger random pools, perhaps because there are more problems to remember.

The conditions of this study are likely to be the worst-case scenario for enabling collaborative cheating on asynchronous exams, since shared problems between exams and homeworks make it easier for students to study these problems if they have information about which ones are on the exam. This means that asynchronous exams which keep all exam problems secret and use random selection will probably see even less score advantage for students who attempt to cheat, with most likely zero or even negative benefits.

Our study suggests that randomization is an effective tool to discourage collaborative cheating and should be adopted as a common practice for asynchronous computerized exams. We note that two levels of randomization were used in our dataset. First, problem generators were used to produce random problem instances (e.g., varying the numbers in a problem), which already means that cheaters need to memorize algorithms to solve parameterized problems rather than simply remembering answers. Second, problem generators were randomly selected from a pool, so that cheaters need to memorize several times more algorithms if they are to have all the information about the exam. We believe that randomization will be effective both for in-person computerized exams, as studied here, and massive open online courses (MOOCs) where exams are almost always asynchronous and students copying answers using multiple accounts has been observed to be a serious issue due [9].

Our results also provide empirical evidence about how much randomization in problem selection is necessary. Our analysis suggests four is a good pool size for randomized problem selection to work well, although a pool size of three is also likely to be effective. Of course larger pool sizes will work better, provided that instructors can produce enough parameterized problems to fill the pools.

In fact, we posit that with sufficiently large pool sizes, one can eliminate the benefit of collaborative cheating by revealing to all students the specifics of the exam construction (e.g., which problem generators are part of which pools and making all of the problem generators public). With a large number of potential problems, memorizing problem-specific solution algorithms is both challenging with respect to short-term memory and less efficient than just learning the course material. It requires further research to investigate how large the pools would need to be to enable this kind of secret-free testing.

ACKNOWLEDGEMENTS

This work was partially supported by NSF DUE-1347722, NSF CMMI-1150490, and the College of Engineering at the University of Illinois at Urbana-Champaign under the Strategic Instructional Initiatives Program (SIIP).

REFERENCES

1. J. D. Angrist and J. Pischke. 2008. *Mostly harmless econometrics: An empiricist's companion*. Princeton University Press.

2. J. D Angrist and J. Pischke. 2014. *Mastering'metrics: The path from cause to effect*. Princeton University Press.

3. A. C. Bugbee Jr. 1996. The equivalence of paper-and-pencil and computer-based testing. *Journal of Research on Computing in Education* 28, 3 (1996), 282–299.

4. B. Chen, M. West, and C. Zilles. 2017. Do performance trends suggest wide-spread collaborative cheating on asynchronous exams?. In *Learning at Scale 2017*.

5. R. F. DeMara, N. Khoshavi, S. D. Pyle, J. Edison, R. Hartshorne, B. Chen, and M. Georgiopoulos. 2016. Redesigning computer engineering gateway courses using a novel remediation hierarchy. In *2016 ASEE Annual Conference & Exposition*. New Orleans, Louisiana.

6. E. Lee, N. Garg, C. Bygrave, J. Mahar, and V. Mishra. 2015. Can university exams be shortened? An alternative to problematic traditional methodological approaches. In *Proceedings of the 14th European Conference on Research Methods*.

7. D. L. McCabe. 2005. Cheating among college and university students: A North American perspective. *International Journal for Educational Integrity* 1, 1 (2005).

8. R. Muldoon. 2012. Is it time to ditch the traditional university exam? *Higher Education Research and Development* 31, 2 (2012), 263–265.

9. C. G. Northcutt, A. D. Ho, and I. L. Chuang. 2016. Detecting and preventing "multiple-account" cheating in massive open online courses. *Computers & Education* 100 (2016), 71–80.

10. C. G. Parshall. 2002. *Practical considerations in computer-based testing*. Springer Science & Business Media.

11. W. Schnedler. 2005. Likelihood estimation for censored random vectors. *Econometric Reviews* 24, 2 (2005), 195–217.

12. L. Wang, Z. Zhang, J. J. McArdle, and T. A. Salthouse. 2008. Investigating ceiling effects in longitudinal data analysis. *Multivariate Behavioral Research* 43, 3 (2008), 476–496.

13. G. Watson and J. Sottile. 2010. Cheating in the digital age: Do students cheat more in online courses? *Online Journal of Distance Learning Administration* 13, 1 (2010).

14. Matthew West, Geoffrey L. Herman, and Craig Zilles. 2015. PrairieLearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. In *2015 ASEE Annual Conference & Exposition*. Seattle, Washington.

15. Craig Zilles, Robert Timothy Deloatch, Jacob Bailey, Bhuwan B. Khattar, Wade Fagen, Cinda Heeren, David Mussulman, and Matthew West. 2015. Computerized testing: A vision and initial experiences. In *2015 ASEE Annual Conference & Exposition*. Seattle, Washington.

16. C. Zilles, M. West, and D. Mussulman. 2016. Student behavior in selecting an exam time in a Computer-Based Testing Facility. In *2016 ASEE Annual Conference & Exposition*. New Orleans, Louisiana.

APPENDIX

The regression coefficients for the score regression and coverage regression and their associated statistics are shown in Tables 1 and 2, respectively.

Visualizations of the score and coverage regressions are shown in Figures 10 and 11, respectively. The solid line segments in the figures correspond to the non-cheaters while the dashed line segments correspond to the cheaters. The dotted line segments are what cheaters would get under the parallel trend assumption. Each vertical dotted line segment indicates what the coefficient is measuring, and negative signs indicate that the coefficient is actually negative.

Coefficient	Value	95% CI		*p*-value
α	79.449	78.892	80.006	0.000
β	−1.814	−4.286	0.659	0.150
γ_1	−4.206	−5.076	−3.336	0.000
γ_2	9.105	7.456	10.755	0.000
γ_4	0.097	−1.553	1.747	0.908
δ_1	12.784	9.062	16.506	0.000
δ_2	2.139	−5.092	9.371	0.562
δ_4	1.548	−5.684	8.779	0.675

Table 1. Coefficients for regression on score and their 95% confidence intervals computed with ordinary least square. The p-value corresponds to the probability of the coefficient being 0 in a two tailed test.

Coefficient	Value	95% CI		*p*-value
$\tilde{\alpha}$	39.219	37.997	40.441	0.000
$\tilde{\beta}$	−6.265	−11.567	−0.963	0.021
$\tilde{\gamma}_1$	0.745	−1.094	2.584	0.427
$\tilde{\gamma}_2$	12.743	10.742	14.744	0.000
$\tilde{\gamma}_4$	0.739	−1.262	2.740	0.469
$\tilde{\delta}_1$	49.198	41.401	56.994	0.000
$\tilde{\delta}_2$	27.903	19.174	36.632	0.000
$\tilde{\delta}_4$	22.907	14.178	31.636	0.000

Table 2. Coefficients for regression on coverage and their confidence intervals computed with ordinary least square. The p-value corresponds to the probability of the coefficient being 0 in a two tailed test.

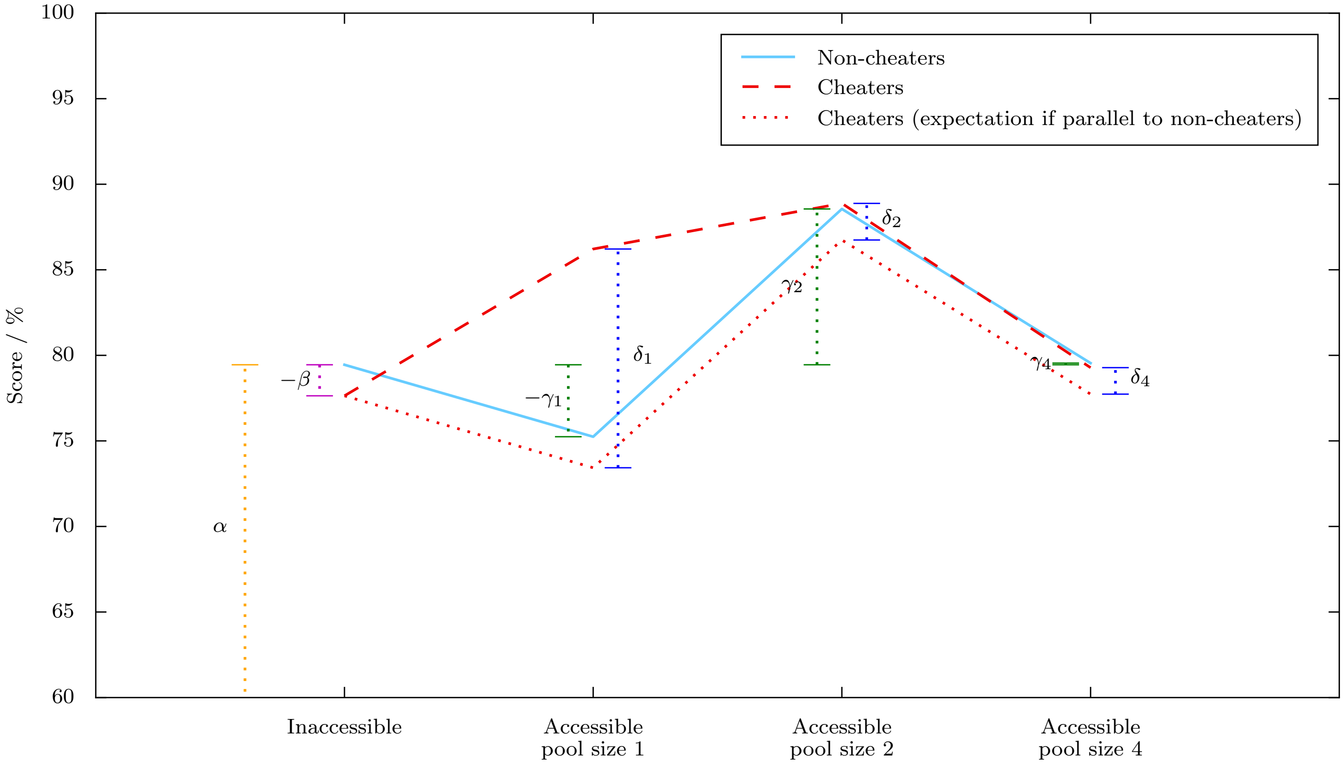

Figure 10. Line graph counterpart of Figure 5 with visualizations showing what each coefficient measures. Negative signs indicate that the coefficient is the negative of the length of the corresponding vertical dotted line segment.

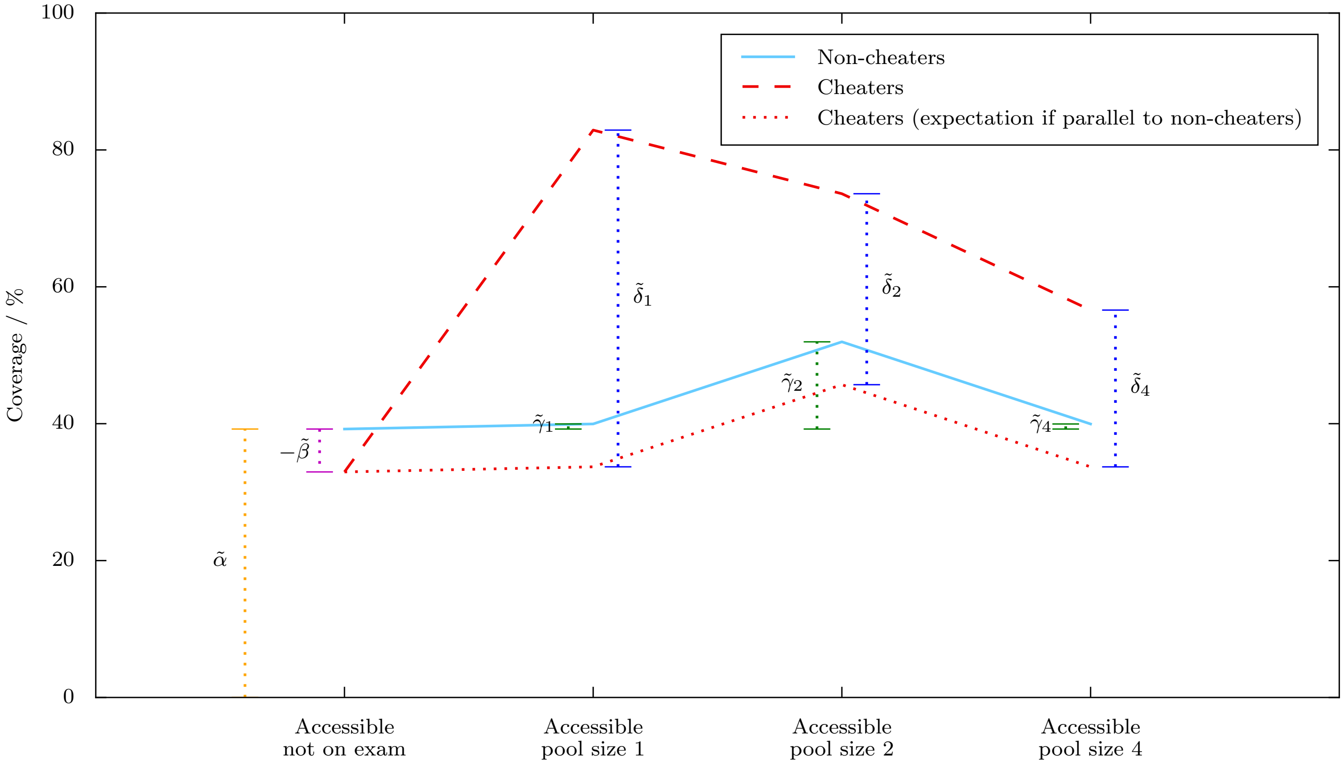

Figure 11. Line graph counterpart of Figure 8 with visualizations showing what each coefficient measures. Negative signs indicate that the coefficient is the negative of the length of the corresponding vertical dotted line segment.