Journal of Universal Computer Science, vol. 19, no. 15 (2013), 2173-2197
submitted: 9/6/13, accepted: 30/8/13, appeared: 1/9/13 © J.UCS

An Algorithm for Peer Review Matching in Massive Courses for Minimising Students' Frustration

Iria Estévez-Ayres, Raquel M. Crespo-García, Jesús A. Fisteus
Carlos Delgado Kloos
(Dpto. Ingeniería Telemática, Universidad Carlos III de Madrid, Spain
{ayres, rcrespo, jaf, cdk}@it.uc3m.es)

Abstract: Traditional pedagogical approaches are no longer sufficient to cope with the increasing challenges of Massive Open On-line Courses (MOOCs). Consequently, it is necessary to explore new paradigms. This paper describes an exploration of the adaptation of the peer review methodology for its application to MOOCs. Its main goal is to minimise the students' frustration through the reduction of the number of committed students that receive no feedback from their peers. In order to achieve this objective, we propose two algorithms for the peer review matching in MOOCs. Both reward committed students by prioritising the review of their submissions. The first algorithm uses sliding deadlines to minimise the probability of a submission not being reviewed. Our experiments show that it reduces dramatically the number of submissions from committed students that do not receive any review. The second algorithm is a simplification of the former. It is easier to implement and, despite performing worse than the first one, it also improves with respect to the baseline.

Key Words: Massive Open Online Courses, peer review, peer assessment, peer evaluation, evaluation, assessment, quality
Category: K.3, K.3.1

1 Introduction

Despite their short history, Massive Open Online Courses (MOOCs) have experienced an incredible growth [Hyman 2012]. Based on their underlying pedagogical approach, MOOCs can be classified into two main trends [Rodriguez 2012]: xMOOCs follow a content-based approach, thus applying an underlying cognitive-behaviourist pedagogy (with some small components from social constructivism); cMOOCs apply a connectivist approach instead, based on connecting learners [Siemens 2005].

In either case, MOOCs rely heavily on the active work of the student, not only as a content consumer but completing exercises and tasks as well. In consequence, evaluation is an intrinsic requirement for this type of courses, even if used just for formative purposes. It also constitutes one of their greatest vulnerabilities, as their massive audience makes it impossible for the teacher to provide the students with appropriate feedback.

Several strategies have been tried to address this issue. Applying automatic grading is the most common and scalable solution. It fits perfectly the massive audience of MOOCs. Alternative solutions rely on the students themselves to evaluate submissions. Peer assessment emphasises the social dimension of MOOCs and allows students not only to receive feedback on their submissions, but also to explore alternative solutions and acquire complementary competencies such as critical thinking or evaluation skills.

Peer review has been receiving increasing attention in the educational context, being widely used in a variety of settings. Although its benefits are strongly supported by empirical evidence in the literature, controversy remains about its reliability, quality and validity. Due to that, its application raises reluctance of both teachers and students.

Two main critics are typically done to the peer review methodology. One is the potential lack of quality of peers' feedback and lack of reliability of peers' grades. The other is the risk of reviewers failing to submit their revisions. In the context of MOOCs, with an extremely low participation rate, this latter risk scales-up and can be an important source of frustration for the students waiting for feedback.

A number of studies in the literature focus on detecting and analysing the effect of emotions on learning [Kort et al. 2001; Baker et al. 2010]. Motivation, enjoyability or entertainment have demonstrated a positive influence [Ebner and Holzinger 2007]. Confusion has also been reported to be positively correlated with learning [Craig et al. 2004; Graesser et al. 2007], which is consistent with the constructivist theory about deep learning associated to cognitive conflict. Regarding frustration, it is generally considered negative and some research has focused on reducing it [Hone 2006; Klein et al. 2002; McQuiggan et al. 2007].

The massive audience makes MOOCs particularly sensitive to problems, because even the simplest problems may scale-up and affect a large community. For any methodology to succeed in such a demanding context, its potential risks must be carefully addressed and tackled to avoid a negative impact on the course progress. In this paper, we focus on the risk of reviewers failing to submit their revisions as one of the biggest problems that jeopardise the successful application of peer review in MOOCs. We propose a solution for reducing, and ideally eliminating, students' frustration due to not receiving the expected feedback from their peers. Concretely, we propose to act on the matching of submissions and reviewers, taking into account students' commitment, and allowing the reassignment of non-reviewed submissions, so that potential missing reviews have a minimum impact on students' expectations and satisfaction.

This paper is organised as follows: Section 2 reviews the related literature and state of the art; Section 3 presents the proposed peer review matching algorithms; Section 4 evaluates the performance of those algorithms and compares them with a baseline algorithm; Section 5 discusses the main conclusions of the evaluation; and, finally, Section 6 concludes and presents the future work.

2 State of the art

The MOOC concept emerged in 2008 with the "Connectivism and Connective Knowledge" course taught by George Siemens and Stephen Downes. However, it is in 2011 when this methodology experienced an unprecedented growth. That year, Peter Norvig and Sebastian Thrun's "Introduction to Artificial Intelligence" class reached more than 150,000 students [DiSalvio 2012], initiating the current MOOC hype [Hyman 2012]. In the last years, with particular emphasis since 2012, top universities have offered hundreds of MOOCs based on new technological platforms like Udacity[1], Coursera[2], edX[3] or MiriadaX[4]. In fact, MOOCs are considered one of the main educational trends in the last months [Daniel 2012; Hyman 2012; Alario-Hoyos et al. 2013].

MOOCs have their roots in the OER (Open Educational Resources) movement and expand e-learning to reach a wide, massive audience. As main characteristics, MOOCs are *open* (students can enrol with no prerequisites), *free*, *online*, intend to reach a *massive* number of students (usually thousands, although there are cases of more than one hundred thousand enrolled students), and usually present a high student per teacher ratio. These characteristics introduce some drawbacks, such as disparate students, small proportion of active students, high attrition rates, and scarce teacher support during enactment [Daniel 2012; Clow 2013; Downes 2010; Kop et al. 2011].

MOOC participants are expected to form a community of learners that support each other and enrich the course with discussions and related contents (crowdsourcing) [Mackness et al. 2010; McAuley et al. 2010]. This is called the "learner as teacher as learner" model [Siemens 2006]. Consequently, MOOCs constitute an ideal environment for *social learning* methodologies, whose importance and impact have dramatically risen since the irruption of the Web 2.0 [Holzinger et al. 2009; Ebner et al. 2006].

Whereas MOOC learners are expected to play the leading role in the learning process, the instructor plays a secondary role. In contrast to traditional e-learning courses, instructors' activity focuses on the design of the course but fades out during enactment. Due to the massive number of students (and the free nature of the course), it is impossible to provide personalised support by teachers [Downes 2010; Kop et al. 2011]. In consequence, MOOCs require alternative assessment methods, which do not depend on the teacher intervention.

[1] www.udacity.com

[2] www.coursera.org

[3] www.edx.org

[4] www.miriadax.net

2.1 Assessment in MOOCs

As emphasised by Sandeen [2013], “within the MOOC world, assessment is a central feature of design from the very beginning. In this new context, assessment is less about compliance than about supporting student learning outcomes and ultimately student success and attainment”. However, Balfour [2013] notes that “the time an instructor spends teaching and evaluating work per student is very low in high enrolment MOOCs”.

Two main strategies have been tried to address this issue. First, automatic grading provides the necessary scalability for MOOCs. Alternatively, students themselves can assume the role of evaluators and provide their peers with feedback, emphasising the social dimension of MOOCs.

Both assessment mechanisms are reported to have a relatively high degree of acceptance by faculty [Sandeen 2013]. According to a recent survey of MOOC faculty conducted by The Chronicle of Higher Education, 74% of respondents used automated grading. Of them, 67.1% found the technique to be “very reliable” and 30.1% found it to be “somewhat reliable”. Thirty-four percent (34%) of respondents used peer grading. Of them, 25.8% found the technique to be “very reliable” and 71% found it to be “somewhat reliable” [Kolowich 2013].

2.2 Automatic grading

The most common and scalable solution is applying automatic grading, which fits perfectly the massive audience of MOOCs. Consequently, MOOCs rely heavily on closed assignments, like multiple choice questions, formulaic problems with correct answers, logical proofs, computer code, and vocabulary activities.

Automatic grading has been successfully applied to multiple choice questions in MOOCs; for example, in the Stanford “Introduction to Artificial Intelligence” course. It has also been applied to coding assignments, with more controversial results; for example, the “HTML5 Game Development” MOOC (Udacity) automatic grading system for coding exercises raised students’ complaints about malfunctioning issues and provoked delays in the course.

There is intense research on automatic grading of open assignments, and scoring and providing feedback on written assignments in MOOCs has been the subject of a number of recent news articles [Balfour 2013]. Automatic Essay Scoring (AES) applications usually apply statistical models for predicting human-assigned scores based on features of essays that have been determined empirically or statistically to correlate with the ways humans rate those essays, as explained by Balfour [2013]. A detailed review on AES mechanisms can be found in [Shermis et al. 2010]. In the MOOC context, EdX has announced that it will use automated essay scoring [Markoff 2013]. Additionally, Balfour [2013] lists three long-standing commercial AES applications that have been tested and are established in the academic literature [Shermis et al. 2010].

Evaluation of AES in the literature reports high correlation with human scores [Attali 2007; Shermis et al. 2010], usually for assignments that fulfil specific requirements. However, there are limitations and these applications fail in more complex contexts in which the essays and topics to grade are not homogeneous [Graesser and McNamara 2012]. Concern remains on AES being limited to a superficial evaluation [NCTE 2013].

2.3 Peer review

Peer review, defined in the educational context as "an arrangement in which individuals consider the amount, level, value, worth, quality, or success of the products or outcomes of learning of peers of similar status" [Topping 1998], has been commonly applied in a widespread range of learning settings [Topping 1998; van Zundert et al. 2010]. Despite its increasing popularity [Topping 2005; van den Berg et al. 2006; van Zundert et al. 2010] and the positive effects documented in the literature [Topping 1998; Falchicov 1995; 1996; Dochy et al. 1999; Topping 2003], there is still controversy about its reliability, effectiveness, and malfunction issues [Nilson 2003]. Literature reviews that analyse the experimental results of applying peer review in the educational context [Topping 1998; Nilson 2003; van Zundert et al. 2010] report mostly beneficial effects. However, they also mention some negative issues, which are usually due to two main reasons: firstly, a correct application of the methodology is not easy to achieve; and secondly, there are some requirements for the process to be effective that are not always taken into account. Some care has to be taken in the application of peer review in order to benefit from its potential advantages, as noted by van der Pol et al. [2008]. Several authors propose measures to improve the quality and positive effects of the process, such as an adequate organisation [van den Berg et al. 2006], the convenience of training the peer assessors [Nilson 2003; Robinson 2001; Russell 2004] or using rubrics [Jonsson and Svingby 2007].

Two main critics are typically done to peer review:

– One is the potential lack of quality of peers' feedback and lack of reliability of peers' grades (particularly if used for official assessment) This problem is tackled in [Robinson 2001; Russell 2004; Goldin 2012; Piech et al. 2013], among others.

– The other is the risk of reviewers failing to submit their revisions. Despite its incidence, little effort has been devoted to alleviate this issue. A remarkable exception is PG [Gehringer and Cui 2002].

In the context of MOOCs, with a low participation rate, the second risk (peer assessors failing to submit their revisions) scales-up and can be an important source of frustration for the students waiting for feedback.

2.4 Conclusions

The particular characteristics of MOOCs require new pedagogical approaches. In particular, assessment poses specific challenges. Either for summative or formative purposes, intense research is currently being devoted to improve the assessment methods quality, reliability, scalability and applicability in the MOOC context.

Proposed solutions for summative evaluation (credits and certifications) imply a radical change in the intrinsic features of MOOCs, with controlled, secure environments and requiring the payment of fees [Gupta and Sambyal 2013]. On the contrary, formative assessment methods need to be smoothly integrated in the MOOC setting, accepting the intrinsic conditions associated to such courses, for students to achieve the intended learning outcomes. The massive number of students, together with the consequent lack of teacher support, make the MOOC community turn to automatic grading and peer evaluation alternatives, each of them having pros and cons.

In this paper, we focus on the risk of reviewers failing to submit their revisions as one of the biggest problems that jeopardise the successful application of peer review in MOOCs. We propose a solution for reducing, and ideally eliminating, students' frustration due to not receiving the expected feedback from their peers. It is based on some improvements to the algorithm that selects the reviewers for each submission.

3 Algorithm for assigning reviewers

Students that do not review the submissions that the system assigns to them pose an important challenge to the application of peer review in MOOCs. They can provoke that some students do not get feedback for their submissions. The typical solution to this problem consists in augmenting the number of reviews that each student must fulfil. However, this increases the workload of the students.

We tackle this problem in a different way. Our main goal is minimising the number of *committed students* that do not receive any review for their submissions. By committed students we mean those that are actively involved in the review process. We believe that they are the ones that actually expect and deserve receiving feedback for their submissions. In addition, as a secondary goal, we try to reduce the required workload of the students for achieving the main objective.

In order to do that, our algorithm assigns submissions only to students that volunteer to review, as we expect volunteers to be more reliable at completing their revisions. In order to receive reviews for their own assignments, students must volunteer and complete the reviews the system assigns to them. On top of that, we introduce two main mechanisms: the sliding deadlines mechanism and the commitment rewarding mechanism. The sliding deadlines mechanism, instead of applying the same global deadline for all the reviews, sets a specific (shorter) deadline for each review, relative to the instant it is assigned to the student. If s/he fails to meet this deadline, the system has still time to assign that review to another student. The commitment rewarding mechanism encourages students to volunteer and complete their reviews by giving their submissions priority when there are less available reviewers than assignments to review.

In this work we propose two algorithms. The *Sliding Deadline Commitment-Rewarding peer matching algorithm* (SDCR) implements the two mechanisms explained above. Since implementing sliding deadlines may be technically or logistically cumbersome in some scenarios, we propose also a simplification, called the *Fixed Deadline Commitment-Rewarding peer matching algorithm* (FDCR), that implements the commitment rewarding mechanism alone, with the conventional fixed deadlines instead of sliding deadlines. The rest of this section explains the two mechanisms in depth.

3.1 The sliding deadlines mechanism

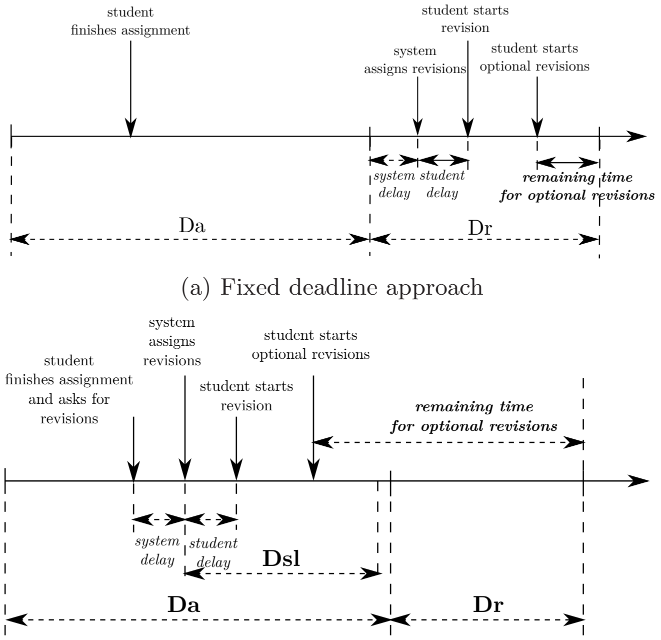
The approach followed by most MOOC platforms is to define only two fixed deadlines (see Figure 1(a)): the assignment deadline (D_a) and the global review deadline (D_r). Students cannot start reviewing until the assignment deadline expires. This practice simplifies the process workflow and allows students to resubmit their work until the assignment deadline, while preventing them from accessing their peers' solutions before having submitted their own work.

In order to reduce the number of assignments that get no revisions, these platforms resort to offering the student the possibility of performing more revisions than the mandatory. However, the deadline for the optional revisions is the same as for the mandatory ones (see Figure 1(a)), which means that additional reviews imply less time available per review.

Since many students will finish their assignments before the assignment deadline, we propose to use this fact to offer the students that are willing to perform optional revisions more time to complete them, by setting a separate sliding deadline for each student (see Figure 1(b)).

The first step consists in asking the student whether s/he wishes to start the review process. By doing so, the algorithm classifies the students into *reviewers* and *non-reviewers*. The algorithm aims to motivate the students to participate in the process by rewarding the *reviewers*: only reviewers will receive feedback about their work. Students that volunteer to perform revisions can no longer re-submit their work.

Once the student volunteers, s/he waits for the system to assign her/him the N_r mandatory submissions to evaluate. When the matching is made, a sliding deadline (D_{sl} in the figure) is set for this student to complete those mandatory



(a) Fixed deadline approach

(b) Sliding deadline approach

Figure 1: Deadlines management approaches for the peer review process.

revisions. The duration of D_{sl} is the same for every student, though each one will have a different starting point and, consequently, a separate due date.

When the student finishes her/his N_r mandatory revisions, s/he can ask for optional ones. The algorithm will assign her/him N_r more, but s/he will have all the remaining time until the global review deadline (D_r) to submit them. The student can ask for optional revisions only once.

As shown in Figure 1, the main differences from the point of view of the students are that: they can start their mandatory revisions before (once they complete their assignment), and they have more time for doing optional revisions.

For students that complete their assignment close to the expiration of the assignment deadline (D_a in the figure), our algorithm behaves, from their point of view, like the algorithm that uses fixed deadlines. The same happens if the course manager decides to postpone the assignment of submissions to review for all the students until the assignment deadline.

However, introducing sliding deadlines allows the system to designate another reviewer if the original one fails to complete a mandatory revision. Once the sliding deadline of a student expires, all the assignments not reviewed by this student return to the pool of submissions to be revised, as depicted in Figure 2.

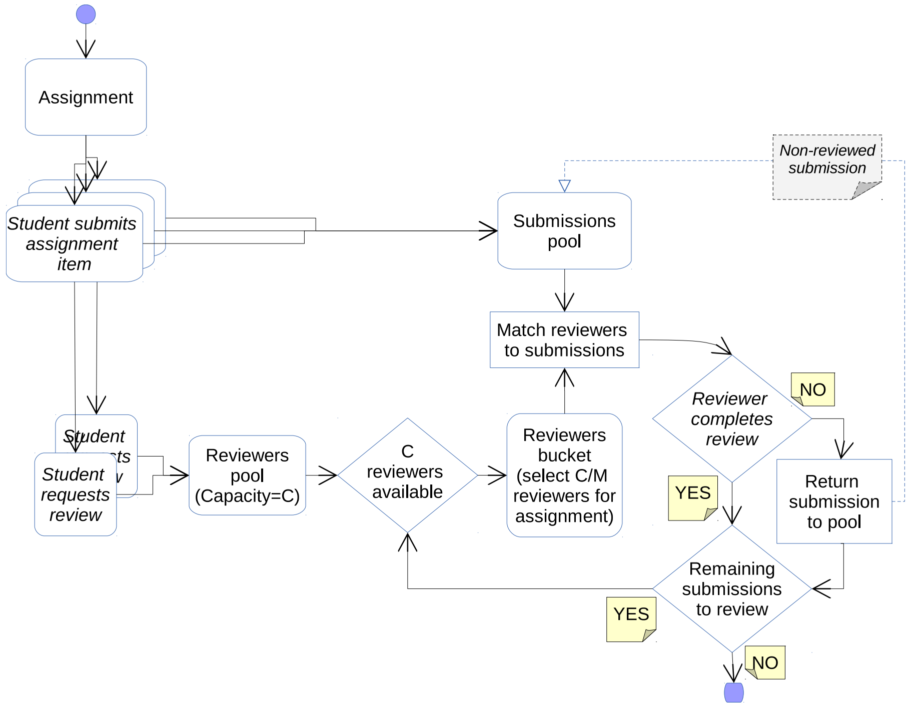

Assignment
Student submits assignment item
Submissions pool
Non-reviewed submission
Match reviewers to submissions
Reviewer completes review
NO
YES
Return submission to pool
Student requests review
Reviewers pool (Capacity=C)
C reviewers available
Reviewers bucket (select C/M reviewers for assignment)
Remaining submissions to review
YES
NO

Figure 2: Assignment of mandatory revisions

Performing the submission-reviewer matching each time a student requests to review would be inefficient, mainly due to the huge number of students in a MOOC. Thus, it is postponed until having C available *first-time reviewers* (students waiting for their mandatory revisions).

In order to avoid the problem of leaving the students that finish (or decide to review) close to the deadline with fewer revisions or no revisions at all, we decided to match only a fraction $(1/M)$ of reviewers. This also increases the degrees of freedom in future allocations, in order to prevent deadlocks and violation of restrictions (e.g. a student having to review his/her own work). Thus, instead of matching all the C available reviewers, the algorithm randomly selects a subset of size C/M of available reviewers, leaving the other $C - C/M$ free for future allocation (as depicted in Figure 3). Notice that the algorithm is triggered every time the pool of first-time reviewers reaches C reviewers. The requests from students waiting for optional revisions are processed also at those instants.

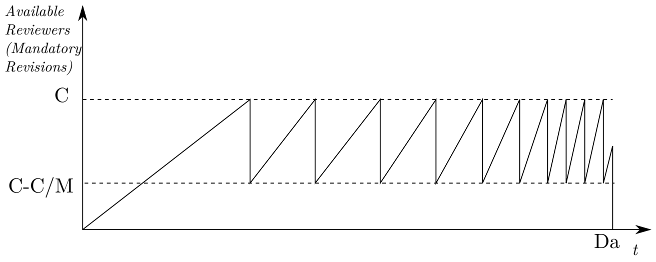


Figure 3: Available reviewers for performing mandatory revisions

3.2 The commitment rewarding mechanism

The *commitment rewarding mechanism* is applied when assigning optional revisions. For each optional request, the algorithm assigns N_r submissions giving priority to the *reviewers'* submissions over the *non-reviewers'* ones. This way committed students are rewarded. The list of candidate assignments to be reviewed is built as follows:

1. Select all the submissions authored by *reviewers* that are not yet assigned to any reviewer.
2. If the length of the list is shorter than N_r, concatenate already assigned *reviewers'* submissions that do not have any revision yet, provided that they are assigned to less than a half of N_r reviewers.
3. If the length of the list is shorter than N_r, concatenate the remaining already assigned *reviewers'* submissions that do not have any revision yet.
4. If the length of the list is shorter than N_r, concatenate the remaining *reviewers'* submissions.
5. If the length of the list is shorter than N_r, repeat steps 1 to 4 for *non-reviewers*, but only if D_a has not expired.

Once the list of candidate submissions has been built, N_r are randomly selected to be reviewed by the student requesting optional revisions, and notified to her/him.

Different policies can be applied to select reviewers. For example, some policies could avoid assigning always *similar* students, or students with the same ranking. Since studying those policies is out of the scope of this paper, the analysis presented in Section 4 assumes a random submission-reviewer matching every time the C/M reviewers are assigned the submissions to revise. Alternative matching criteria could be applied in order to enrich this algorithm.

4 Evaluation

In this section, the performance and behaviour of the two submission-reviewer matching algorithms we propose (*SDCR* and *FDCR*) is evaluated and compared to the *Baseline* peer matching algorithm (with no enhancements at all) via simulation with a full factorial experimental design. The simulator used to perform the experiments was based on SimPy[5], a process-based discrete event simulation framework for Python.

Section 4.1 justifies the selection of a simulation-based methodology for objectively measuring the impact of the algorithms. Section 4.2 details the experimental design. Finally, Section 4.3 presents the results of the evaluation.

4.1 Methodology

As reminded by [Cohen et al. 2007], "simulations have been used in the natural sciences and economic forecasting for several decades". In the educational context, simulation is used in different research areas, including studying group learning and analysing the effects of different goal structures on individuals and groups of learners [Spoelstra and Sklar 2008], educational change [Ridgway 1998], school effectiveness [Tymms 1996] and, in general, understanding education systems.

Among the main advantages associated to simulations by Bailey [1994] (economy, visibility, control and safety), it is control the one that makes them suitable for this case. Computer simulations are especially useful for evaluating our matching algorithms because they enable the researcher to control and manipulate variables and components, thus being particularly appropriate for testing alternative configurations and effects [Cohen et al. 2007]. In this work, simulations are the most adequate evaluation methodology because they allow replicating a complex context, and abstracting the intrinsic variety of real-world settings. It would be impossible to replicate similar conditions for comparing the alternative solutions in real-world MOOCs: having similar number of participants, with similar motivation, behaviour, personal restrictions, participation and attrition rates, etc. Such contextual factors would have a significant influence on the experimental results, disguising the algorithm outcomes.

4.2 Experimental Design

The experiments were designed to assess the influence of several parameters, including those related to the student's behaviour and other parameters defined by the context (the assignment, the platform, etc.) Section 4.2.1 details those parameters and 4.2.2 defines the performance metrics we used.

[5] `simpy.sourceforge.net`

A full factorial experimental design has been applied to evaluate the algorithms. In order to reduce the bias due to the chosen seed, we performed $M = 256$ executions of the simulation for each experimental setting, and computed the mean values of their results. A *successful execution* is defined as an execution with at least 5 students willing to review. However, the results for a given setting were discarded if less than $m = 5$ of its executions were successful, in order to avoid potentially unreliable mean values. We adjusted the values of M and m after testing the simulator with a battery of preliminary simulations.

4.2.1 Problem parameters

Table 1 summarises the values and ranges of the problem parameters we analysed in the experiments:

- *System-dependent parameters*: parameters set by the instructor when s/he defines a peer review assignment, or fixed beforehand by the platform (grey rows in Table 1).
- *Student's behaviour-dependent parameters*: parameters depending on the behaviour of the students (white-background rows in Table 1).

As the evaluation is based on simulations, instead of data collected from real courses, students' behaviour is simulated with several probabilistic variables. Specifically, the simulation used left-truncated normal distributions (truncated on the origin) to model the temporal behaviour of each student regarding assignments or revisions (i.e. how long it takes for them to complete the task); and Bernouilli distributions to model the behaviour of each student when faced to a decision, such as starting an assignment, engaging with the reviewing process or being willing to review additional peer submissions.

4.2.2 Performance metrics

Our algorithms aim to favour committed students at the expense of non committed ones. Thus, we collected separate performance metrics for the groups of reviewer students and the group of non-reviewers. In particular, our analysis focuses on the effect of the above parameters on the following outcomes:

1. Percentage of students that receive no revisions: for those who just submitted the assignment, and for those participating in the peer review process as reviewers.
2. Number of total revisions received by each student: for those who just submitted their assignment, and for those who participated in the peer review process as reviewers.

Parameter	Values	Meaning
D_a	15	Global assignment deadline to submit the assignment within the platform.
D_r	7	Global revision deadline, when the whole peer review process finishes.
N_r	$\{1, 3, 5\}$	Number of mandatory revisions.
D_{sl}	$\{4, 5.5, 7\}$	Sliding deadline to perform the mandatory revisions.
C	$\{0.25, 0.5, 1, 2\}$	Capacity of the pool of available reviewers, expressed as a percentage (%) of the total number of students. When this pool is full triggers the execution of the matching algorithm.
M	$\{1/2, 1/3\}$	Fraction of the pool of reviewers to assign for each execution of the peer assignment algorithm.
N	$\{1000, 5000, 10000, 25000\}$	Number of students enrolled in the course.
p_a	$\{0.05, 0.5, 0.15, 0.20\}$	Probability of a student starting an assignment.
p_r	$\{0.25, 0.5, 0.75\}$	Probability of a student starting the review task.
p_{mr}	$\{0.25, 0.5, 0.75, 1\}$	Probability of an student willing to review additional assignments.
μ_a	$\{5.5, 7, 12.5\}$	Mean of the truncated normal distribution of the time spent by each student for completing an assignment.
σ_a^2	1	Variance of the truncated normal distribution of the time spent by each student for completing an assignment.
μ_d	$D_{sl}/2$	Mean of the truncated normal distribution of the delay of each student for starting the review task.
σ_d^2	1	Variance of the truncated normal distribution of the delay of each student for starting the review task.
μ_r	$\{0.5, 1\}$	Mean of the truncated normal distribution of the time spent by each student for completing the review of each assigned submission.
σ_r^2	1	Variance of the truncated normal distribution of the time spent by each student for completing the review of each assigned submission.

Table 1: Values and ranges for the problem parameters used in the experiments (all time values are in days)

4.3 Results

A total of 397, 440 successful executions of the peer review process simulation have been performed. Specifically, 72, 572 scenarios for the *Sliding Deadline Commitment-Rewarding* (SDCR) peer matching algorithm were evaluated. Out of these scenarios, only 3, 452 were evaluated for the *Fixed Deadline Commitment-Rewarding peer matching algorithm* (FDCR) and 3, 452 for the Baseline algorithm. The simulation of the SDCR algorithm requires a higher number of scenarios because it has specific parameters (such as the capacity of the reviewers pool or the duration of the sliding period) that have no sense (nor impact) for the

Number of students Algorithm	1000	5000	10000	25000	Total
SDCR	10367	20735	20735	20735	72572
FDCR	863	863	863	863	3452
baseline	863	863	863	863	3452

Table 2: Number of scenarios evaluated per algorithm per number of students

FDCR or the Baseline algorithms. Thus the number of different combinations of parameters is greater for the SDCR algorithm.

Table 2 shows the scenarios evaluated for each algorithm and number of enrolled students. The lower number of scenarios for 1000 students in the SDCR algorithm is due to the impact of the range of C on calculating the capacity of the pool of reviewers. For $C = 0.25\%$, the capacity of the pool is 2 students and we decided to not execute simulations with a capacity of the pool of reviewers lower than 10 students. However, the range makes sense for 10000 students, where the capacity of the pool with this value of C is 25 students.

4.3.1 Performance of the SDCR algorithm. Effect of the sliding deadline and the number of revisions

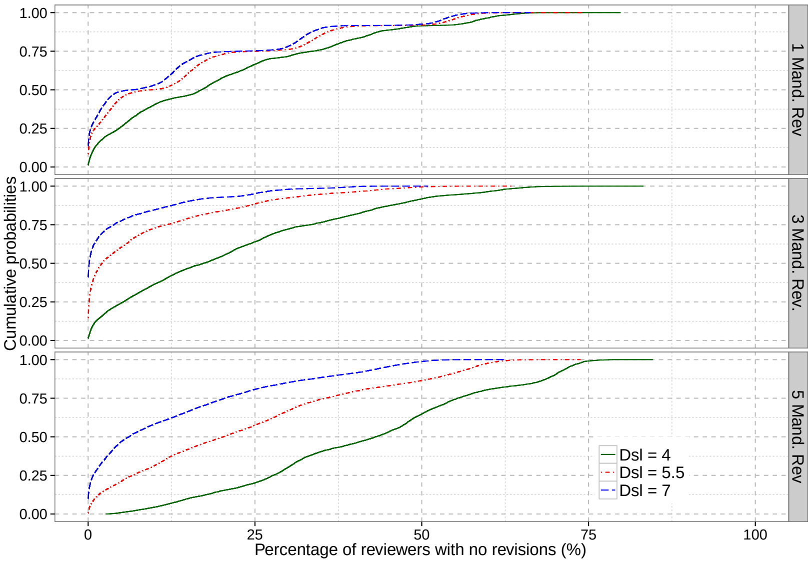
The evaluation metric applied for analysing the performance of the SDCR is the quantity (percentage) of students that receive no revision for their work. Figure 4 shows the cumulative distribution function (CDF) of this percentage for reviewers (Figure 4(a)) and non-reviewers (Figure 4(b)).

For example, the line of $D_{sl} = 5.5$ in the first sub-figure of Figure 4(a) shows that, with just 1 mandatory revision, there is a probability of 0.75 that 25% or less students get no revisions. Mathematically, $F(25\%) = P(X \leq 25\%) = 0.75$, where X is the random variable that represents the percentage of students with no revisions.

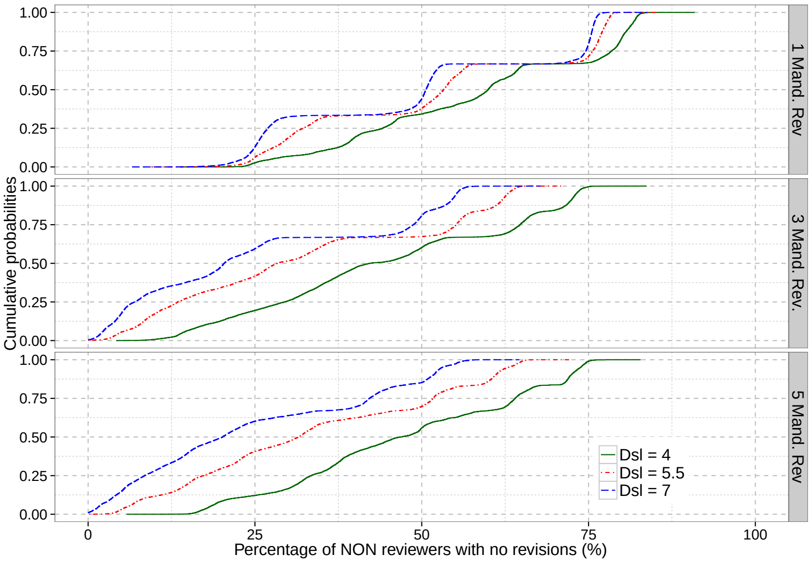
The desired behaviour is to leave as few students as possible with no revisions, i.e. to maximise the probability of having 0% students with no revisions. In fact, $F(0\%) = 1$ for the ideal algorithm. The closer to that ideal, the better the peer matching algorithms are, i.e. the steeper $F(x)$ grows to 1 the better the algorithm is (regarding this evaluation metric).

As expected, the duration of the sliding period, D_{sl}, affects the number of students that receive revisions, showing better performance as the sliding deadline grows towards its upper bound (fixed by the global revision deadline, D_r). Actually, all the plots in Figure 4 show a better behaviour for $D_{sl} = D_r (= 7)$.

Figure 4 also shows how the reviewers are rewarded for their commitment (Figure 4(a)), while non-reviewers (Figure 4(b)) obtain a worse treatment from the algorithm. An algorithm that ensures that non-reviewers obtain no revisions



(a) Students willing to review



(b) Students not willing to review

Figure 4: $F(x) = P(X \leq x)$ CDFs of the percentage of students that receive no revisions for the SDCR algorithm, depending on the length of the sliding deadline and the number of mandatory revisions.

would be $F(x) = 0$ for $x < 100$ and $F(100\%) = 1$. However, the proposed algorithm assumes that the non-reviewers could change their mind until the deadline expires. So, their assignments are reviewed, though with less priority.

The different behaviour between reviewers and non-reviewers is even clearer when analysing the effect of the number of mandatory revisions on the algorithm performance (see also Figure 4). For only one mandatory revision ($N_r = 1$) with $D_{sl} = 7$, $F(25\%) \approx 0.75$ for reviewers, while $F(25\%) \approx 0.1$ for non-reviewers (and falling to around 0.02 for $D_{sl} = 4$), as the former have more priority than the latter and there are fewer optional revisions to assign.

4.3.2 Comparative evaluation of the algorithms. Percentage of students that receive no submissions

Figures 5 and 6 compare the behaviour of the SDCR, the FDCR and the Baseline algorithms, depending on the number of mandatory revisions (Figure 5) and on the number of students enrolled in the course (Figure 6). In order to make this comparison as fair as possible, the length of the sliding deadline for the SDCR algorithm is fixed to $D_{sl} = D_r = 7$.

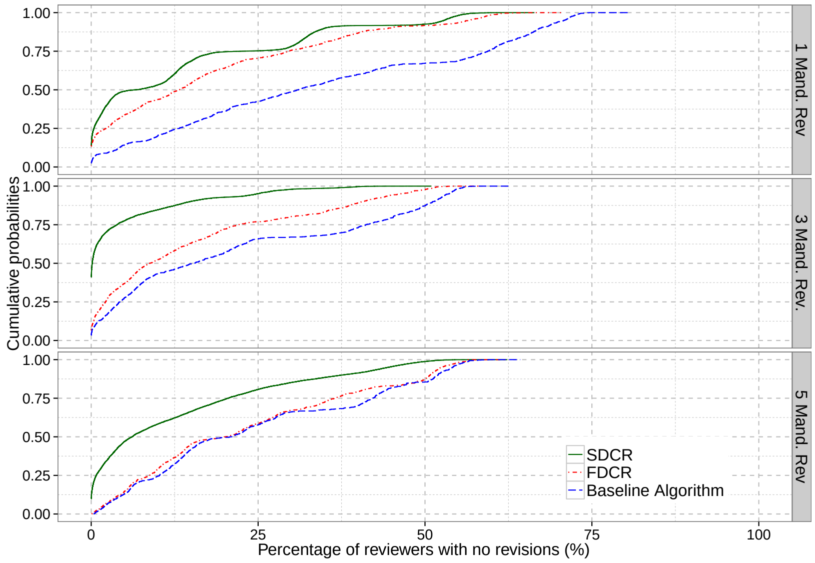
Globally, it can be seen that the proposed improved algorithms perform better than the baseline for reviewers, i.e. the probability of reviewers not receiving feedback is smaller (Figures 5(a) and 6(a)), though at the expense of worse results for non-reviewers (Figures 5(b) and 6(b)), as expected.

The SDCR algorithm shows the best performance (according to the evaluation metrics defined). Regarding the FDCR algorithm, it shows a better behaviour than the Baseline algorithm (Figure 5(a)), tending to it as the number of mandatory revisions grows. The only difference between both algorithms is how the optional revisions are managed, because both offer the possibility of optional revisions, but in the FDCR the reviewers are prioritised.

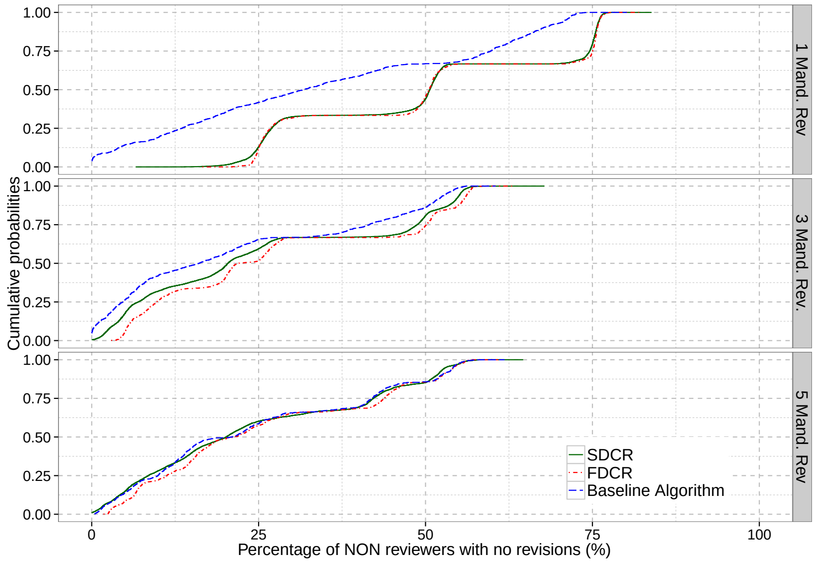
As it can be seen in Figure 5(a), the probability that 25% of reviewer students get no revisions for FDCR with $N_r = 3$ mandatory revisions is $F(25\%) = 0.77$, better than for the Baseline algorithm, $F(25\%) = 0.658$. Comparing these values with the result for both algorithms with $N_r = 5$ mandatory revisions ($F(25\%) \approx 0.58$), it could be concluded that the optimal value of revisions is $N_r = 3$.

However, these results depend on the actual parameters and also on the student model. Our conclusion is that, when enough time for performing more revisions is granted to students, the behaviour of FDCR is better than the Baseline algorithm.

Figure 5(a) also shows how, given a number of mandatory revisions, the performance of the SDCR algorithm regarding reviewers, is better than the performance of both the FDCR algorithm and the Baseline algorithm. Furthermore, the behaviour of SDCR with only one mandatory revision ($N_r = 1$) is equal or better than the behaviour of the FDCR and the Baseline algorithm for any



(a) Students willing to review



(b) Students not willing to review

Figure 5: $F(x) = P(X \leq x)$ CDF of the percentage of students that receive no revisions, depending on the number of mandatory revisions and implemented algorithm

value of N_r. This allows course managers to dramatically reduce the students' workload without jeopardising the performance of the system.

As expected, the Baseline algorithm does not punish non-reviewer students, while both SDCR and FDCR reward the reviewers. As explained in the previous section, this effect is more patent when a student has only $N_r = 1$ mandatory revision, (first plot in Figures 5(b) and 6(b)), as there are fewer revisions opportunities to allocate and the algorithms favour the reviewers.

Figure 6 shows how the algorithms behave depending on the number of enrolled students (1000, 5000, 10000, and 25000). It can be seen that the behaviour of the algorithms does not depend on this number.

4.3.3 Comparative evaluation of the algorithms. Number of total revisions received by each student

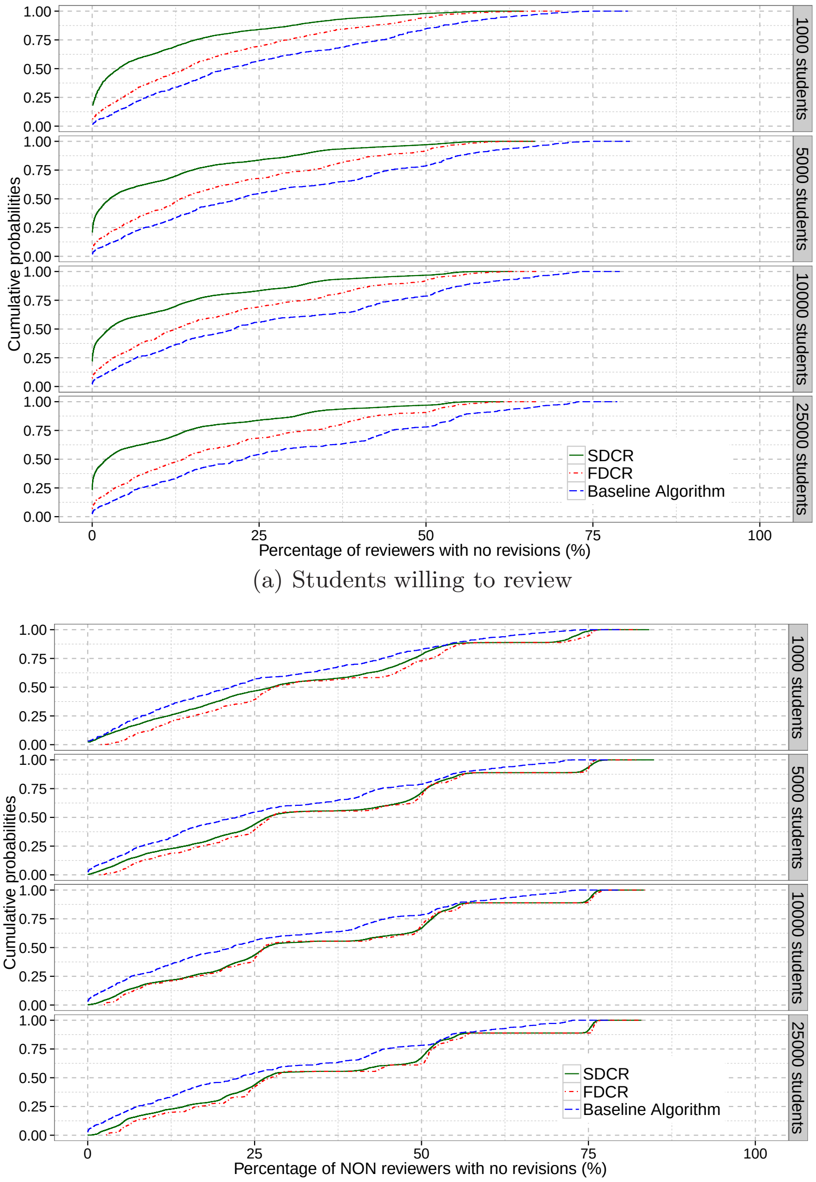
This section analyses the number of total revisions received by each student. In order to perform this analysis, we computed the cumulative distribution functions (CDFs) of the number of revisions for each student (both for reviewers and non-reviewers) depending on the number of mandatory revisions (see Figure 7).

As it can be seen in Figure 7(a), the SDCR algorithm shows a better behaviour for any number of mandatory revisions (as stated in the previous section), as the probability of reviewers getting 0 revisions is $F(0) = 0.1$ for one mandatory revision ($N_r = 1$); $F(0) = 0.02$ for $N_r = 3$; and $F(0) = 0.08$ for $N_r = 5$. For the Baseline algorithm, these probabilities are $F(0) = 0.25$ for $N_r = 1$; $F(0) = 0.14$ for $N_r = 3$; for $F(0) = 0.18$ with $N_r = 5$. Moreover, the number of revisions received by the reviewers is greater for the SDCR algorithm than for the Baseline algorithm. For example, the probability of getting 3 or more revisions with $N_r = 3$ is $P(revisions_{received} = 3|N_r = 3) = 1 - F(2) = 0.3$ for the SDCR algorithm, and $P(revisions_{received} = 3|N_r = 3) = 1 - F(2) = 0.15$ for the Baseline algorithm.

5 Discussion

The results of the experiments show that the mechanisms we propose for improving the baseline algorithm provide significant performance gains.

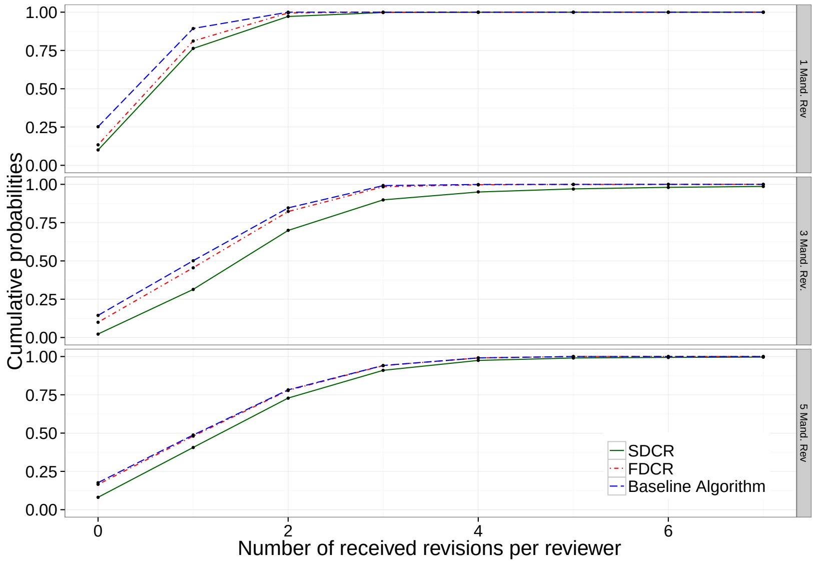
On the one hand, giving priority to the reviewers when assigning optional revisions (FDCR algorithm) allows the algorithm to reward the commitment of the students. However, this measure is only effective if the students have enough time and are willing to perform additional revisions within the set deadline. Thus, in order to ensure the effectiveness of this measure, the deadline should be set to provide students with extra time, more than the strictly needed, to perform the mandatory revisions. According to the results of the simulations, this is more



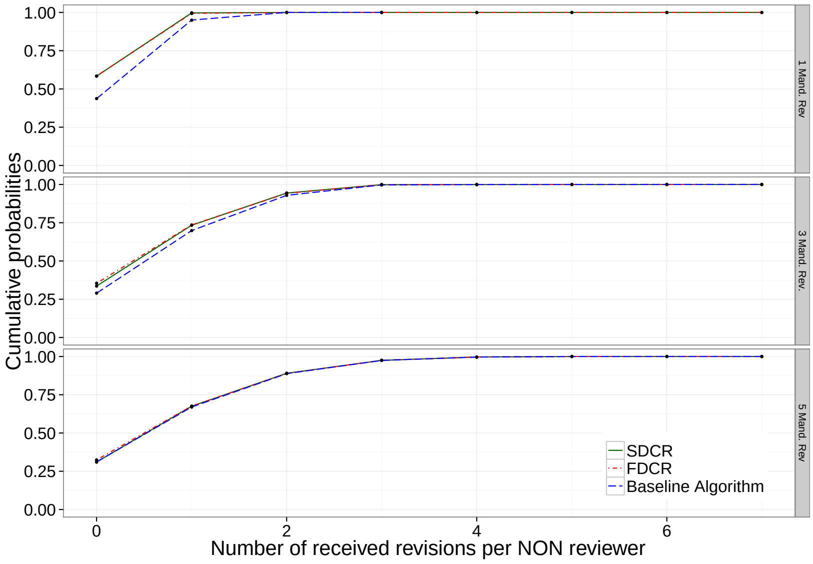
(a) Students willing to review

(b) Students not willing to review

Figure 6: $F(x) = P(X \leq x)$ CDF of the percentage of students that receive no revisions, depending on the number of enrolled students and implemented algorithm



(a) Reviewers



(b) Non-Reviewers

Figure 7: $F(x) = P(X \leq x)$ CDF of the number of received revisions, depending on the number of mandatory reviews per student

important than the number of mandatory revisions. It is also necessary for the whole system to motivate the students to perform these optional revisions.

On the other hand, the usage of sliding deadlines together with the commitment rewarding mechanism (SDCR algorithm) allows reassigning revisions. In other words, if a reviewer fails to perform the revision of one or more assignments, the system is able to reassign them to another student. The algorithm is thus able to minimise the number of non reviewed assignments.

The SDCR algorithm also allows decreasing the student workload. As shown before, the SDCR shows a better behaviour, even with fewer mandatory revisions, than the Baseline algorithm for any number of mandatory revisions. Concretely, in our experiments, the proposed solution shows optimal performance for $N_r = 3$, though it beats the performance of the Baseline algorithm even for only one mandatory review per student. A typical value in MOOC peer review processes is $N_r = 5$, which means a significant reduction of the student workload with the proposed improvements.

Regarding the duration for the sliding deadline, the SDCR algorithm shows the best performance, as expected, when it is equal to the global revision deadline. In such case, reviewers are granted more time for performing their task, thus minimising the risk of failing to complete it.

The simulator we developed would also allow recommending optimal values for typical parameters in the peer review process. The main challenge is to define an appropriate student temporal model. The simulations included in this paper rely on using left-truncated normal distributions to model the students' behaviour. It would be interesting to feed the simulator model with actual data from real MOOCs instead. With such information, the simulator could be used to recommend, given the characteristics of a given course and its audience, the best algorithm to be applied and the optimal values for its parameters.

Finally, as emphasised before, the effectiveness of these algorithms depends heavily on the students' participation. In consequence, they should be complemented with motivational mechanisms, such as reputation, in order to increase student's commitment to the review process and its quality.

First, as said before, to ensure the completion of the mandatory revisions, students that do not complete the mandatory review process, should not be allowed to access the feedback from their peers and, maybe, be penalised on their reputation. However this is not enough, as the behaviour of these algorithms relies on the willingness of the students to perform optional revisions. So, performing optional revisions should be encouraged through augmenting the student reputation if optional revisions are made.

Additionally, in order to improve the quality of the received revisions (and thus decrease the consequent student frustration over the lack of it), the system should allow the author of a submission to evaluate the revisions s/he received.

It could also integrate natural language processing technology to automatically classify (and evaluate) the revisions. With all this information, the system could change the reputation of an student (increasing or decreasing it gradually) depending on the quality of her/his revisions.

6 Conclusions and Future Work

The characteristics of MOOCs pose new educational challenges, particularly for assessment. In this paper, we have focused on the use of peer review in MOOCs. We have presented an algorithm, the Sliding Deadline Commitment Rewarding (SDCR) peer matching algorithm, for selecting the reviewers that minimises students' frustration due to not receiving the expected reviews.

Traditional matching algorithms try to alleviate this problem by increasing the submissions assigned to each student for review, thus aggravating their workload. The SDCR prioritises committed students, ensuring that those who actually submit the assigned reviews receive feedback for their own work. Besides, it uses sliding deadlines to take advantage of the students that wish to revise early, thus making the reassignment of non reviewed assignments possible. Experimental results confirm that the SDCR algorithm allows reducing the students workload while maintaining a minimal rate of active students with no reviews, much lower than the Baseline algorithm, thus reducing students' frustration.

Regarding future work, it is planned to deploy the algorithm and evaluate it in real MOOCs, with actual students (instead of using simulations). This will allow us to study the performance of the algorithm depending on the profile of different students and to study how their behaviour might change depending on how the peer review process is introduced to them. As stated in the Discussion Section, the effectiveness of our algorithm depends on the number of optional revisions performed by the students and it is necessary that the whole system motivates the students to perform these optional revisions. So, if the deployment platform allows it, it is also planned to introduce some kind of gamification within the system, such as a point system or a leaderboard based on the quantity and quality of the performed optional revisions, and to study its effects in the behaviour of the students and also in the performance of our algorithm. Open source MOOC platforms, like OpenEdX[6], Google Course Builder[7], or OpenMooc[8], would facilitate the integration of the algorithm.

Additionally, if public datasets were available (with real values of the temporal behaviour of the students), they could be used for testing the performance of the algorithms too.

[6] code.edx.org

[7] code.google.com/p/coursebuilder/

[8] openmooc.org

Acknowledgements

This work was partially funded by the Spanish Competitiveness and Economy Ministry (*Ministerio de Economía y Competitividad*) project *"EEE – Espacios Educativos Especulares"* (TIN2011-28308-C03-01), and by the Madrid regional project *"eMadrid"* (S2009/TIC-1650).

References

[Alario-Hoyos et al. 2013] Alario-Hoyos, C., Pérez-Sanagustín, M., Delgado-Kloos, C., and Parada G., H. A. 2013. Analysing the impact of built-in and external Social Tools in a MOOC on Educational Technologies. In *Proceedings of the Eight European Conference on Technology Enhanced Learning (EC-TEL).*

[Attali 2007] Attali, Y. 2007. *On-the-fly customization of automated essay scoring (RR-07-42).* Technical Report. ETS Research & Development, Princeton, NJ. http://www.ets.org/Media/Research/pdf/RR-07-42.pdf

[Bailey 1994] Bailey, K. D. 1994. *Methods of Social Research* (4 ed.). The Free Press, New York.

[Baker et al. 2010] Baker, R. S.J.d., D'Mello, S. K., Rodrigo, Ma.M. T., and Graesser, A. C. 2010. Better to be frustrated than bored: The incidence, persistence, and impact of learners cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies* 68, 4 (April 2010), 223–241.

[Balfour 2013] Balfour, Stephen P. 2013. Assessing Writing in MOOCs: Automated Essay Scoring and Calibrated Peer Review^{TM}. *Research & Practice in Assessment* 8, 1 (2013), 40–48.

[Clow 2013] Clow, D. 2013. MOOCs and the funnel of participation. In *Proc. of 3rd Conf. on Learning Analytics and Knowledge*. Leuven, Belgium.

[Cohen et al. 2007] Cohen, L., Manion, L., and Morrison, K. 2007. *Research Methods in Education* (6 ed.). Routledge, Oxon, UK.

[Craig et al. 2004] Craig, S., Graesser, A., Sullins, J., and Gholson, B. 2004. Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media* 29, 3 (Oct. 2004), 241–250.

[Daniel 2012] Daniel, J. 2012. *Making Sense of MOOCs: Musings in a Maze of Myth, Paradox and Possibility.* Technical Report. Korea National Open University. http://www.tonybates.ca/wp-content/uploads/Making-Sense-of-MOOCs.pdf Retrieved June 2013.

[DiSalvio 2012] DiSalvio, P. 2012. Pardon the disruption: Innovation changes how we think about higher education. *New England Journal of Higher Education* (2012).

[Dochy et al. 1999] Dochy, F., Segers, M., and Sluijsmans, D. 1999. The Use of Self-, Peer and Co-assessment in Higher Education: a review. *Studies in Higher Education* 24, 3 (1999), 331–350.

[Downes 2010] Downes, S. 2010. The Role of the Educator. *Huffington Post Education* (2010). www.huffingtonpost.com/stephen-downes/the-role-of-the-educator_b_790937.html Retrieved June 2013.

[Ebner and Holzinger 2007] Ebner, M. and Holzinger, A. 2007. Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers & Education* 49, 3 (2007), 873–890.

[Ebner et al. 2006] Ebner, M., Zechner, J., and Holzinger, A. 2006. Why is Wikipedia so successful? Experiences in establishing the principles in Higher Education. In *Proc. 6th Int. Conf. on Knowledge Management (I-KNOW).* Graz, Austria, 527–535.

[Falchicov 1995] Falchicov, N. 1995. Peer feedback marking-Developing peer assessment. *Innovations in Education and TrainingInternational* 32 (1995), 175–187.

[Falchicov 1996] Falchicov, N. 1996. Improving learning through critical peer feedback and reflection. In *Different Approaches: Theory and Practice in Higher Education. Proceedings HERDSA Conference 1996*. Perth, Western Australia.

[Gehringer and Cui 2002] Gehringer, E. F. and Cui, Y. 2002. An effective strategy for the dynamic mapping of peer reviewers. In *Proceedings of the 2002 American Society for Engineering Education Annual Conference and Exposition*. ASEE.

[Goldin 2012] Goldin, I. M. 2012. Accounting for Peer Reviewer Bias with Bayesian Models. In *Proceedings of the Workshop on Intelligent Support for Learning Groups at the 11th International Conference on Intelligent Tutoring Systems*. Chania.

[Graesser et al. 2007] Graesser, A., D'Mello, S., Chipman, P., King, B., and McDaniel, B. 2007. Exploring relationships between affect and learning with AutoTutor. In *Artificial Intelligence in Education: Building Technology Rich Learning Contexts that Work*. IOS Press, Amsterdam, 16–23.

[Graesser and McNamara 2012] Graesser, A. C. and McNamara, D. S. 2012. *APA handbook of research methods in psychology*. Vol. 1: Foundations, planning, measures, and psychometrics. American Psychological Association, Washington, DC, Chapter Automated analysis of essays and open-ended verbal responses, 307–325.

[Gupta and Sambyal 2013] Gupta, R. and Sambyal, N. 2013. An understanding Approach towards MOOCs. *International Journal of Emerging Technology and Advanced Engineering* 3, 6 (2013), 312–315.

[Holzinger et al. 2009] Holzinger, A., Kickmeier-Rust, M.D., and Ebner, M. 2009. Interactive technology for enhancing distributed learning: a study on weblogs. In *Proc, 23rd British HCI Group Annual Conf. on People and Computers: Celebrating People and Technology (BCS-HCI '09)*. 309–312.

[Hone 2006] Hone, K. 2006. Empathic agents to reduce user frustration: the effects of varying agent characteristics. *Interacting with Computers* 18 (2006), 227–245.

[Hyman 2012] Hyman, P. 2012. In the year of disruptive education. *Commun. ACM* 55, 12 (Dec. 2012), 20.

[Jonsson and Svingby 2007] Jonsson, A. and Svingby, G. 2007. The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review* 2, 2 (Jan. 2007), 130–144.

[Klein et al. 2002] Klein, J., Moon, Y., and Picard, R. 2002. This computer responds to user frustration–theory, design, and results. *Interacting with Computers* 14, 2 (2002), 119–140.

[Kolowich 2013] Kolowich, S. 2013. The professors who make the MOOCs. *The Chronicle of Higher Education* (March 21 2013).

[Kop et al. 2011] Kop, R., Fournier, H., and Mak, J.S.F. 2011. A pedagogy of Abundance or a Pedagogy to Support Human Beings? Participant Support on Massive Open Online Courses. *International Review of Research in Open and Distance Learning* 12, 7 (2011), 74–93.

[Kort et al. 2001] Kort, B., Reilly, R., and Picard, R.W. 2001. An affective model of interplay between emotions and learning: reengineering educational pedagogy-building a learning companion. In *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on*. Madison, WI, 43–46.

[Mackness et al. 2010] Mackness, J., Mak, S., and Williams, R. 2010. The Ideals and Reality of Participating in a MOOC. In *Proc. of 7th Int. Conf. on Networked Learning*. Aalborg, Denmark, 266–274.

[Markoff 2013] Markoff, J. 2013. Essay-grading software offers professors a break. *The New York Times* (2013). `http://www.nytimes.com/2013/04/05/science/new-test-for-computers-grading-essays-at-college-level.html`

[McAuley et al. 2010] McAuley, A., Stewart, B., Siemens, G., and Cormier, D. 2010. The MOOC model for digital practice. (2010).

[McQuiggan et al. 2007] McQuiggan, S.W., Lee, S., and Lester, J.C. 2007. Early prediction of student frustration. In *Proceedings of the Second International Conference on Affective Computing and Intelligent Interaction.* 698–709.

[NCTE 2013] NCTE, National Council of Teachers of English. 2013. Machine scoring fails the test. NCTE Position Statement on Machine Scoring. http://www.ncte.org/positions/statements/machine_scoring. (2013).

[Nilson 2003] Nilson, Linda. 2003. Improving Student Peer Feedback. *College teaching* 51, 1 (2003), 34–38.

[Piech et al. 2013] Piech, C., Huang, J., and Chen, Z. 2013. Tuned Models of Peer Assessment in MOOCs. In *Proceedings of the 6th International Conference on Educational Data Mining.* Memphis, Tennessee.

[Ridgway 1998] Ridgway, J. 1998. *The Modeling of Systems and Macro-Systemic Change: Lessons for Evaluation from Epidemiology and Ecology.* National Institute for Science Education, University of Wisconsin-Madison.

[Robinson 2001] Robinson, R. 2001. Calibrated Peer Review$^{\mathrm{TM}}$: An Application To Increase Student Reading & Writing Skills. *The American Biology Teacher* 63, 7 (2001), 474–480.

[Rodriguez 2012] Rodriguez, O. 2012. MOOCs and the AI-Stanford like courses: Two successful and distinct course formats for massive open online courses. *European Journal of Open and Distance Learning* (2012).

[Russell 2004] Russell, A. A. 2004. Calibrated Peer Review–A Writing and Critical-Thinking Instructional Tool. *Teaching Tips: Innovations in Undergraduate Science Instruction* (2004), 54.

[Sandeen 2013] Sandeen, C. 2013. Assessments Place in the New MOOC World. *Research & Practice in Assessment* 8 (2013), 5–12.

[Shermis et al. 2010] Shermis, M. D., Burstein, J., Higgins, D., and Zechner, K. 2010. *International encyclopedia of education* (3rd ed.). Elsevier, Oxford, England, Chapter Automated essay scoring: Writing assessment and instruction, 7580.

[Siemens 2005] Siemens, G. 2005. Connectivism: A learning theory for the digital age. In *International Journal of Instructional Technology and Distance Learning.*

[Siemens 2006] Siemens, G. 2006. *Knowing knowledge.* BC: Lulu Press, Vancouver.

[Spoelstra and Sklar 2008] Spoelstra, M. and Sklar, E. 2008. Agent-based Simulation of Group Learning. In *Multi-Agent-Based Simulation VIII*, AntunesL, PaolucciM, and NorlingE (Eds.). LNAI, Vol. 5003. Springer-Verlag, Berlin, Heidelberg, 69–83.

[Topping 1998] Topping, K. 1998. Peer Assessment Between Students in Colleges and Universities. *Review of Educational Research* 68, 3 (Fall 1998), 249–276.

[Topping 2005] Topping, K. 2005. Trends in peer learning. *Educational psychology* 25, 6 (12 2005), 631–645.

[Topping 2003] Topping, K. J. 2003. Self and peer assessment in school and university: Reliability, validity and utility. In *Optimizing new modes of assessment: In search of qualities and standard*, SegersMSR, DochyFJRC, and CascallarEC (Eds.). Kluwer Academic Publishers, Dordrecht, Netherlands, 55–87.

[Tymms 1996] Tymms, P. 1996. Theories, models and simulations: school effectiveness at an impasse. In *Merging Traditions: The Future of Research on School Effectiveness and School Improvement*, GrayJ, ReynoldsD, Fitz-GibbonCT, and JessonD (Eds.). Cassell, London.

[van den Berg et al. 2006] van den Berg, I., Admiraal, W., and Pilot, A. 2006. Design principles and outcomes of peer assessment in higher education. *Studies in Higher Education* 31, 3 (2006), 341–356.

[van der Pol et al. 2008] van der Pol, J., van den Berg, B.A.M., Admiraal, W.F., and Simons, P.R.J. 2008. The nature, reception, and use of online peer feedback in higher education. *Computers & Education* 51, 4 (Dec. 2008), 1804–1817.

[van Zundert et al. 2010] van Zundert, M., Sluijsmans, D., and van Merriënboer, J. 2010. Effective peer assessment processes: Research findings and future directions. *Learning and Instruction* 20, 4 (Aug. 2010), 270–279.