Doctoral Thesis

Design and Analysis of Allocation Methods for Peer Assessment in Education

Hideaki Ohashi

March 2020

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Doctoral Thesis
submitted to Department of Social Informatics,
Graduate School of Informatics,
Kyoto University
in partial fulfillment of the requirements for the degree of
DOCTOR of INFORMATICS

Thesis Committee: Masatoshi Yoshikawa, Professor
Keishi Tajima, Professor
Hisashi Kashima, Professor

Design and Analysis of Allocation Methods for Peer Assessment in Education*

Hideaki Ohashi

Abstract

Peer assessment in education is a framework for reviewing submissions among students, and its merits, such as pedagogical benefits and cognitive gains, have been discussed for a long time. In recent years, online education has attracted attention, especially on massive open online courses (MOOCs), where tens of thousands of students may participate in a single class. As a result, the scalability of peer assessment has been focused on in addition to the conventional usefulness. As described above, peer assessment is regarded as an effective tool in education from various aspects.

This research mainly focuses on the online peer assessment system, not the traditional offline peer assessment. Online peer assessment in education has a serious drawback as follows: There are many students who do not work because they have negative opinions in peer assessment. Problems with peer assessment where students are dissatisfied include the following: "Imbalance of the number of reviews due to dropouts" and "Low reliability of of students' reviewing results".

Our research attempts to solve these issues in order to improve peer assessment to a more attractive tool. In this study, we focus on student-submission allocation because there are few studies that focus on student-submission allocation.

The objectives and impacts of our studies are described below. In the first study, in order to solve the above first problem, we developed a new adaptive allocation method which achieves that the submission of one student is reviewed as many times as the same student reviews the submissions of others in most cases. Additionally, we extended the proposed method to the method which can consider the second problem. We theoretically analyzed the degree of imbalance when using our first proposed method, and compared the imbalance between proposed allocation methods and existing allocation methods through simulation.

*Doctoral Thesis, Department of Social Informatics, Graduate School of Informatics, Kyoto University, KU-I-DT6960-29-1489, March 2020.

In the second study, we analyzed what kind of student-submission allocation is effective for the existing score estimation method from multiple students' scores, which is developed to solve the above second problem. This analysis indicates that some allocation methods which are considered to be used in actual peer assessment has bad effect on estimation accuracy, and random allocation is superior.

The above two studies recommend different student-submission allocation methods for two different objectives. In third study, we pointed out that, when using the allocation method proposed in the first study, the estimation accuracy decreases under certain circumstances. Then, we proposed an allocation method that considers the trade-off between two objectives. In addition, we discussed the application possibility of the methodology used in the proposed allocation method.

As described above, we developed and analyzed student-submission allocation methods for peer assessment, which have been rarely focused on, for improving the problems of peer assessment.

Keywords: Educational Technology, Peer Assessment, MOOCs, Algorithm, Statistical Model

Contents

List of Figures

List of Tables

CHAPTER 1

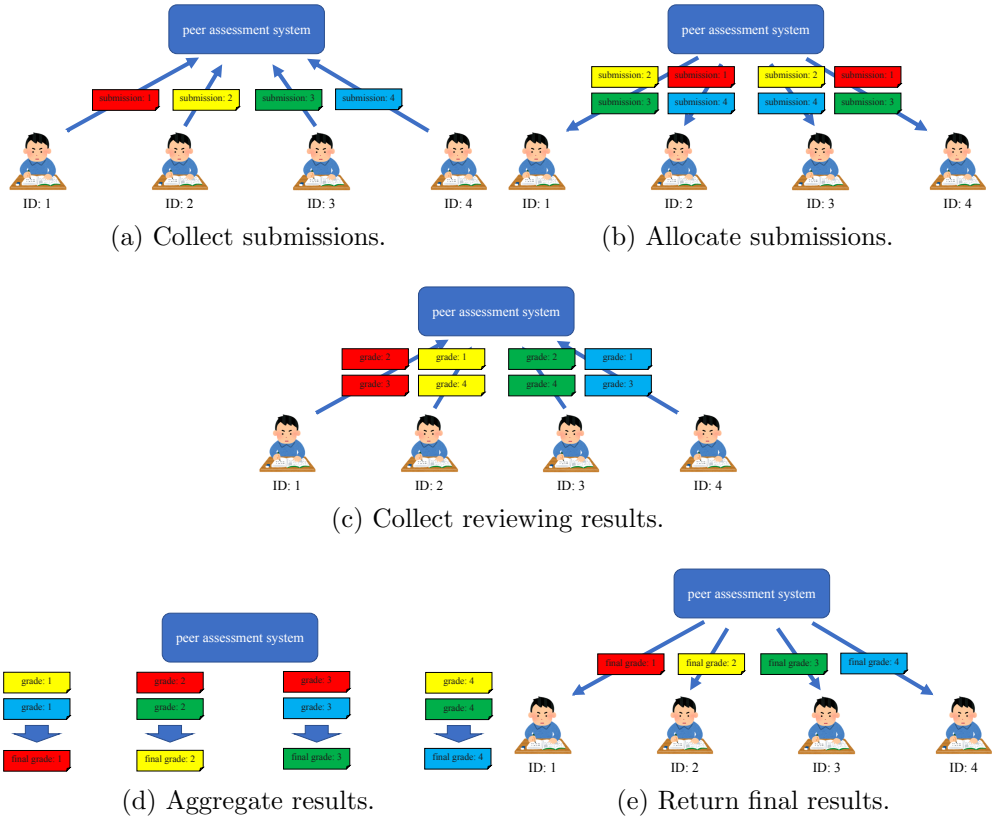
INTRODUCTION

In this chapter, we first give an overview of peer assessment in education. Next, we discuss two problems of online peer assessment: imbalance of the number of reviews due to dropouts and low reliability of reviews in peer assessment. After that, we outline our research. Our research mainly focuses on student-submission allocation in peer assessment, and develops and analyzes allocation methods to improve the above problems. Finally, we describe the structure of this paper.

1.1 Background

1.1.1 Peer Assessment in Education

Peer assessment in education is a framework for reviewing submissions among students. Peer assessment has been conducted for a long time and its usefulness, such as pedagogical benefits and cognitive gains, has been discussed [1–3]. In recent years, online education has attracted attention, especially on massive open online courses (MOOCs), where tens of thousands of students may participate in a single class [4,5]. As a result, the scalability of peer assessment has been focused on in addition to the conventional usefulness [5, 6]. Scalability is a characteristic that the number of reviewers increases by the number of submissions in peer assessment, so that it is possible to review a large number of submissions that



(a) Collect submissions. (b) Allocate submissions.

(c) Collect reviewing results.

(d) Aggregate results. (e) Return final results.

Figure 1.1. The flow of peer assessment.

cannot be reviewed by few teachers or TAs. As described above, peer assessment is regarded as an effective tool in education from various aspects.

This research mainly focuses on the online peer assessment system [7, 8], not the traditional offline peer assessment [1, 2]. Peer assessment system generally consists of five steps: "collect submissions", "allocate submissions", "collect reviewing results", "aggregate results", and "return final results" (see Figure 1.1). In addition, the reviewing approach can be roughly classified into two types, score-based reviewing and comment-based reviewing. Note that the "aggregate results" step is not always necessary in comment-based reviewing.

We assume that:

1. Peer assessment deal with open-ended assignments, such as design problems and essays, where answers are not uniquely determined

2. A teacher discloses reviewing criteria for students in advance

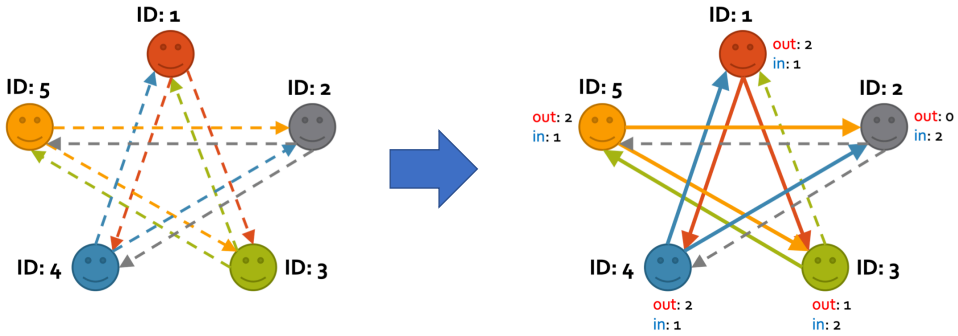


Figure 1.2. Existing student-submission allocation and students' dropout.

3. A submission is reviewed by multiple students

The reason for first assumption is that it is common to replace the manual review for assignments where the answer is uniquely determined with automatic review in online peer assessment system [9, 10]. The second assumption is because the student's reviewing ability is basically lower than the teacher's reviewing ability, so that supplementary materials are effective for making the reviewing results consistent [11]. The reason for final assumption is that, as in second assumption, it is preferable to use multiple reviewing results rather than a single result because the reviewing ability of the students is low [5].

1.1.2 Problems of Peer Assessment

Online peer assessment in education has a serious drawback. There are many students who do not work because they have negative opinions in peer assessment [12–14]. Problems with peer assessment where students are dissatisfied include the following [12]:

- Imbalance of the number of reviews due to dropouts
- Low reliability of of students' reviewing results

Before discussing the first problem, we explain the relationship between student-submission allocation and dropout in the conventional peer assessment with reference to Figure 1.2. In the graph on the left of this figure, nodes represent students, and the dotted edges between nodes indicate student-submission allocation. Note that each edge represents an allocation, drawing from a reviewer

to a reviewee. For example, in the case of this figure, the student with ID 1 is allocated to the students' submissions with ID 3 and ID 4. Each student is allocated to two submissions, and each student's submission is allocated to two students in this figure.

The graph on the right shows the actual reviewing, where solid edges indicate that the reviewing was actually performed. In this case, imbalance occurs in the number of actual reviews. For example, a student with ID 1 reviews two students, but his submission is reviewed by only one student. This imbalance can be attributed to a student with ID 3 who drops out reviewing. Especially when comment-based reviewing is performed, student dissatisfaction is likely to increase due to imbalance in peer assessment allocation.

The second problem is mainly due to the student's lack of reviewing ability. The above-mentioned reviewing criteria and multiple reviewing are examples of efforts to improve the reliability of reviewing results. However, lack of reviewing ability is still a serious problem.

1.2 Overview of Our Research

Our research attempts to solve the above two issues in order to improve peer assessment to a more attractive tool. As described above, there are five steps in the peer assessment, however, there are only two steps that have a high possibility of improvement: "allocate submissions" and "aggregate results". In this study, we focus on student-submission allocation because many studies deal with score aggregation [5, 15], but few studies focus on improving student-submission allocation. Further details on the position of our research in the existing studies on peer assessment are described in Chapter 2.

The objectives and impacts of our studies are described below. In the first study, in order to solve the above first problem, we develop a new adaptive allocation method which achieves that the submission of one student is reviewed as many times as the same student reviews the submissions of others in most cases. Additionally, we extend the proposed method to the method which can consider the second problem. We theoretically analyze the degree of imbalance when using our first proposed method, and compare the imbalance between proposed allocation methods and existing allocation methods through simulation.

In the second study, what kind of student-submission allocation is effective for

the existing score estimation method from multiple students' scores [5], which is developed to solve the low reliability of student reviewing. This analysis indicates that some allocation methods which are considered to be used in actual peer assessment have bad effect on estimation accuracy, and random allocation is superior.

The above two studies recommend different student-submission allocation methods for two different objectives. In third study, we point out that, when using the allocation method proposed in the first study, the estimation accuracy decreases under certain circumstances. Then, we propose an allocation method that considers the trade-off between two objectives. In addition, we discuss the application possibility of the methodology used in the proposed method.

1.3 Dissertation Structure

The structure of this study is as follows. Chapter 2 introduces related work. In Chapter 3, we develop an allocation method to solve imbalance of the number of reviews due to dropouts. In Chapter 4, we analyze what kind of student-submission allocation is effective for the existing score estimation method, which is developed to solve low reliability of students' reviewing results. In Chapter 5, we propose an allocation method that considers above two different objective simultaneously. Finally, we discuss conclusion and future work.

CHAPTER 2

Related Work

In this chapter, we first outline MOOCs, to which peer assessment is effectively applied. Next, we describe related work on peer assessment and mention the position of our research.

2.1 MOOCs

MOOCs is an abbreviation of massive open online courses, which has three features as follows:

1. Students can attend various lectures through the web
2. Many lectures are open to the public for free
3. The number of students attending a single lecture has sometimes reached tens of thousands

As mentioned in Chapter 1, peer assessment has scalability, so it is robust against the third feature of MOOCs. Peer assessment is regarded as the only method that can review a large amount of submissions which cannot be automatically reviewed.

Around 2012, representative MOOCs platforms such as edX [9], coursera [10], and UDACITY [16] were opened [4]. In the following year, a study that first

including \ focusing on	Analysis	Design		
No mathematical formulation	[1-3, 7, 8, 12-14, 22-27]	[11, 32-37]		
Mathematical formulation	[28-31] + research 2	Score aggregation [5, 15, 38-51]	Allocation [52-58] + research 1, 3	Others [64-66]

Table 2.1. Classification of existing research for peer assessment.

analyzed MOOC data appeared. [17]. After that, a study evaluated the design of MOOCs based on huge data [18] and a study proposed statistical models to model student behavior [19].

However, Reich et al. [20] have pointed out that unless the data of MOOCs is utilized more carefully, it will not lead to reliable research results. It has also been pointed out that the number of students participating in MOOCs has decreased in recent years. [21]. They state that the current MOOCs only support the existing learning and are not fundamentally changing higher education.

From the above, it is considered that the existing mechanism of MOOCs is not yet sufficient and improvement is essential. Improvement of peer assessment in this study is expected to help improve MOOCs.

2.2 Peer Assessment

We classify research related to peer assessment according to whether research focuses on analysis or design, and whether research includes mathematical formulation or not (see Table 2.1). In recent years, the spread of online educational systems represented by MOOCs has made it easier to track educational data, and computer scientists have flowed into educational field, resulting in an increase the research including mathematical formulation. Especially, research on score aggregation has increased rapidly in recent years. This is because “aggregate results” step (see Figure 1.1), which is often included in the conventional online peer assessment system, is an important research target. In addition, the mathematical problem definition for score aggregation is easy to set up. On the other hand, there are few papers focusing on student-submission allocation, which is also included in peer assessment system. In this study, we worked on studies focusing on

both analysis and design for student-submission allocation. The existing studies are described below in the order of "research focusing on analysis and not including mathematical formulation", "research focusing on analysis and including mathematical formulation", "research focusing on design and not including mathematical formulation", and "research focusing on design and including mathematical formulation".

2.2.1 Research focusing on Analysis and not including Mathematical Formulation

The existing research classified here analyzes the conventional peer assessment system from various directions without being limited to including mathematical formulation, and the scope of the research is wide. Also, there are many studies that motivate our research.

Topping et al. [1] published the first review paper for peer assessment in 1998. After meta-analysis of 31 existing studies, they conclude that peer assessment shows positive formative effects on student achievement and attitudes. Falchikov et al. [2] later criticized Topping's review for qualitative, and performed a new quantitative meta-analysis on 48 studies. Some papers classified and organized the characteristics of peer assessment, though they did not include meta-analysis for a large amount of research as in the above-mentioned papers [3, 22]. The advantages of peer assessment include cognitive gains, improvements in writing, and possible savings of teachers' time, while disadvantages include the reliability and validity of students' review. This drawback motivates our second and third studies.

In recent years, some studies pointed out that peer assessment in online education has the disadvantage that the number of participants is small [12–14, 23]. This drawback is considered to be due to the difficulty of student control in online education compared to the conventional class-based peer assessment. Acosta et al. [12] examined participants' opinions on peer assessment, and obtain the points of dissatisfaction such as lack of reviews and unreliable student reviews. These points are the motivations for our research.

Some other studies discussed peer assessment in online education like MOOCs [7, 8, 24]. In addition, there are research that conducted an experiment on the relationship between thinking styles and feedback formats of students [25, 26] and

on relationship between self-grading and peer-grading [27].

2.2.2 Research focusing on Analysis and including Mathematical Definition

Our second studies fall into this category. Marsico et al. [28] examined whether the topology of the student-submission allocation graph affects score estimation in peer assessment and addressed research questions similar to those of our study. However, Marsico et al. used a simple Bayesian-network-based model based on scores assessed by teachers [29] and analyzed the effect of propagation based on teachers' scores in the student-submission allocation graph. By contrast, we examine the accuracy of estimation for a general score estimation method [5], which does not assume scoring by teachers.

Some studies developed a machine learning model that predicts student progress and student participation using data obtained by peer assessment [30, 31]. All of these studies are considered to be studies analyzing whether rich data obtained from online peer assessment is meaningful with machine learning techniques.

2.2.3 Research focusing on Design and not Including Mathematical Definition

This category include studies to improve the design of the existing peer assessment without being bound by the mathematical formulation. Gehringer et al. [11] have published a survey paper on methods for improving peer assessment, including a method of giving students submission examples in advance and performing pre-traning for reviewing (calibration), a method of using the quality of reviewing comment (helpfulness ratings), and others. In addition, other studies proposed unique approaches. In CrowdGrader [32], each student gets an overall crowdgrade that combines three grades; consensus grade, accuracy grade and helpfulness grade. In Mechanical TA [33], human teaching assistants (TAs) is involved as a way to assure review quality, and TAs promote students from supervised to independent. In PeerStudio [34], there are many designs for rapid peer feedback. Various peer assessment systems have been proposed and their usefulness has been investigated [35–37].

2.2.4 Research focusing on Design and including Mathematical Definition

In this study, this category is further classified into existing research on score aggregation, student-submission allocation, and other existing research. Although only important studies on score aggregation are listed, but as mentioned earlier, the number of studies on score aggregation is very large.

Score aggregation

Early on, only a few studies were conducted in attempts to improve the accuracy of score estimation in peer assessment [38], but following a study by Piech et al. [5], an increasing number of such studies have emerged. Piech et al. proposed several statistical score estimation methods for peer assessment data. They first proposed a basic statistical method (PG1) and then extended PG1 to a method that exploits the estimation results for different assignments (PG2) and a method based on the hypothesis that there is a correlation between a student's reviewing ability and the score of that student's own submission (PG3). Subsequently, Mi et al. proposed PG4 and PG5, which extend the relationship between a student's reviewing ability and the student's own score to a probabilistic relationship [39]. In the above studies, peer assessment data were collected through absolute evaluation, but score estimation methods based on relative evaluation have also been proposed [15, 40–42]. Research on relative evaluation has been motivated by the hypothesis that relative evaluation is easier for humans than absolute evaluation is. Other similar studies include research involving matrix factorization [43] and work inspired by PageRank [44]. These studies were influenced by quality control research in the context of crowdsourcing [45–47].

As mentioned above, various score estimation methods have been proposed, but there are skeptical claims that there is actually little difference in accuracy among these methods [48]. Similar discussions have been taking place recently with respect to crowdsourcing, which is a field that is adjacent to peer assessment [49]. Therefore, in our second and third studies, we utilize PG1, which is a basic and representative method, instead of a more complicated and advanced method.

There are also studies that have used data other than the score data to improve the accuracy of score estimation. For example, Chan et al. used data on students' social connections [50], and Sunahase et al. proposed a method using corrected

parts in submissions [6]. In addition, a score estimation method for small private online courses (SPOCs), in which online lessons and offline lessons are conducted in parallel, is also under discussion [51].

Although we do not propose a new score estimation method here, we analyze which allocation pattern is most appropriate when using the most basic estimation method (PG1); thus, we contribute to improving score estimation performance indirectly.

Student-submission allocation

Est'vez-Ayres et al. [52] proposed an allocation mechanism to avoid lack of reviews due to dropout and confirmed its usefulness through a simulation. They assumed that some students are willing to review other submissions even when their reviewing number exceeded their reviewed number. We do not assume such optimistic student characteristics in our first study.

Han et al. [53] proposed an allocation method that minimizes the differences between the sums of the reviewing ability value of the reviewers allocated to work, based on an algorithm called Longest Processing Time. They assumed that a student's reviewing ability value is given and, like our first work, aimed to find allocations to achieve fair reviews. However, they did not consider dropout.

Chan et al. proposed a method of adaptively allocating students to submissions while sequentially estimating scores based on the assumption that each student's reviewing ability is known in advance [54]. There are two general approaches to student-submission allocation: one is nonadaptive allocation, as considered in our second study, in which student-submission allocation is performed in advance, and the other is adaptive allocation, as considered by Chan et al. and in our first study, in which allocation is progressively performed. However, in crowdsourcing research, it has been theoretically proven that when a given quality control method is applied, conducting sequential worker-task allocation while progressively estimating the task scores and the abilities of the workers is no better than performing batch estimation after advance allocation [55]. Therefore, in our second study, we analyze the relationship between student-submission allocation and score estimation in the nonadaptive setting rather than the adaptive setting. Additionally, few other studies focus on the student-submission allocation [56–59].

It has been pointed out that research on peer assessment and research on crowdsourcing are closely related [60]. For example, as mentioned earlier, a score

aggregation method influenced by crowdsourcing research is proposed in the context of peer assessment. Some task allocation methods have been proposed for crowdsourcing. [55, 61–63]; however, there have been few proposals for task allocation methods in peer assessments. The difference between task allocations for crowdsourcing and those for peer assessment is the strength of the incentive; crowdsourcing can use clear incentives, such as money, that are unavailable in peer assessment situations. Consequently, dropout is more likely to occur in peer assessments; thus, peer assessment research must consider the effect of dropout. Therefore, our first study focuses on the imbalance caused by dropout and worked on solving this problem. As far as we know, there is only the research [52] that focuses on dropouts, and we do not assume the optimistic students' characteristics in this existing research.

Others

A NLP-based method of automatically assessing review comments (automated meta-reviewing) that prompts the reviewer to correct and improve review comments has been proposed. [64]. Other examples include research that finds suggestions from review comments using machine learning [65] and research that finds inconsistencies between numerical scores and textual feedback [66]. Both are ambitious studies that formulate new problems in peer assessment, but they are in an orthogonal position with our research.

CHAPTER 3

Adaptive Balanced Allocation for Peer Assessments

In this chpater, we focused on the imbalance of the number of reviews due to dropouts in peer assessment, and proposed an allocation algorithm RRB to achieve fair peer assessment using an adaptive allocation approach. In addition, we extended RRB to ARRB that considers students' reviewing ability. We analyzed the RRB algorithm theoretically and show its robustness. We also confirmed the usefulness of the proposed allocation algorithms through experiments.

3.1 Motivation

Some reports indicate that students are not willing to participate in online peer assessments; one reason is that students are disheartened by the lack of reviews [12, 13]. Therefore, we need to develop methods of peer assessment that allow students to receive sufficient feedback based on the number of reviews to increase the number of students who participate in peer assessments.

A major reason for the existence of insufficient review numbers is that peers dropout without reviewing allocated submissions [12,31]. In existing peer assessment systems, each student is usually asked to review a predefined number of submissions, and submissions are allocated to students before the peer assess-

ments start. If a certain number of students drop out of the review process, an imbalance occurs between the number of submissions a student reviews (termed the "reviewing number") and the number of peers who review the submission of the same student (termed the "reviewed number"). When the total imbalance increases, students who diligently finish reviews may suffer from a lack of reviews and be discouraged to participate in future peer assessments.

To address this problem, we develop a new adaptive allocation approach in which students are allocated submissions only when requested. Students can request one submission to review at any time; they can request second and subsequent submissions to review only after they have finished the review of the previously requested submission. This rule is more suitable for a realistic situation in which some students drop out during peer assessments.

Under the above approach, our goal is to reduce the sum of the absolute values of the differences between the reviewing number and reviewed number of each student, termed RR imbalance (reviewing-reviewed imbalance). We propose an allocation algorithm called the RRB (reviewing-reviewed balanced) allocation algorithm, which reduces the RR imbalance, which means that it is highly possible that the submission of one student will be reviewed as many times as that same student reviews the submissions of others. It can be expected that this algorithm resolves dissatisfaction about the lack of reviews and incentivizes students to review the submissions of their peers.

To demonstrate the usefulness of the RRB algorithm, we theoretically prove that the RRB algorithm guarantees an upper bound of the RR imbalance, which does not depend on the number of students; instead, it depends on the maximum reviewing number among students. In practical situations, the maximum reviewing number usually does not increase, even if the number of students grows. Therefore, our results show that the average difference between the reviewing number of each student and the reviewed number decreases as the number of students increases. This property is desirable in MOOC settings from the viewpoint of fairness among students.

However, unfairness still remains up to the amount of the upper bound. To reduce the RR imbalance, extra effort is required. For instance, in MOOC settings, lecturers and TAs could perform extra reviews for students whose reviewing number is above their reviewed number at the end of the peer assessment. In this case, the obtained upper bound can be used to estimate the number of reviews

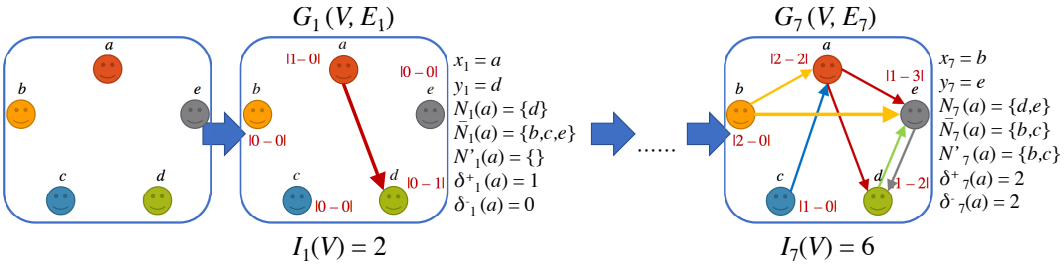


Figure 3.1. Example of adaptive allocation behavior.

the lecturers and TAs need to perform.

In addition, we also consider students's reviewing ability in addition to the RR imbalance. We assume that a scalar reviewing ability value for each student are given in advance, similar to the existing research [53, 54]. When the average reviewing ability value of the reviewers allocated to a submission varies, it means that the least (or most) skillful reviewers concentrate on only one submission. Hence, we want to make this average value be balanced among submissions. Therefore, we extend RR imbalance to a metric, called ARR imbalance (ability-aware reviewing-reviewed imbalance), to measure the imbalance of the average ability of the reviewers. We propose an allocation algorithm, called ARRB (the ability-aware reviewing-reviewed balanced allocation algorithm) to minimize the ARR imbalance.

To show the effectiveness of the proposed two algorithms, we experimentally compare the performance with that of the existing nonadaptive allocation through experiments using two types of data.

The remainder of this paper is organized as follows. We describe the problem definitions in this research in Section 3.2. In Section 3.3, we describe the RRB algorithm and ARRB algorithm. In Section 3.4, we prove the upper bound of the RR imbalance by the RRB algorithm. We present the experimental results in Section 3.5, and finally, we conclude this work and suggest future work in Section 3.6.

3.2 Problem Setting

Initially, we explain our problem setting intuitively through Figure 3.1. In this research, to deal with realistic situations in which some students drop out during

the peer assessment process, we propose an allocation algorithm that uses an adaptive allocation approach. Under this approach, a new submission is allocated to a student only when he or she requests one, and he or she can request an additional submission to review only after he or she has finished the review of the previous submission. In addition, we assume that students always complete the requested review. This assumption is considered to be valid because students who are not willing to review do not request a submission in the first place.

In Figure 3.1, we assume that there are five students, a, b, c, d, and e, and each vertex represents a student. First, a requests a submission; then, the submission of d is allocated to a. This allocation is denoted by the directed edge from a to d. We assume that no one can review his or her own submission and that each student can review a given submission only once. After the first allocation, the next allocation occurs when another student requests a submission, and then, a directed edge is drawn. These steps are repeated under an adaptive allocation approach.

Let V be a set of students, E_i be the edge set and G_i be the graph created up to the i-th allocation. Note that $E_0 = \emptyset$. The RR imbalance (reviewing-reviewed imbalance) in graph G_1, which consists of single edge, is the sum of all the absolute values of the differences between the reviewing number (outdegree) and the reviewed number (indegree) as follows: $|1-0| + |0-0| + |0-0| + |0-1| + |0-0| = 2$. Now, let us assume that there are seven allocations during this peer assessment. The final RR imbalance in graph G_7 is $|2-2| + |2-0| + |1-0| + |1-2| + |1-3| = 6$. In this study, we propose an allocation algorithm that reduces the RR imbalance at the end of a peer assessment. We also propose an allocation algorithm to minimize the ARR imbalance that considers reviewing ability value in addition to RR imbalance. Note that, most of the existing peer assessment utilize a nonadaptive approach, namely, determining the number of reviews per student and allocating submissions to all students before peer assessment begins. In our comparison experiments, we apply two algorithms under such a nonadaptive approach as the compared methods.

Some definitions are provided below. Let a student doing the i-th request under the adaptive allocation approach be $x_i \in V$. A submission by a student $y_i (\neq x_i) \in V$ is allocated to x_i before a student x_{i+1} can request a submission. This allocation is represented by a directed edge from x_i to y_i. In the graph G_i, let the set of students whose submissions are allocated to student $v \in V$ be $N_i(v)$ and

$\bar{N}_i(v) = V \setminus \{N_i(v) \cup \{v\}\}$; then, $y_{i+1} \in \bar{N}_i(x_{i+1})$. Moreover, use $N'_i(v)$ to denote the set of students who review the submission of student $v \in V$. The reviewing number (outdegree) of student v in graph G_i is defined as $\delta_i^+(v)(= |N_i(v)|)$, and the reviewed number (indegree) is defined as $\delta_i^-(v)(= |N'_i(v)|)$.

We explain the above definitions using Figure 3.1. In Figure 3.1, we assume five students, a, b, c, d, and e; thus, $V = \{a, b, c, d, e\}$. Initially, student a requests a submission, and the submission of student d is allocated to a; therefore, x_1 and y_1 are a and d, respectively. The edge set E_1 of the graph $G_1(V, E_1)$ contains only one directed edge from a toward d. In addition, $N_1(a) = \{d\}$, $\bar{N}_1(a) = \{b, c, e\}$ and $N'_1(a) = \{\}$, and the node a has an outdegree of 1 and an indegree of 0; consequently, $\delta_1^+(a) = 1$ and $\delta_1^-(a) = 0$.

Let the reviewing ability value of a student $v \in V$ be a nonnegative real number $w(v)$, and a larger value represents a better reviewing ability. In this work, for simplicity, we assume that the reviewing ability value is given as in [53, 54]. Note that estimating the reviewing ability value is out of the scope of this research.

In this study, we first aim at achieving fair assessment based on the number of reviews. Our goal is to reduce the RR imbalance when the last allocation is done during the peer assessment period. The RR imbalance is defined as the sum of the absolute values of the difference between the reviewing number and the reviewed number for all students. That is, when the t-th allocation is finished, RR imbalance $I_t(V)$ can be calculated by the following equation:

$$I_t(V) = \sum_{v \in V} |\delta_t^+(v) - \delta_t^-(v)|$$

Next, we extend RR imbalance considering reviewing ability. We denote the average of the reviewing ability values of the students who review $v \in V$'s submission as $W_t(v)$ and the average value of $W_t(v)$ of all students as $\hat{W}_t(V)$. Our goal is to minimize ARR imbalance, the sum of the RR imbalance and the absolute sum of the difference between $W_t(v)$ and $\hat{W}_t(V)$ for all students. The ARR imbalance $I'_t(V)$ when the t-th allocation is finished is given by the following equation.

$$I'_t(V) = \sum_{v \in V} |\delta_t^+(v) - \delta_t^-(v)| + \lambda \cdot \sum_{v \in V} |W_t(v) - \hat{W}_t(V)|$$

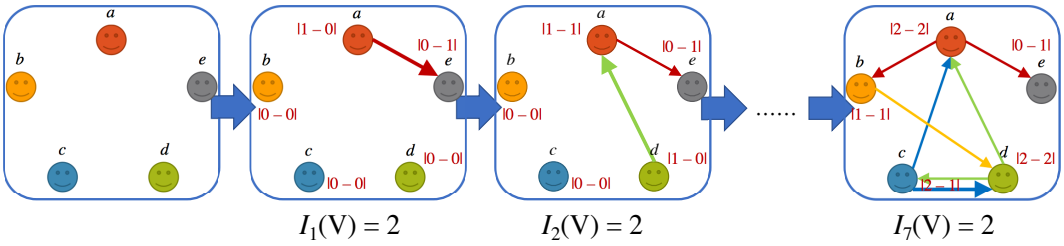


Figure 3.2. Example of RRB behavior.

Note that

$$W_t(v) = \sum_{v' \in N'_t(v)} w(v')/|N'_t(v)|$$
$$\hat{W}_t(V) = \sum_{v \in V} W_t(v)/|V|$$

Here, λ is a nonnegative real number parameter. To emphasize the number of reviews rather than the review quality, λ should be decreased, while to emphasize review quality over review quantity, λ should be increased.

3.3 Algorithm

In this section, we propose an allocation algorithm to reduce the RR imbalance, termed RRB, and the algorithm to reduce the ARR imbalance, termed ARRB. A theoretical analysis of the RRB algorithm is given in Section 3.4, and experiments to evaluate the performance of the RRB and ARRB algorithms are presented in Section 3.5.

3.3.1 RRB (reviewing-reviewed balanced allocation algorithm)

The RRB algorithm adopts a greedy approach to reduce the RR imbalance. We propose an algorithm that y_{i+1} is determined according to the following formula. Note that y_{i+1} is selected randomly when multiple candidates exist.

$$y_{i+1} \in \arg\max_{v \in \bar{N}_i(x_{i+1})} (\delta_i^+(v) - \delta_i^-(v))$$

We provide an intuitive explanation of the above algorithm using Figure 3.2. In this figure, it is assumed that there are five students, a, b, c, d, and e, whose requesting order is $\langle a, d, b, a, c, d, c\rangle$. First, the difference between the reviewing number and the reviewed number of each student is 0; therefore, the submission is randomly allocated to a. Let us assume that the submission of e is randomly selected. Next, because the difference between the reviewing number and reviewed number of a is the maximum, the submission of a is allocated to d. Subsequently, a's difference between reviewing number and reviewed number becomes 0, while for d, the difference becomes 1. Therefore, the submission of d is preferentially allocated in the next step. In Figure 3.2, the above allocation is repeated, showing intuitively how the RRB algorithm aims to reduce the RR imbalance.

3.3.2 ARRB (ability-aware reviewing-reviewed balanced allocation algorithm)

Next, we extend the RRB to an algorithm that reduces the ARR imbalance. Here, $W'_t(v)$ represents the average reviewing ability value of the reviewers who reviews $v \in V$'s submission and the reviewer x_{i+1}. $\hat{w}(V)$ represents the average reviewing ability values of all students. We propose an algorithm that allocates y_{i+1} to x_{i+1} based on the following formula. Note that y_{i+1} is selected randomly when multiple candidates exist.

$$y_{i+1} \in \underset{v \in \bar{N}_i(x_{i+1})}{\arg\max} (\delta_i^+(v) - \delta_i^-(v) - \lambda \cdot |W'_i(v) - \hat{w}(V)|)$$

where

$$W'_i(v) = \sum_{v' \in N'_i(v) \cup \{x_{i+1}\}} w(v')/(|N'_i(v)| + 1)$$
$$\hat{w}(V) = \sum_{v' \in V} w(v')/|V|$$

Ideally, instead of $\hat{w}(V)$, we would use $\hat{W}_t(V)$ to obtain the ARR imbalance; however, $\hat{W}_t(V)$ can be determined only after all allocations are complete. Thus, $\hat{w}(V)$ is used as an approximated value.

3.4 Theoretical Analysis for the RRB Algorithm

In this section, we show that when the maximum outdegree of graph G_i is k and the number of students exceeds $k^2 + k + 1$, the RRB algorithm ensures that the upper bound of the RR imbalance in the graph G_i is $\mathcal{O}(k^2)$. The upper bound does not depend on the total number of students n; it depends only on the maximum number of reviews performed by any one reviewer. When an enormous number of students exist, such as in an MOOC, k is expected to be considerably smaller than n because one student cannot review submissions by everyone. In other words, the proposed algorithm should be extremely effective on MOOCs. Although we assume that the number of students is larger than $k^2 + k + 1$, this is equivalent to the assumption that the total number of students is larger than the square of the reviewing number of any one student. It is natural to use this assumption when many students are participating. In the following section, after presenting two lemmas, we prove our assertion of the upper bound.

Lemma 1 *For a vertex subset $V' \subseteq V$ of graph G_i, suppose that the following inequality holds for all vertices $v \in V'$:*

$$\delta_i^+(v) - \delta_i^-(v) \leq 0$$

We define the set of edges from $V \setminus V'$ to V' as $E_I \subseteq E_i$ and the set of edges from V' to $V \setminus V'$ as $E_O \subseteq E_i$. Then, the following equation is satisfied:

$$I_i(V') = |E_I| - |E_O|$$

Proof 1 *From the assumption, $|\delta_i^+(v) - \delta_i^-(v)| = \delta_i^-(v) - \delta_i^+(v) \geq 0$ is satisfied for any $v \in V'$. Therefore, the RR imbalance on V' is as follows:*

$$I_i(V') = \sum_{v \in V'} \delta_i^-(v) - \delta_i^+(v) = \sum_{v \in V'} \delta_i^-(v) - \sum_{v \in V'} \delta_i^+(v)$$

Here, we define the edge set in V' as $E' \subseteq E_i$, and the following two equations are satisfied:

$$\sum_{v \in V'} \delta_i^-(v) = |E'| + |E_I|$$

$$\sum_{v \in V'} \delta_i^+(v) = |E'| + |E_O|$$

Hence, $I_i(V') = (|E'| + |E_I|) - (|E'| + |E_O|) = |E_I| - |E_O|$ □

Lemma 2 *The maximum outdegree* $\max_{v\in V}\{\delta_i^+(v)\}$ *in* G_i *is defined as* k_i*. Assuming that* $n > k_i^2 + k_i + 1$*, the case that the RR imbalance increases with the* $i+1$*-th allocation, or* $I_{i+1}(V) > I_i(V)$*, is limited to the following case, and the increment is 2.*

$$\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1}) \geq 0 \ \textit{and} \ \delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) = 0$$

Proof 2 *We separate the cases as follows:*

0. $\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1}) < 0$ & $\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) > 0$

1. $\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1}) \geq 0$ & $\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) > 0$

2. $\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1}) < 0$ & $\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) \leq 0$

3. $\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1}) \geq 0$ & $\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) < 0$

4. $\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1}) \geq 0$ & $\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) = 0$

Adding the edges (x_{i+1}, y_{i+1}) *means that* $\delta_i^+(x_{i+1})$ *and* $\delta_i^-(y_{i+1})$ *are incremented by 1. That is,* $\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1})$ *increases by 1 and* $\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1})$ *decreases by 1. Therefore, it is obvious that the RR imbalance decreases for case 0. Next, in cases 1 and 2, the RR imbalance does not change because either* $|\delta_i^+(x_{i+1}) - \delta_i^-(x_{i+1})|$ *or* $|\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1})|$ *increases by 1, but the other decreases by 1. In case 3, because the RRB algorithm chooses a* y_{i+1} *that meets* $\delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) < 0$*, we require the condition that* $\delta_i^+(v) - \delta_i^-(v) \leq \delta_i^+(y_{i+1}) - \delta_i^-(y_{i+1}) < 0$ *for any* $v \in \bar{N}_i(x_{i+1})$*. That is,* $|\delta_i^+(v) - \delta_i^-(v)| \geq 1$ *for any* $v \in \bar{N}_i(x_{i+1})$*. Here, because* $|N_i(x_{i+1})| \leq k_i$*,* $|\bar{N}_i(x_{i+1})| \geq n - k_i - 1$*, the RR imbalance on* $\bar{N}_i(x_{i+1})$ *satisfies the following inequality:*

$$I_i(\bar{N}_i(x_{i+1})) \geq n - k_i - 1 \tag{3.1}$$

In contrast, the number of edges from $N_i(x_{i+1})$ *to* $\bar{N}_i(x_{i+1})$ *is at most* k_i^2 *because* $|N_i(x_{i+1})| \leq k_i$*; therefore, the following inequality holds by Lemma 1:*

$$I_i(\bar{N}_i(x_{i+1})) \leq k_i^2 \tag{3.2}$$

From the above two inequalities (3.1 and 3.2), $n - k_i - 1 \leq k_i^2$*. However, this contradicts the assumption of Lemma 2* $n > k_i^2 + k_i + 1$*. Therefore, case 3 cannot occur.*

In addition, the RR imbalance increases by two in case 4. Thus, we complete the proof of Lemma 2. □

Theorem 1 *We assume that $n > k_i^2 + k_i + 1$. After the i-th allocation based on the RRB algorithm is completed, the RR imbalance in graph G_i satisfies the following condition:*

$$I_i(V) \leq 4k_i^2 - 4k_i + 2$$

Proof 3 *We provide an outline of the proof and prove Theorem 1 using mathematical induction. First, using Lemma 2, we show two conditions where the RR imbalance increases during the $i+1$-th allocation. Then, we divide the student sets into $\{x_{i+1}\}$, $N_i(x_{i+1})$ and $\bar{N}_i(x_{i+1})$ and consider the number of edges between sets and in each set to derive the upper bound of the RR imbalance.*

We begin our proof of Theorem 1 by mathematical induction on the number of allocations i. The proposition clearly holds when $i = 1$. We assume that the proposition holds in the case of $i = l(\geq 2)$. $1 \leq k_l \leq k_{l+1}$; thus, the condition when $4k_l^2 - 4k_l + 2 \leq 4k_{l+1}^2 - 4k_{l+1} + 2$ is satisfied. Then, when the RR imbalance does not increase in the $l+1$-th allocation—that is, when $I_{l+1}(V) \leq I_l(V)$ is satisfied—the following condition is met:

$$I_{l+1}(V) \leq I_l(V) \leq 4k_l^2 - 4k_l + 2 \leq 4k_{l+1}^2 - 4k_{l+1} + 2$$

Therefore, from Lemma 2, we should consider only the following equation:

$$\delta_l^+(x_{l+1}) - \delta_l^-(x_{l+1}) \geq 0 \ \ \& \ \ \delta_l^+(y_{l+1}) - \delta_l^-(y_{l+1}) = 0 \tag{3.3}$$

In addition, if $\delta_l^+(x_{l+1}) = k_l$, then $k_{l+1} = k_l + 1$ holds. From Lemma 2, the RR imbalance increment is at most 2. Consequently, the following holds:

$$\begin{aligned} I_{l+1}(V) &\leq (4k_l^2 - 4k_l + 2) + 2 \\ &\leq 4(k_l+1)^2 - 4(k_l+1) + 2 \\ &= 4k_{l+1}^2 - 4k_{l+1} + 2 \end{aligned}$$

Therefore, we need to consider only the following case:

$$\delta_l^+(x_{l+1}) \leq k_l - 1 \tag{3.4}$$

Since the vertex set of graph G_l is $\{x_{l+1}\} \oplus \bar{N}_l(x_{l+1}) \oplus N_l(x_{l+1})$ (see Figure 3.3), $I_l(V) = I_l(\{x_{l+1}\}) + I_l(\bar{N}_l(x_{l+1})) + I_l(N_l(x_{l+1}))$. Subsequently, the values on the right side of the expression can be calculated individually.

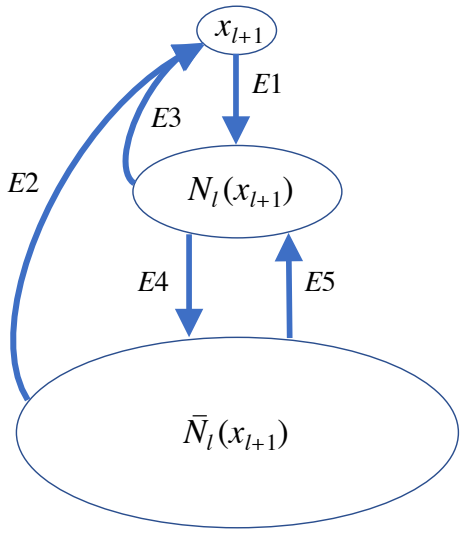


Figure 3.3. Grouping for proof of Theorem 1.

1 $I_l(\{x_{l+1}\})$*: We consider the edge sets* $E1, E2$*, and* $E3$ *in Figure 3.3. From conditions (3.3) and (3.4), the following condition holds:*

$$\begin{aligned} I_l(\{x_{l+1}\}) &= |\delta_l^+(x_{l+1}) - \delta_l^-(x_{l+1})| \\ &= \delta_l^+(x_{l+1}) - \delta_l^-(x_{l+1}) \\ &\leq \delta_l^+(x_{l+1}) \leq k_l - 1 \end{aligned}$$

2 $I_l(\bar{N}_l(x_{l+1}))$*: We consider the edge sets* $E2, E4$*, and* $E5$ *and the edges in* $\bar{N}_l(x_{l+1})$ *in Figure 3.3. From condition (3.3), the RRB algorithm selects a* y_{+1} *that meets* $\delta_l^+(y_{l+1}) - \delta_l^-(y_{l+1}) = 0$*. Then, because the RRB algorithm chooses a* $v \in \bar{N}_l(x_{l+1})$ *with the maximum* $\delta_l^+(v) - \delta_l^-(v)$*, the following condition holds:*

$$\forall v \in \bar{N}_l(x_{l+1}),\ \delta_l^+(v) - \delta_l^-(v) \leq 0 \tag{3.5}$$

Therefore, from Lemma 1, the RR imbalance on $\bar{N}_l(x_{l+1})$ *is less than* $|E4|$ *(the number of edges from* $N_l(x_{l+1})$ *to* $\bar{N}_l(x_{l+1})$*). From condition (3.4),* $|N_l(x_{l+1})| \leq k_l - 1$ *holds. Then, because the maximum outdegree is* k_l*, the following is satisfied:*

$$I_l(\bar{N}_l(x_{l+1})) \leq |E4| \leq k_l(k_l - 1) \tag{3.6}$$

3 $I_l(N_l(x_{l+1}))$*: We consider the edge sets* $E1, E3, E4$*, and* $E5$ *and the edges in* $N_l(x_{l+1})$ *in Figure 3.3. We utilize the fact that the RR imbalance on* $N_l(x_{l+1})$ *is less than the sum of the outdegree and indegree in* $N_l(x_{l+1})$*—which can be written as follows:*

$$I_l(N_l(x_{l+1})) = \sum_{v \in V} |\delta_l^+(v) - \delta_l^-(v)| \leq \sum_{v \in V} (\delta_l^+(v) + \delta_l^-(v))$$

From condition (3.4), because $|N_l(x_{l+1})| \leq k_l - 1$*, the outdegree is less than* $k_l(k_l - 1)$*, and the indegree is the sum of the edges from* $\{x_{l+1}\}$*,* $N_l(x_{l+1})$ *and* $\bar{N}_l(x_{l+1})$*.*

- *(a) Edges from* $\{x_{l+1}\}$ *(*$E1$*): From condition (3.4), the number of edges is less than* $k_l - 1$*.*
- *(b) Edges between* $N_l(x_{l+1})$*: From condition (3.4),* $|N_l(x_{l+1})| \leq k_l - 1$*. Then, the number of edges is less than* $(k_l - 1)(k_l - 2)$ *because no self-loop occurs.*
- *(c) Edges from* $\bar{N}_l(x_{l+1})$ *(*$E5$*): From condition (3.5) and Lemma 1, the following is satisfied:*

  $$I_l(\bar{N}_l(x_{l+1})) = |E4| - (|E2| + |E5|) \geq 0$$

  Therefore, from condition (3.6), $|E5| \leq |E2| + |E5| \leq |E4| \leq k_l(k_l - 1)$ *holds.*

Hence, the sum of the indegree is less than $(k_l - 1) + (k_l - 1)(k_l - 2) + k_l(k_l - 1) = 2k_l^2 - 3k_l + 1$*. Then, the sum of the outdegree and indegree is less than* $k_l(k_l - 1) + 2k_l^2 - 3k_l + 1 = 3k_l^2 - 4k_l + 1$*, and* $I_l(N_l(x_{l+1})) \leq 3k_l^2 - 4k_l + 1$*.*

Therefore, after the l*-th allocation, the following condition holds:*

$$I_l(V) \leq k_l - 1 + k_l(k_l - 1) + 3k_l^2 - 4k_l + 1 = 4k_l^2 - 4k_l$$

The RR imbalance increment is 2 from Lemma 2, and $k_l = k_{l+1}$ *because of condition (3.4); thus, the following condition is satisfied after the* $l + 1$*-th allocation:*

$$I_{l+1}(V) \leq 4k_l^2 - 4k_l + 2 = 4k_{l+1}^2 - 4k_{l+1} + 2$$

which concludes the proof of Theorem 1. □

Based on the above proof, when using the RRB algorithm, the upper bound of the RR imbalance in the graph G_i is $\mathcal{O}(k^2)$, when the maximum outdegree of graph G_i is k and the number of students exceeds k^2+k+1. By Theorem 1, even if the number of students is large, when $k = 5$, we can know beforehand that the upper bound becomes $4 \cdot 5^2 - 4 \cdot 5 + 2 = 82$.

3.5 Experiments

We experimentally compare the proposed algorithms under the adaptive allocation approach to algorithms under the existing nonadaptive allocation approach using two types of data. First, we describe the data characteristics, and then, we describe baselines and present the experimental results.

3.5.1 Simulation Data based on Real Data

We use the simulation data based on the data published by Canvas Network*. This data is comprised of de-identified data from March 2014 - September 2015 of Canvas Network open courses. However, this data has the following two problems:

- Although the data includes the reviewing number for each student, it does not include information concerning whose submissions the student reviewed.
- Moreover, the data does not include the reviewing order.

Therefore, in this experiment, we complement the data in several ways, as described below.

In our experiments, we utilize those data whose class ID is 770000832960949 and whose assignment ID is 770000832930436 (denoted as real data 1) and those data whose class ID is 770000832945340 and assignment ID is 770000832960431 (denoted as real data 2). Specifically, we extract the submission ID, the ID of the student who commented on the submission (the reviewer ID), and the volume of comments from the table called $submission\_comment\_fact$.

In this study, we regard that the more comments a reviewer writes, the higher his reviewing ability is. Therefore, we define the reviewing ability as follows: We take the average of the aggregated volume of comments for each reviewer and

*https://dataverse.harvard.edu/dataset.xhtml?persistentId =doi:10.7910/DVN/XB2TLU

then set the reviewing ability value to 0.2, 0.4, 0.6, 0.8 and 1.0 based on the ascending order of the aggregated average. Note that the numbers of reviewers with each reviewing ability value are adjusted to be as equal as possible.

In the Canvas Network data, because the submission ID is not linked with the ID of the student, it is impossible to determine whose submission a student reviewed. This situation occurs because of the anonymization process to prevent data disclosure. Thus, it is impossible to calculate the actual RR imbalance and ARR imbalance using the real data. Therefore, in this research, we measure the effectiveness of the proposed methods through simulations based on the real data.

In addition, it is not possible to read the strict reviewing order from the Canvas Network data. Therefore, we construct the reviewing order using the following two methods and measure:

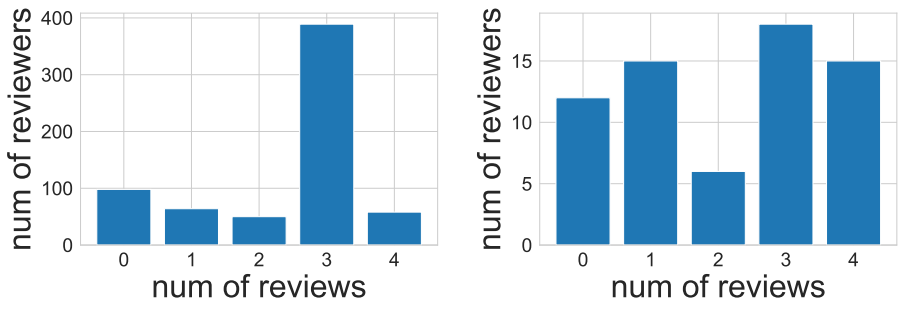
Construct the reviewing order based on the time when the comments were created

This method uses the timestamp in the table called *submission_comment_dim* for when the comment was created, that is, when the review is completed. We need the time when the review is started to construct the accurate reviewing order, but in this method, we arrange the reviewer IDs in decreasing order based on the available timestamp instead. The data complemented based on this method are considered to be the most realistic data used in this experiment.

Construct the reviewing order based on reviewer transition model

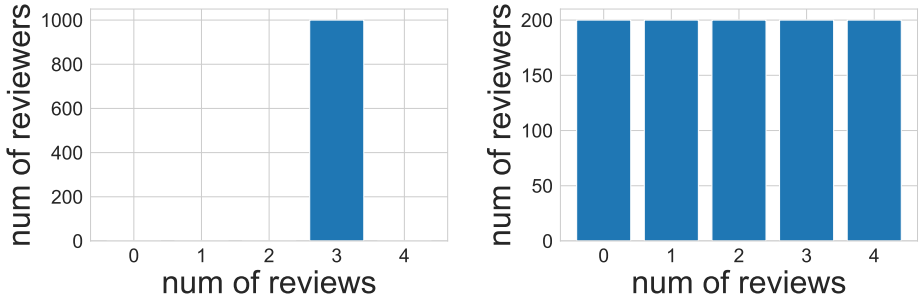
We set the probability that reviewer x_{i+1} is the same as the previous reviewer x_i to P, and arrange the reviewing IDs according to this probability. Note that, when the previous reviewer x_i cannot review another submission, the reviewer x_{i+1} is randomly selected regardless of x_i. For example, when $P = 0$, reviewer x_{i+1} is randomly chosen regardless of the previous reviewer x_i, and when $P = 1$, reviewer x_{i+1} is selected to be the previous reviewer x_i.

Figure 3.4 shows a plot of the number of reviewers for each number of reviews from the datasets real data 1 and real data 2. Note that, in real data 1, one reviewer performed 25 reviews just before the end of peer assessment. We consider this value as representing a lecturer or TA who reviewed the student submissions whose reviewed number is insufficient. Thus, we replaced the reviewing number of



(a) real data 1 (b) real data 2

Figure 3.4. The number of reviewers for each number of reviews from the real data.



(a) biased data (b) flat data

Figure 3.5. The number of reviewers for each number of reviews from the synthetic data.

this reviewer with 3, the mode value of the reviewing number from the real data 1 dataset. In addition, no information is available for reviewers who did not review any submission from table *submission_comment_fact*. Therefore, we instead use the value obtained by subtracting the total number of reviewer IDs from the total number of submission IDs as the number of reviewers whose reviewing number is 0.

3.5.2 Simulation Data based on Synthetic Data

We use the simulation data based on the following two types of synthetic data. The first dataset includes only those reviewers whose reviewing number is 3 (we

term this the biased data), and the second dataset includes those reviewers whose reviewing numbers are uniformly between 0 and 4 (we term this the flat data). In both datasets, the total number of students is 1000 (see Figure 3.5). The biased dataset can be regarded as an extreme example of data with the similar tendency as that of real data 1, and the flat data can be regarded as an extreme example of data with the similar tendency as that of real data 2. We set the reviewing ability value to $0.2, 0.4, 0.6, 0.8$, and 1.0 randomly as the numbers of reviewers with each reviewing ability value are adjusted to be equal. The reviewing order is simulated based on the reviewer transition model described in Section 3.5.1.

3.5.3 Comparison Methods

As comparison methods, we utilize two algorithms under a nonadaptive approach in which the number of submissions that a student should review is typically fixed, and submissions are allocated to students before assessment starts. Most of the existing peer assessment methods adopt this approach. We assume that students can request an additional submission after they complete reviewing the allocated submissions like the existing work [52].

To utilize nonadaptive approach, we need to set the number of submissions allocated to one student, but that actual number in each real data is not available. For the simulation of real data 1, the most natural approach may be to set the number of submissions allocated to a student to 3 in the comparison methods. For the simulation of real data 2, it is difficult to determine a fixed number of submissions that should be allocated to a student; however, we also set the value to 3 in our experiments. Note that, we assume that students who review 4 submissions request an additional submission. In addition, each synthetic data is an extreme case of each real data; hence we set the number of submissions that a student should review to 3. In this case, the descending order of dropout rate is considered to be the rate in the flat data, real data 2, real data 1 and biased data datasets. In particular, no one dropout data exists in the biased data dataset.

The detail of the comparison methods are as follows:

Naive allocation algorithm in the nonadaptive approach

This algorithm adopts random allocations so that both the reviewing number and the reviewed number for all students are 3. We denote this algorithm as Random.

Table 3.1. Experimental results on the simulation based on real data 1 using the time when the comments were created.

	RRB	ARRB	Random	LPT
RR imbalance	12	10	724	792
ARR imbalance	80.1	66.6	810.7	863.5

Table 3.2. Experimental results on the simulation based on real data 2 using the time when the comments were created.

	RRB	ARRB	Random	LPT
RR imbalance	12	14	90	94
ARR imbalance	18.9	19.9	98.2	100.7

Ability-aware allocation algorithm in nonadaptive approach

This algorithm approximately allocates as the dispersion between the total reviewing ability values of the reviewers allocated to individual submissions becomes small, and such that the reviewing number and the reviewed number for all students are 3. This algorithm is based on an allocation algorithm called Longest Processing Time [53]; hence, we denote this algorithm as LPT.

3.5.4 Experimental Results

Real data with the reviewing order in Section 3.5.1

We apply two algorithms and two comparison algorithms to the real data 1 and real data 2 datasets whose reviewing order is constructed based on the creation date and time of the reviewing comments. In this experiment, the parameter λ, which is used for the ARR imbalance and ARRB, is set to 1. The results are shown in Tables 3.1 and 3.2. Small values are preferable for both RR imbalance and ARR imbalance; therefore, the above results show that the proposed algorithms work more effectively than do the existing algorithms. In addition, because the maximum reviewing number is 4 in the real data 1 and 2 datasets, the upper bound of RR imbalance (as described in Section 3.4) is 50, and the results are satisfied with this upper bound.

ARRB obtains results that are superior to RRB regarding both RR imbalance and ARR imbalance on the real data 1 dataset. In contrast, RRB is superior

to ARRB regarding both RR imbalance and ARR imbalance on the real data 2 dataset. These results do not consist with the aim of RRB and and ARRB. We compare and examine RRB and ARRB in more detail in subsequent experiments.

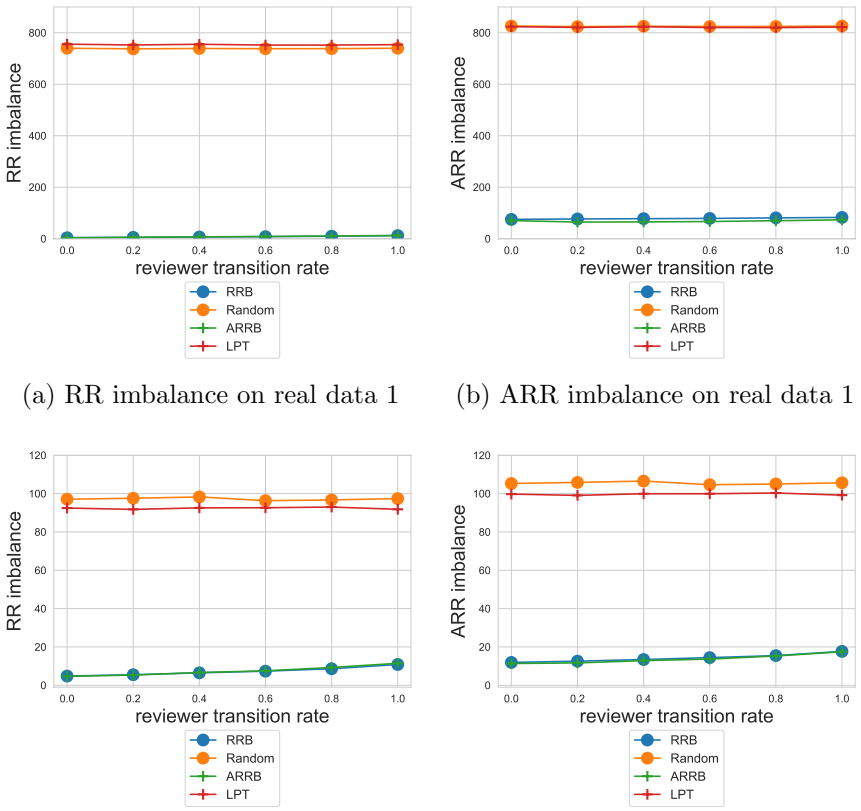
Real data with the reviewing order in Section 3.5.1

We use the real data 1 and real data 2 datasets whose reviewing order is constructed based on the reviewer transition model, in which the probability P is 0, 0.2, 0.4, 0.6, 0.8, or 1. The parameter λ, which is used for ARR imbalance and ARRB, is set to 1. For each method, we generate 100 reviewing order and apply the algorithms to these data. We obtain the average value of RR imbalance and the average value of ARR imbalance.

The results are shown in Figure 3.6. The vertical axis represents the RR imbalance or the ARR imbalance, and the horizontal axis represents the probability value P. The four types of lines plotted in each figure represent the following.

- *RRB*: Imbalance using RRB (blue)
- *Random*: Imbalance using Random (orange)
- *ARRB*: Imbalance using ARRB (green)
- *LPT*: Imbalance using LPT (red)

Figure 3.6 shows that the performance of the two proposed algorithms greatly exceeds those of the two existing algorithms. We can confirm that the upper bound of RR imbalance discussed in Section 3.4 is established. We can also see that the performances of the two proposed algorithms deteriorate when the probability P is high—that is, the same reviewers continue reviewing. Although the RRB algorithm tries to minimize the RR imbalance and the ARRB algorithm tries to minimize the ARR imbalance, there is no discernable performance difference between the two algorithms from the results shown in Figure 3.6(a)(c)(d). However, we can observe that ARRB is superior to RRB in Figure 3.6(b). This experiment suggests that ARRB can reduce the ARR imbalance further than can RRB while achieving an RR imbalance equally as good as that of RRB.



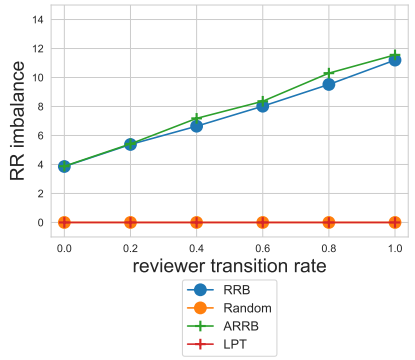
(a) RR imbalance on real data 1 (b) ARR imbalance on real data 1

(c) RR imbalance on real data 2 (d) ARR imbalance on real data 2

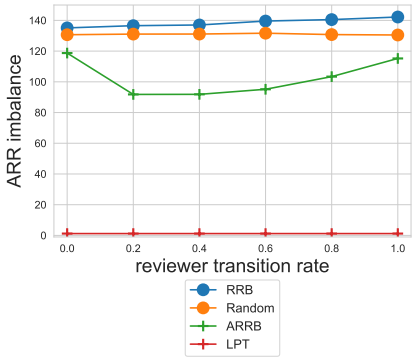
Figure 3.6. Experimental results on real data complemented by the reviewer transition model when $\lambda = 1$.

Synthetic data with the reviewing order in Section 3.5.1

We conducted an experiment similar to the second experiment but on the biased data and the flat data. The result is shown in Figure 3.7. We can confirm that the upper bound of the RR imbalance described in Section 3.4 is established. In the case of flat data, the proposed algorithm greatly outperforms the existing algorithm, but in the biased data the proposed algorithm's performance is inferior to that of the existing algorithm. This result occurs because all the submissions allocated before assessment are reviewed, that is, there is no dropout; thus, the RR imbalance is always 0. In fact, we can see from Figure 3.7(a) that the RR imbalance remains at 0 with the Random algorithm. In addition, as shown in Figure 3.7(b), the ARR imbalance is the smallest with LPT. However, when

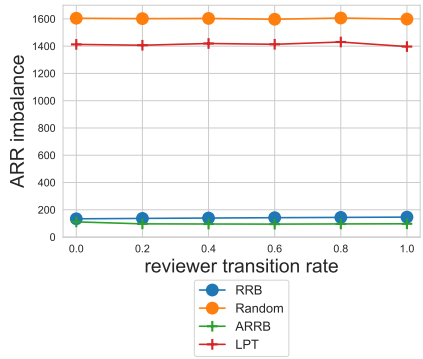


(a) RR imbalance on biased data



(b) ARR imbalance on biased data

(c) RR imbalance on flat data



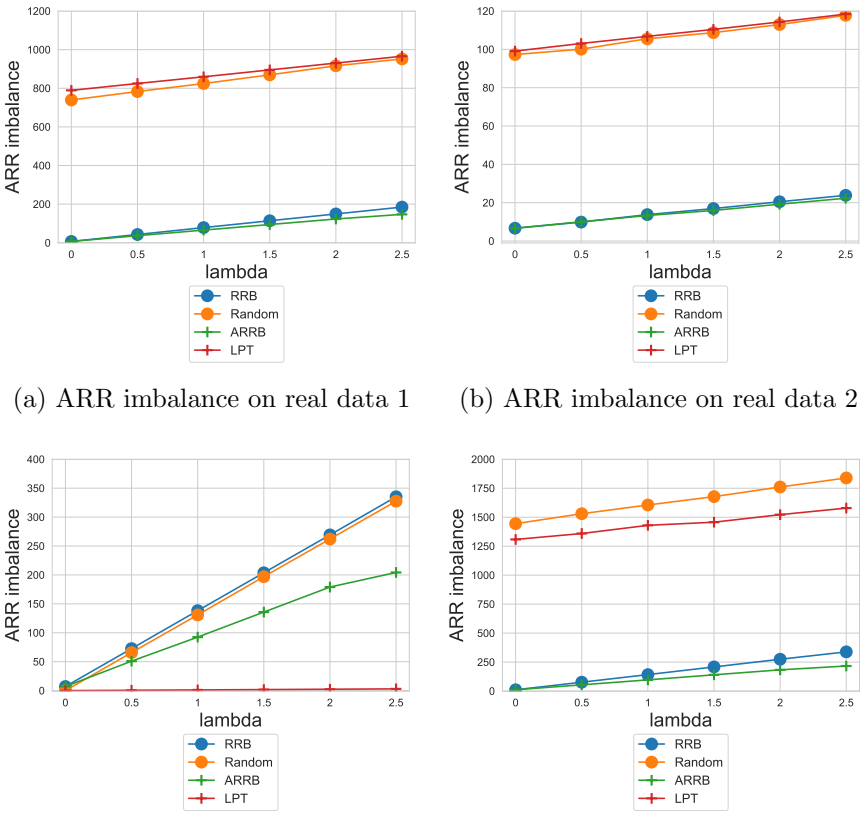
(d) ARR imbalance on flat data

Figure 3.7. Experimental results on synthetic data complemented by the reviewer transition model when $\lambda = 1$.

many students have the same number of reviews and only a few students with different reviewing numbers exist, as in real data 1 (see Figure 3.4(a)), the results using Random and LPT become worse (see Figure 3.6(a)(b)). Therefore, under nonadaptive allocation, the Random and LPT algorithms work effectively only in certain special situations. In addition, in Figure 3.7(a)(c), little performance difference between the two proposed algorithms can be observed, and as Figure 3.7(b) and (d) show, ARRB is superior to RRB.

Experiments for λ in ARR imbalance

We fixed the transition rate P to 0.5 and varied λ using the values 0, 0.5, 1.0, 1.5, 2.0, and 2.5. We obtained the ARR imbalance values for the two real and two syn-



(a) ARR imbalance on real data 1

(b) ARR imbalance on real data 2

(c) ARR imbalance on biased data

(d) ARR imbalance on flat data

Figure 3.8. Experimental results on real data and synthetic data complemented by the reviewer transition model when the transition rate is $P = 0.5$.

thetic datasets. The results are shown in Figure 3.8. In Figure 3.8(a)(b)(d), the proposed algorithms are superior to the existing algorithms and ARRB is superior to RRB, similar to the previous experiments. In addition, as shown in Figure 3.8(c), the LPT algorithms work effectively in the biased data, but this occurs only in special situation as mentioned in the third experiment. We also find that, when λ is larger, the difference between ARR imbalance by ARRB and that by RRB tends to become larger. This is because ARRB considers the fairness in the reviewing ability and that is emphasized when λ is large.

The results from all four experiments suggest that the proposed algorithm outperforms the existing algorithms in many cases. In addition, ARRB performs comparably to RRB with respect to RR imbalance and achieves better perfor-

mance with respect to ARR imbalance.

3.6 Conclusion

In this study, we propose the allocation algorithms RRB and ARRB to achieve fair peer assessment with respect to the number and contents of reviews using an adaptive allocation approach and considering a situation where dropout can occur during peer assessment. We analyze the RRB algorithm theoretically and show its robustness. We also confirm the usefulness of the proposed allocation algorithms through experiments using both real and synthetic data. In future work, we plan to study how to estimate the reviewing ability values from the students' past behavioral data.

CHAPTER 4

Analysis of the Effect of Student-Submission Allocation on Peer Assessment Accuracy

In peer assessment, student reliability is regarded as a problem; consequently, various methods of estimating highly reliable grades from scores given by multiple students have been proposed. However, student-submission allocation for peer assessment has not been well studied. In this chapter, we analyzed the effect of student-submission allocation on score estimation in peer assessment. We deal with three types of allocation methods; random allocation, circular allocation and group allocation, which are considered to be commonly used in peer assessment. Through simulation experiments, we show that circular allocation and group allocation tend to yield lower accuracy than random allocation does.

4.1 Motivation

Peer assessment has a problem of low reliability because it relies on students' reviews. Therefore, in peer assessment, a reviewing criterion called a rubric is often used, and a single student's submission is reviewed by multiple other students [3, 7]. In addition, methods of using statistical models to estimate a sin-

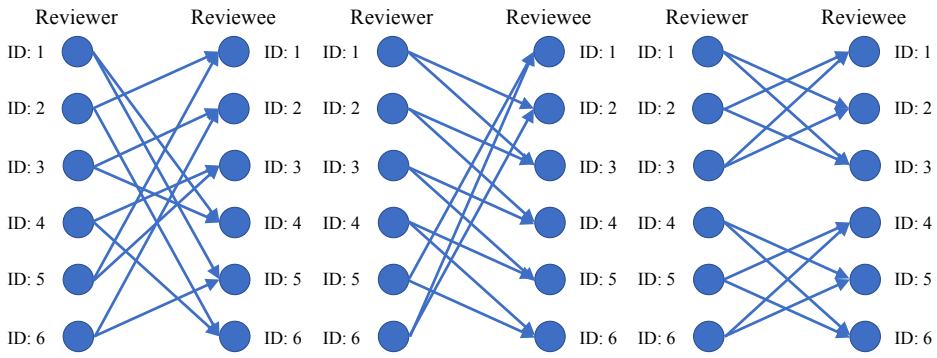


Figure 4.1. Allocation patterns.

gle reliable score by combining the scores given by multiple students have been proposed [5, 6, 15, 41]. However, the related studies have not focused on student-submission allocation. In this study, we analyze the effect of student-submission allocation on score estimation.

In particular, this study focuses on student-submission allocation patterns that satisfy the following conditions:

1. A student cannot grade his or her own submission.
2. A student cannot grade the same submission twice.
3. Each student reviews the same total number of submissions, and each submission is reviewed by the same total number of students.

We call these conditions the basic principles of student-submission allocation. Note that we refer to the number of submissions a student reviews as the "reviewing number" and to the number of students who review a single submission as the "reviewed number". The first and second of the basic principles are obvious. The third is generally applied to avoid unfairness in the reviewing number and the reviewed number among the students.

We analyze three allocation methods: random allocation, circular allocation, and group allocation. In this study, we provide empirical evidence to the effectiveness of these allocation methods. These allocation methods are depicted in Figure 4.1. Each node represents a student; the left side of each bipartite graph represents the reviewers, and the right side represents the reviewees. Note that

nodes to which the same ID is assigned on the left side and the right side represent the same student. Each edge drawn from a reviewer to a reviewee represents a student-submission allocation. For example, in Figure 4.1 (a), the student with ID 1 reviews the submissions of the students with ID 4 and ID 5. In Figure 4.1, each student's reviewing number and reviewed number are 2, satisfying the third condition described above.

Figure 4.1 (a) illustrates random allocation. Random allocation is a method of simply allocating students to submissions randomly while satisfying the above conditions.

Figure 4.1 (b) illustrates circular allocation. Circular allocation is a method in which a certain order relation is assigned to the students (in this case, an order relation based on ID), and each student is allocated to review the submissions of the next k students, where k is the reviewing number. In this figure, the student with ID 1 reviews the submissions of the students with ID 2 and ID 3, the student with ID 2 reviews the submissions of the students with ID 3 and ID 4, and so on. This allocation method can easily satisfy the basic principles and is often used in actual peer assessment.

Figure 4.1 (c) illustrates group allocation. In group allocation, the students are divided into several sets, and student-submission allocations are generated within each set. In this figure, six students are divided into two sets of three students each, and allocation is performed within each of these sets. This method is also often applied to satisfy the basic principles. More specifically, if each student's reviewing number (and reviewed number) is k, the students are first divided into sets, each of which consists of $k + 1$ students. Then, the basic principles can be achieved by allocating each student to all submissions other than his or her own within his or her assigned group. In addition, group allocation is often used in an effort to divide a class in consideration of the students' profiles [67].

In the experiments reported in this paper, we applied the above allocation methods to artificial data and real data and applied a typical score estimation method proposed by Piech et al. [5]. Then, we compared the allocation methods using the root mean square error (RMSE) as the evaluation index. Our experimental results show that circular allocation and group allocation tend to yield lower accuracy than random allocation does.

The remainder of this work is organized as follows. Section 4.2 describes the setting of our experimental analysis. In Section 4.3, we present the experimental

results. Finally, we conclude this work and suggest future work in Section 4.4.

4.2 Setting

This study analyzes the effects of three types of student-submission allocation on score estimation. This section first describes the three student-submission allocation algorithms considered: random allocation, circular allocation, and group allocation. Then, we introduce the existing score estimation method applied in our experiments.

4.2.1 Allocation Algorithms

This study focuses on student-submission allocation patterns that satisfy the basic principles described in Section 4.1. The number of students is n, and the reviewing number and reviewed number are both k ($< n$). We consider a set of students $V = \{v_1, \ldots, v_n\}$. Additionally, we consider a graph with the student set V as the node set, where the set of directed edges of this graph is denoted by E. Note that a directed edge (v_i, v_j) indicates that student v_i is allocated to review student v_j's submission. The following algorithms take as input a student set V and a reviewing number (reviewed number) k and output an edge set E that represents the student-submission allocations.

Random Allocation

The random allocation algorithm (Algorithm 1) generates allocations that satisfy the basic principles randomly and sequentially. Note that since the candidates to which a given student v_i can be allocated change during the sequential allocation process, they are managed by the variable $C(v_i)$. Specifically, given a Graph $G(V, E)$, $C(v_i)$ is the set of students, excluding v_i him- or herself, to whose submissions v_i has not yet been allocated and whose submissions have been assigned to be reviewed by fewer than k students each, as follows:

$$C(v_i) = \{v_j | v_j \neq v_i, (v_i, v_j) \notin E, R(v_j) < k\}$$

Here, $R(v_j)$ represents the number of students who are reviewing the submission of v_i.

Algorithm 1 Random Allocation Algorithm

INPUT: $V = \{v_1, \ldots, v_n\}$ ▷ a set of n students
INPUT: k ▷ reviewing (reviewed) number
OUTPUT: E ▷ student-submission allocations

1: $E \leftarrow \{\}$
2: **for** $i \leftarrow 1$ to n **do**
3: **for** $t \leftarrow 1$ to k **do**
4: **if** $C(v_i)$ is empty **then**
5: **go to** 1
6: **end if**
7: v_j is selected from $C(v_i)$ at random
8: $E \leftarrow E \cup \{(v_i, v_j)\}$
9: **end for**
10: **end for**

Due to greedy allocation, the candidate set $C(v_i)$ may be empty. For example, when v_n is being assigned to submissions and the only candidate submission that is not already being reviewed by v_n is the submission of v_n him- or herself, $C(v_n)$ becomes empty. In this case, this algorithm terminates, as shown in the fifth line, and the allocation process repeats from the beginning.

Circular Allocation

The circular allocation algorithm (Algorithm 2) allocates a given student to the submissions of students with adjacent IDs.

Algorithm 2 Circular Allocation Algorithm

INPUT: $V = \{v_1, \ldots, v_n\}$ ▷ a set of n students
INPUT: k ▷ reviewing (reviewed) number
OUTPUT: E ▷ student-submission allocations

1: $E \leftarrow \{\}$
2: **for** $i \leftarrow 1$ to n **do**
3: **for** $j \leftarrow 1$ to k **do**
4: $E \leftarrow E \cup \{(v_i, v_{((i+j) \mod n)})\}$
5: **end for**
6: **end for**

This algorithm differs from the random allocation algorithm in that the output is fixed to one type and there is no need to terminate.

Group Allocation

The group allocation algorithm (Algorithm 3) divides students into several groups and then performs random allocation in each group. In the following algorithm, the number of groups is represented by d. Note that the group allocation algorithm considers only cases in which n/d is an integer for simplicity.

Algorithm 3 Group Allocation Algorithm

INPUT: $V = \{v_1, \ldots, v_n\}$ ▷ a set of n students
INPUT: k ▷ reviewing (reviewed) number
INPUT: d ▷ number of groups
OUTPUT: E ▷ student-submission allocations

1: $E \leftarrow \{\}$
2: $l \leftarrow n/d$ ▷ group size
3: **for** $i \leftarrow 1$ to d **do**
4: $\quad V' \leftarrow \{v_{(l\cdot(i-1)+1)}, \ldots, v_{l\cdot i}\}$
5: $\quad E \leftarrow E \cup RandomAllocationAlgorithm(V', k)$
6: **end for**

If d and k are set such that $n/d = k + 1$, an algorithm that satisfies the basic principles can be easily realized in a manner similar to the circular allocation algorithm.

4.2.2 Estimation Method

In this study, we focus on the statistical estimation model known as PG1 [5]. We choose this model because it is the most basic model, as described in Section 2.2.4. The details of PG1 are as follows:

$$
\begin{aligned}
(Reliability)\ \tau_v &\sim \mathcal{G}\left(\alpha_0,\ \beta_0\right) \\
(Bias)\ b_v &\sim \mathcal{N}\left(0,\ \frac{1}{\eta_0}\right) \\
(True\ score)\ s_u &\sim \mathcal{N}\left(\mu_0,\ \frac{1}{\gamma_0}\right) \\
(Observed\ score)\ z_u^v &\sim \mathcal{N}\left(s_u + b_v,\ \frac{1}{\tau_v}\right)
\end{aligned}
$$

$\mathcal{G}$ is a gamma distribution with fixed hyperparameters α_0 and β_0, while η_0 and γ_0 are the hyperparameters for the priors over the biases and true scores, respectively. τ_v represents the reliability of student v, and b_v represents the bias of student v. s_u represents the true score of the submission created by student u, and z_u^v represents the score of the submission of student u as reviewed by student v. We estimate τ_v, b_v, and s_u in accordance with this model given each student's reviewing scores z_u^v.

In the estimation process performed in this study, Gibbs sampling was used, with α_0, β_0, η_0, and γ_0 all set to 1. The number of iterations was 3,000, and the first 1000 iterations were used for burn-in. PyMC3 was used for the implementation.

4.3 Experiments

In this section, we first show the experimental results on the artificial dataset, then we give the results on the real dataset. Note that since the size of the real dataset is small, the experimental results on the real dataset is less reliable than the results on the artificial dataset. The reason why we utilize the small dataset is described in Section 4.3.2.

4.3.1 Simulation on the Artificial Dataset

We first explain the artificial dataset used in our experiments. Then we give an experimental overview and compare the results obtained with the above three algorithms.

Table 4.1. RMSE results on the artificial dataset + random allocation simulation.

n	PG1(3)	PG1(5)	PG1(10)	avg(3)	avg(5)	avg(10)	PG1/avg(3)	PG1/avg(5)	PG1/avg(10)
5	0.482			0.504			0.971		
10	0.487	0.375		0.519	0.399		0.946	0.948	
20	0.478	0.374	0.259	0.518	0.406	0.279	0.926	0.929	0.937
50	0.482	0.363	0.249	0.524	0.405	0.286	0.921	0.900	0.873
100	0.485	0.364	0.244	0.526	0.409	0.287	0.923	0.891	0.852
1000	0.483	0.361	0.241	0.527	0.408	0.289	0.918	0.885	0.835

Table 4.2. RMSE results on the artificial dataset + circular allocation simulation.

n	PG1(3)	PG1(5)	PG1(10)	avg(3)	avg(5)	avg(10)	PG1/avg(3)	PG1/avg(5)	PG1/avg(10)
5	0.487			0.511			0.968		
10	0.489	0.375		0.516	0.397		0.954	0.959	
20	0.496	0.376	0.262	0.523	0.401	0.284	0.950	0.948	0.928
50	0.497	0.384	0.261	0.525	0.408	0.287	0.949	0.946	0.917
100	0.496	0.381	0.261	0.525	0.406	0.289	0.946	0.942	0.909
1000	0.500	0.385	0.263	0.527	0.408	0.289	0.948	0.943	0.910

Artificial Dataset

For the artificial dataset, τ_v was generated as a uniform random number from 1 to 2, b_v was generated as a uniform random number from -1 to 1, and s_u was generated as a uniform random number from 1 to 5. Then, z_u^v was generated in accordance with the fourth equation of the PG1 model. Note that the range of s_u was set to 1 to 5 to consider a five-level evaluation. In addition, we set τ_v and b_v such that z_u^v would vary within approximately one level below or above s_u. For each simulation, we generated 500 data subsets in accordance with the specified number of students n, reviewing number k, and allocation algorithm.

Experimental Overview

The simulation results on the artificial dataset are shown in Tables 4.1 and 4.2. In these experiments, we performed 500 simulations and obtained the average RMSEs of the estimated values while varying the number of students n, the reviewing number k, and the allocation pattern as follows: random allocation (Table 4.1) or circular allocation (Table 4.2).

The reason why we did not explicitly perform group allocation is as follows. Group allocation is an allocation method in which the student set is divided into

small groups and then random allocation is performed in each group. Therefore, when there are two sets of allocations with the same reviewing number, but one has a larger total number of students than the other, the former can be interpreted as random allocation, while the latter can be interpreted as group allocation. Accordingly, in this study, to evaluate the performance of group allocation, we used the results of random allocation with a small number of students instead of results obtained explicitly through group allocation.

Now, let us explain how to read the experimental result tables, using Table 4.1 as an example. The leftmost column represents the number of students n, and each row represents the results obtained using data generated under the assumption of n students. The second to fourth columns (PG1(k), $k = 3, 5, 10$) show the average RMSEs of the estimated values when PG1 is applied to the generated 500 subsets of data. The values in the second column (PG1(3)) are the results for a reviewing number of 3, the values in the third column (PG1(5)) correspond to $k = 5$, and the values in the fourth column (PG1(10)) correspond to $k = 10$.

The fifth to seventh columns (avg(k)) show the average RMSEs of the simple average ($\hat{s}_u = \sum_v z_u^v$). These three columns similarly present the results for $k = 3, 5, 10$.

The 8th to 10th columns (PG1/avg(k)) show the average values obtained by dividing the RMSE obtained with PG1 by the RMSE obtained through simple averaging for the 500 data subsets. The reason why we derive not only the average RMSE value of PG1 (PG1(k)) but also the average value of the ratio between the RMSEs of PG1 and simple averaging is as follows: When artificial data generated from the same distribution are used, the expected RMSE value of the simple average is independent of n. However, as seen from the avg(k) columns in Table 4.1, the RMSE value is smaller when n is smaller. Therefore, we consider that some kind of sampling error occurred due to the use of the RMSE, and therefore, we need to normalize out this effect.

First, we will compare random allocation and circular allocation on artificial data based on Tables 4.1 and 4.2. Next, we will compare random allocation and group allocation using Table 4.1.

Table 4.3. RMSE results on the real dataset + random allocation simulation.

n	PG1(3)	PG1(5)	PG1(7)	avg(3)	avg(5)	avg(7)	PG1/avg(3)	PG1/avg(5)	PG1/avg(7)
4	1.555			1.467			1.080		
6	1.547	1.349		1.466	1.231		1.066	1.105	
8	1.582	1.392	1.299	1.493	1.276	1.175	1.066	1.096	1.108
10	1.565	1.394	1.258	1.496	1.283	1.153	1.050	1.092	1.096

Random Allocation vs. Circular Allocation

As seen by comparing the values in the corresponding cells in the PG1/avg columns of Tables 4.1 and 4.2, the values in Table 4.1 are smaller in most cases. In particular, as n increases, the difference becomes larger. This finding indicates that random allocation is superior to circular allocation.

The cause of this difference in performance might be explained in the following manner. When we ignore the direction of edges in allocation graph, the distances between nodes in the circular allocation graphs generally tend to be larger than the distances between nodes in the random allocation graphs. Since the PG1 model can be interpreted as recursively using adjacent values in the allocation graph at the time of estimation, the greater the distance between nodes is, the more adversely the estimation may be affected.

Random Allocation vs. Group Allocation

We compare random allocation and group allocation based on Table 4.1. As seen from Table 4.1, the larger n is, the smaller the values of PG1/avg, indicating that random allocation without division of the students is superior to group allocation. Considering the recursive estimation method of the PG1 model, the available information increases as the value of n increases; hence, PG1 may function more effectively as a result.

4.3.2 Simulation on the Real Dataset

We explain the real dataset, then show an experimental results. The real dataset used in our experiment is small, so the results on the real dataset is less reliable than those on the artificial dataset. Note that we did not compare random allocation with circular allocation using real dataset. This is because when the number of students n is small (e.g. $n = 5$ or 10), considering the simulation results

on artificial data, there is no significant difference between random allocation and circular allocation (see the results in Section 4.3.1).

Real Dataset

For the real data, we used an open word pair similarity dataset [68]. Although this dataset was collected in the context of crowdsourcing research, the situation in which low-skilled workers review multiple tasks is similar to a peer assessment setting; therefore, it seems reasonable to use it in this experiment. In this dataset, workers can be interpreted as reviewers, and tasks can be interpreted as submissions. Note that the set of reviewers and the set of reviewees who created the submissions are not the same in this case, but this does not pose a problem for PG1 because this estimation method does not assume that the reviewer set and the reviewee set are the same. In addition, this dataset consists of a total of 300 scores assigned to all 30 tasks by 10 workers. The data size is small, but the allocation graph is very dense. When performing a simulation using the real dataset, restoration extraction was performed 500 times from the data such that the specified n and k were satisfied. Therefore, it was desirable for the allocation graph of the original dataset to be tightly connected, but difficult to create large data with such an allocation graph. Hence, we considered the word pair similarity dataset to be suitable for our purposes, though the size is small.

Random Allocation vs. Group Allocation

We perform a comparison between random allocation and group allocation on the real dataset. The results of the simulation are shown in Table 4.3. How to read the table is the same as in simulation on the artificial dataset. We consider the cases of $n = 4, 6, 8, 10$ and $k = 3, 5, 7$. The performance of PG1 is inferior to the performance of the simple average because the data are too few, but the ratio between PG1 and the simple average decreases as n increases. Therefore, it is suggested that random allocation is superior to group allocation based on this dataset.

All of the experimental results reported in Section 4.3 suggest that random allocation is superior to both circular allocation and group allocation.

4.4 Conclusion

In this chapter, we have demonstrated that circular allocation and group allocation, both of which are often used in peer assessment, have a bad effect on the estimation results when using a typical statistical score estimation method. Since our study offers only experimental results, we plan to further consider this issue from a theoretical perspective in the future. In addition, it should be noted that student reviewing is not always performed in accordance with the predefined allocations due to dropout; therefore, we also plan to consider the effect of allocation with dropout on scoring accuracy.

CHAPTER 5

Adaptive Peer Assessment Allocation considering Fairness and Accuracy

Chapter 3 and Chapter 4 have discussed student-submission allocation in peer assessment, each focusing on different objectives. In Chapter 3, we developed an allocation method (RRB algorithm) to solve the imbalance of the number of reviews due to dropout. In Chapter 4, we analyzed what kind of student-submission allocation improves peer assessment accuracy. We concluded that the different allocation methods are superior in each study due to the difference of the objectives. In this chapter, we point out that the allocation method proposed in Chapter 3 may adversely affect peer assessment accuracy in certain situations. Then, we propose a method that considers the trade-off between fairness based on RR imbalance and peer assessment accuracy. The proposed method is to replace a part of the allocation in the RRB algorithm with random allocation. This study asserts the usefulness of the proposed method through simulation. In addition, we discuss the usefulness of the methodology of replacing part of the allocation that worsens the estimation accuracy with random allocation.

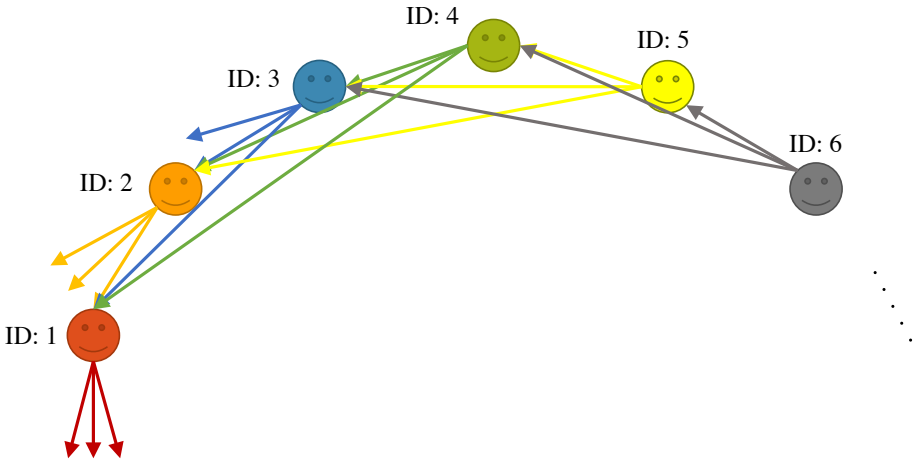


Figure 5.1. Circular allocation-like example where RRB is applied.

5.1 Motivation

In Chapter 3, we proposed an adaptive allocation method (RRB algorithm) to solve the imbalance in the number of reviews due to dropouts. In the RRB algorithm, students request to review a submission, then a submission is allocated to the requesting student. Note that the submission of the student who contributes the most at each point in time, that is, the student whose difference between his or her reviewing number and reviewed number is the largest, is allocated with priority. In Chapter 3, we proved that the sum of the absolute value of the difference between the reviewing number and the reviewed number (RR imbalance) can be limited to a certain amount when using the RRB algorithm. However, this allocation method does not take into account the effect on the peer assessment accuracy as in Chapter 4. We point out that the RRB algorithm may have an adverse effect on peer assessment accuracy.

When students review multiple submissions, students often review them collectively. Therefore, when applying RRB algorithm to practical settings, students are expected to make requests continuously.

We consider an extreme situation where all students request submissions in succession. At this time, student-submission allocation is as shown in Figure 5.1. Note that each node represents a student, and each edge represents the allocation from the reviewer to the reviewee. Also, in this figure, it is assumed that the students request three submissions in succession in the clockwise order

from the student with ID 1.

First, when the student with ID 1 requests submissions, the difference between the reviewing number and reviewed number of all the student is 0 or less, so three submissions of other students are randomly allocated to the student with ID 1. In this figure, we omit the nodes which are selected randomly in RRB algorithm. In addition, the submissions of students with IDs 1-6 are not subject to random allocation. Subsequently, when the student with ID 2 requests the submission, the difference between the reviewing number and the reviewed number of the student with ID 1 is the largest (= 3), so the submission of the student with ID 1 is allocated first to the student with ID 2, and then two other submissions are randomly allocated. The student with ID 3 is allocated to the submissions of ID 1 and ID 2 first, then one submission is randomly allocated. The student with ID 4 is allocated to the submissions of ID 1, ID 2 and ID 3 with priority. Then, after the student with ID 4, the requesting student is allocated to the three submissions of students who request just before him- or herself.

In Chapter 4, we pointed out that the allocation method called circular allocation has an adverse effect on peer assessment accuracy. Circular allocation is a method that assigns some order relation to students and allocates next k students' submissions to students, where k is the reviewing number. The allocation in Figure 5.1 is equal to the circular allocation except for students with IDs 1-3. Therefore, it is considered that the allocation in Figure 5.1 also has an adverse effect on peer assessment accuracy.

We propose a method that considers the trade-off of peer assessment accuracy and RR imbalance. The proposed method is a simple method that replaces the RRB algorithm allocation with a random allocation at specified intervals. In this study, we conducted experiments to confirm the usefulness of the proposed method using artificial reviewing order data and real reviewing order data. We performs the evaluation of the proposed method while changing the frequency of random allocation. As a result, it is confirmed that the peer assessment accuracy can be improved without impairing the RR imbalance by making a small substitution to the random allocation.

The above experimental results suggest that the estimation accuracy can be improved by replacing the allocation that adversely affect score estimation with random allocation. Therefore, we confirm the results of applying the same methodology to group allocation explained in Chapter 4, and discuss the usefulness of

partially replacing with random allocation.

The structure of this study is described below. Section 5.2 explains the proposed method of our research. In Section 5.3, we present the experimental results. In Section 5.4, we discuss the usefulness of the methodology used in the proposed method. Finally, we conclude this work and suggest future work in Section 5.5.

5.2 Proposed Method

The proposed method is as follows:

Algorithm 4 RRB algorithm with partially random allocation

INPUT: $V = \{v_1, \ldots, v_n\}$ ▷ a set of n students
INPUT: $S = \langle v_{i_1}, \ldots, v_{i_m} \rangle$ ▷ reviewing order
INPUT: h ▷ an interval of random allocation
OUTPUT: E ▷ student-submission allocation

```
E ← {}
for l ← 1 to m do
    if l mod h == 0 then
        v_j is selected from V \ {v_{i_l}} at random
    else
        v_j is selected according to RRB algorithm
    end if
    E ← E + (v_{i_l}, v_j)
end for
```

Here, V is a student set, S is a student's reviewing order, and h is an interval of random allocation. At every h request, the allocation in the RRB algorithm is replaced with random allocation. This method has been inspired by the experimental results in Chapter 4 where random allocation showed excellent results.

5.3 Experiment

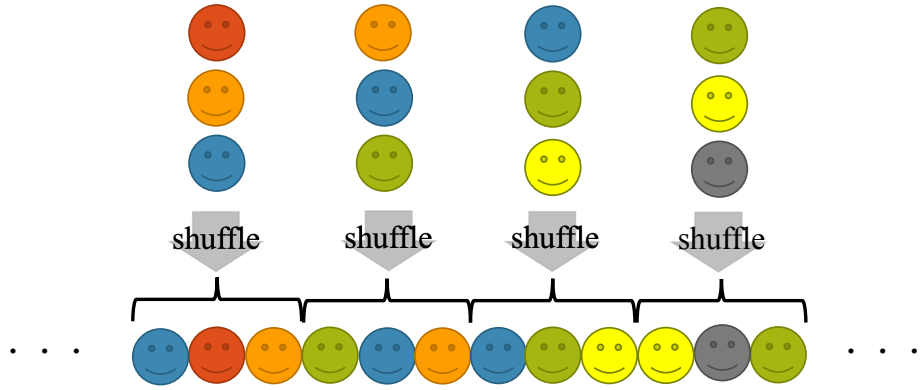
In this experiment, we utilize the reviewing order based on the real data and the artificial reviewing order, and create the studetnt-submission allocation using the proposed method while changing the interval of random allocation h. Then,

(a) Order that students review continuously

(b) Order that students review alternately

Figure 5.2. A part of real reviewing order.

(a) Order that students review continuously



(b) Order that students review alternately

Figure 5.3. A part of artificial reviewing order.

according to the settings in Chapter 3 and in Chapter 4, the created allocations are compared. This section first describes the details of the real reviewing order and the artificial reviewing order, and then describes the experimental results.

5.3.1 Dataset

Real reviewing order

In this research, we utilize real data 1 and real data 2 in Section 3.5.1. We construct the reviewing order using real data based on the time when the comments were created in the same way in Section 3.5.1. Figure 5.2 shows the examples of extracting part of the real reviewing order.

Each node in Figure 5.2 represents a student who reviews a submission, and nodes of the same color represent the same student. Real reviewing orders shown in Figure 5.2 are both examples in which students tend to review submissions collectively. Most of the requests are continuous in Figure 5.2(a), but the requests are alternated in Figure 5.2(b). Such order in Figure 5.2(b) can occur when the number of students who make requests at a specific time is large. This situation occurs frequently in environments with a large number of participants, such as MOOCs. In this study, the following two artificial orders are created to consider two extreme cases.

Artificial reviewing order

The first artificial reviewing order is a sequence in which each student requests k times consecutively, assuming that the reviewing number is k (Figure 5.3(a)).

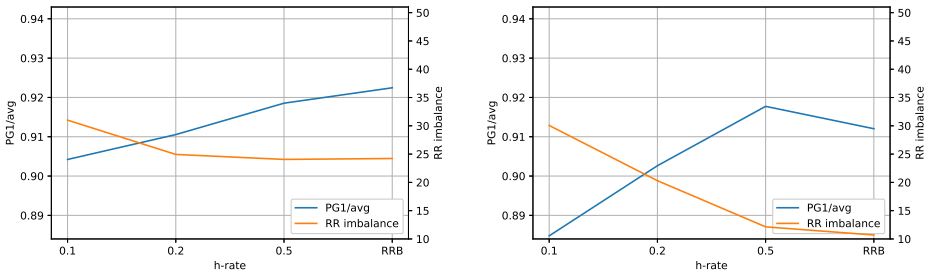
Another artificial reviewing order is consisted by the following algorithm.

Algorithm 5 Create artificial order that students review alternately

INPUT: $V = \{v_1, \ldots, v_n\}$ ▷ a set of n students
INPUT: k ▷ reviewing number
OUTPUT: S ▷ reviewing order

1: $S \leftarrow \langle\rangle$
2: **for** $i \leftarrow 1$ to $k-1$ **do**
3: $\quad S.append(shuffle(\langle v_1, \ldots, v_i \rangle))$
4: **end for**
5: **for** $i \leftarrow 1$ to $n-k+1$ **do**
6: $\quad S.append(shuffle(\langle v_i, \ldots, v_{i+k-1} \rangle))$
7: **end for**
8: **for** $i \leftarrow n-k+2$ to n **do**
9: $\quad S.append(shuffle(\langle v_i, \ldots, v_n \rangle))$
10: **end for**

Figure 5.3(b) shows the example of the reviewing order and how the algorithm works. With this algorithm, except for the first and last $k(k-1)/2$ students, student set $\{v_i, \ldots, v_{i+k-1}\}(i = 1, \ldots, n-k+1)$ are sorted randomly and review submissions in order. This algorithm generates a sequence as shown in Figure 5.3(b), which can be regarded as an extreme case of the order that students review alternately.



(a) order that students review continu-ously

(b) order that students review alternately

Figure 5.4. Experimental results with artificial reviewing order ($n = 50, k = 5$).

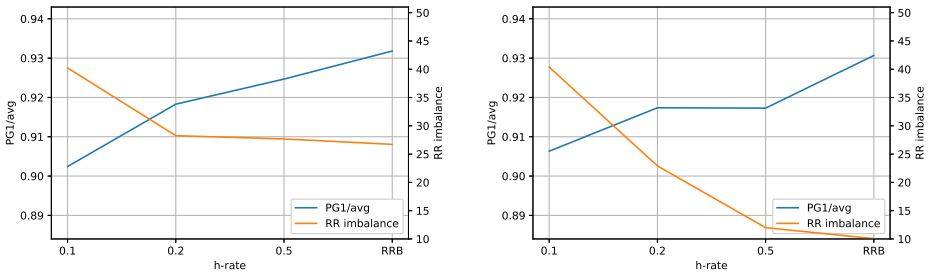
5.3.2 Experimental Results

We give experimental results on the artificial reviewing order, then we show the results on the real reviewing order. For each reviewing order, we demonstrate the RR-imbalance and estimation accuracy when using proposed adaptive allocation method while changing an interval of random allocation h. We show the estimation accuracy based on setting in Section 4.3.1. Note that we generate 100 subsets of data in this experiment instead of 500 subsets as in Chapter 4.

The results are shown in Figure 5.4, 5.5, 5.6 and 5.7. Each horizontal axis indicates a parameter h-rate that adjusts the size of an interval of random allocation h. Note that an interval of random allocation $h = \lceil n \cdot h\text{-rate} \rceil$, when n is the number of students. For example, if $n = 36$ and h-rate $= 0.2$, then $h = 7$. When h is larger than the length of the reviewing order, our proposed algorithm is equal to the RRB algorithm. Therefore, we represent the results where the RRB algorithm is applied in the rightmost in each figure. The vertical axis indicates the PG1/avg and RR imbalance, with the blue line representing PG1/avg and the orange line representing RR imbalance. Small values are preferable for both PG1/avg and RR imbalance.

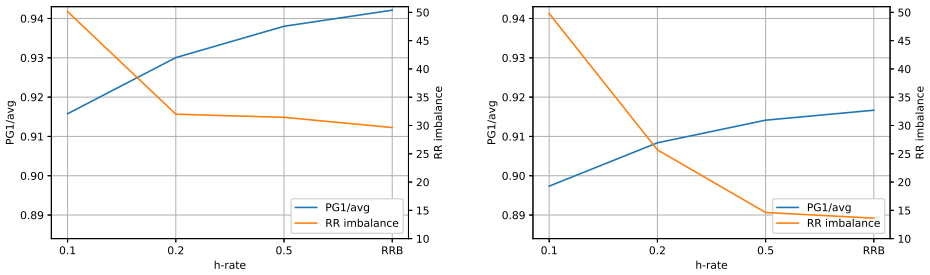
Artificial reviewing order

We describe the results when using artificial data. Figure 5.4, Figure 5.5, and Figure 5.5 show the results when the number of students $n = 50$, $n = 100$ and $n = 1000$, and all the reviewing number $k = 5$. In addition, each subfigure



(a) order that students review continu-(b) order that students review alternately ously

Figure 5.5. Experimental results with artificial reviewing order ($n = 100, k = 5$).

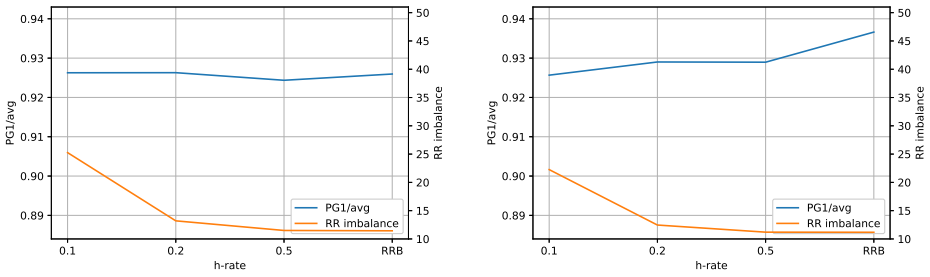


(a) order that students review continu-(b) order that students review alternately ously

Figure 5.6. Experimental results with artificial reviewing order ($n = 1000, k = 5$).

(a) shows the results with the order that students review continuously, and each subfigure (b) shows the results with the order that students review alternately.

In most cases, the RR imbalance decreases and PG1/avg increases as h-rate increases, that is, our proposed algorithm approaches the RRB algorithm. In addition, the peer assessment accuracy (PG1/avg) is improved without imparing the RR imbalance by making a small substitution to the random allocation in some cases. For example, in Figure 5.4, RR-imbalance when h-rate is 0.5 is almost the same as that when using RRB, but PG1/avg when h-rate is 0.5 is lower than that when using RRB.



(a) real data 1 (b) real data 2

Figure 5.7. Experimental results with real reviewing order.

In each subfigure (a), except when h-rate is 0.1, the values of RR imbalance are similar. However, in each subfigure (b), the RR imbalance when h-rate is 0.2 is clearly larger than that when h-rate is 0.5. Therefore, when we focus on RR imbalance, the proposed method is considered to be more effective for the order that students review continuously than for the order that students review alternately. On the other hand, when we focus on the values of PG1/avg, it is difficult to find the obvious difference between the two experimental results in each figure, but both results show the same tendency that the PG1/avg increases as h-rate increases.

Real reviewing order

In addition, we describe the experimental results when using real data (see Figure 5.7). Real data 2 shows a similar tendency to the result when using artificial data, but real data 1 shows almost no change in estimation accuracy even if h-rate changes. This suggests that the student-submission allocation on real data 1 created by the RRB algorithm contains enough randomness of allocation. This is probably because real data 1 has a large number of students, and the reviewing requests at a specific time are more crowded than the case where using Algorithm 5.

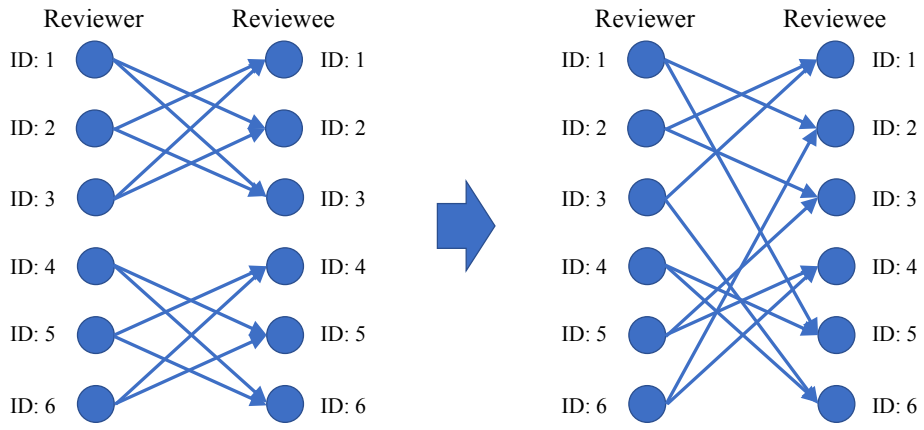


Figure 5.8. Group allocation partially replaced with random allocation.

5.4 Discussion about Partially Replacing with Random Allocation

The above experimental results suggest that replacing a part of student-submission allocation with random allocation can improve the estimation accuracy. In this section, to confirm this hypothesis, we perform a simple experiment for group allocation described in Chapter 4.

Group allocation is an allocation that divides a student set into groups and generates allocation in each group. There is an assertion that learning on a group has some benefits [69], so group allocation in peer assessment may be adopted in practical setting. It has already been pointed out in Chapter 4 that the estimation accuracy deteriorates when using group allocation. Here, we consider allocation that replaces a part of group allocation with random allocation (see Figure 5.8)

We obtain estimation accuracy (PG1/avg) using the settings in Chapter 4, where r out of k allocations for each student are replaced with random allocation in group allocation. When the number of students $n = 100$, the number of groups $d = 10$, $k = 9$, and $r = 0, 1, 2, 3, 4$, the corresponding values of PG1/avg are $0.997, 0.903, 0.880, 0.864, 0.854$. It can be seen that the accuracy is improved by inserting random allocations into group allocation. Therefore, it is suggested that the methodology of partially replacing with random allocation is useful for improving accuracy.

5.5 Conclusion

We propose a method that considers the trade-off between fairness based on RR imbalance and peer assessment accuracy. The proposed method is to replace a part of the allocation in the RRB algorithm with random allocation. This study asserts the usefulness of the proposed method through simulation. In addition, we discuss the usefulness of the methodology of replacing part of the allocation that worsens the estimation accuracy with random allocation. In future work, we plan to apply the above methodology to other student-submission allocation and prove the usefulness of the methodology theoretically.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

We developed and analyzed student-submission allocation methods for peer assessment, which has been rarely focused on, and improved the problems of peer assessment as follows:

- Imbalance of the number of reviews due to dropouts
- Low reliability of of students' reviewing results

In the first study, in order to solve the first problem, we developed a new adaptive allocation method which achieves that the student's submission is reviewed as many times as the student reviews other submissions. Additionally, we extend the proposed method to the method which can consider the students' reviewing ability. We theoretically analyzed the degree of imbalance when using our proposed method, and compared the imbalance between proposed allocation methods and existing allocation methods through simulation.

In the second study, we analyzed what kind of student-submission allocation is effective for the existing score estimation method from multiple students' scores, which is developed to solve the low reliability of student reviewing. This analysis indicates that some allocation methods which are considered to be used in actual

peer assessment have bad effect on estimation accuracy, and random allocation is superior.

The above two studies recommend different student-submission allocation methods for two different objectives. In third study, we pointed out that, when using the allocation method proposed in the first study, the estimation accuracy decreases under certain circumstances. Then, we proposed an allocation method that considers the trade-off between two objectives. In addition, we discuss the usefulness of the methodology of replacing a part of student-submission allocation with random allocation.

6.2 Future Work

The major issue in the future is as follows. In our research, simulation was often used to confirm the usefulness of each student-submission allocation, so we would like to confirm the usefulness of each allocation under practical conditions.

In the first study, we plan to study how to estimate the reviewing ability values from the students' past behavioral data and combine our proposed allocation algorithm with the estimation algorithm.

In the second study, we plan to further consider the problem from a theoretical perspective. In addition, it should be noted that student reviewing is not always performed in accordance with the predefined allocations due to dropout; therefore, we also plan to consider the effect of allocation with dropout on scoring accuracy.

In the third study, we plan to apply the proposed methodology to other student-submission allocation and prove the usefulness of the methodology theoretically.

Acknowledgements

I would like to thank my supervisors, Professor Masatoshi Yoshikawa and Assistant Professor Toshiyuki Shimizu, for their generous supports and constructive feedback. In addition, I would like to thank Professor Yasuhito Asano, for working together and giving me insightful comment. In addition, I would like to thank my other two academic advisors, Professor Keishi Tajima and Hisashi Kashima for their valuable suggestions and comments on my research. I would like to thank all our Lab faculties for their help in every possible way. During the Lab seminars, Associate Professor Kazunari Sugiyama and Adam Jatowt made valuable comments on my research. Also, the Lab secretaries, Ms. Yoko Nakahara and Ms. Rika Ikebe helped me much on daily business. I would like to thank all the Lab mates.

Hideaki Ohashi, March 2020

References

[1] Keith Topping. Peer assessment between students in colleges and universities. *Review of educational Research*, 68(3):249–276, 1998.

[2] Nancy Falchikov and Judy Goldfinch. Student peer assessment in higher education: A meta-analysis comparing peer and teacher marks. *Review of educational research*, 70(3):287–322, 2000.

[3] Keith J Topping. Peer assessment. *Theory into practice*, 48(1):20–27, 2009.

[4] Laura Pappano. The year of the mooc. *The New York Times*, 2(12):2012, 2012.

[5] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. In *Educational Data Mining*, 2013.

[6] Takeru Sunahase, Yukino Baba, and Hisashi Kashima. Probabilistic modeling of peer correction and peer assessment. In *Educational Data Mining*, 2019.

[7] Hoi K Suen. Peer assessment for massive open online courses (moocs). *The International Review of Research in Open and Distributed Learning*, 15(3), 2014.

[8] Dmytro Babik, Edward F Gehringer, Jennifer Kidd, Ferry Pramudianto, and David Tinapple. Probing the landscape: Toward a systematic taxonomy of online peer assessment systems in education. In *Educational Data Mining (Workshops)*, 2016.

[9] edX. https://www.edx.org.

[10] Coursera. https://coursera.org.

[11] Edward F Gehringer. A survey of methods for improving review quality. In *International Conference on Web-Based Learning*, pages 92–97, 2014.

[12] Enrique Sánchez Acosta, Juan José Escribano Otero, and Gabriela Christie Toletti. Peer review experiences for mooc. development and testing of a peer review system for a massive online course. *The New Educational Review*, 37(3):66–79, 2014.

[13] Daniel FO Onah, Jane Sinclair, and Russell Boyatt. Dropout rates of massive open online courses: behavioural patterns. *International Conference on Education and New Learning Technologies*, pages 5825–5834, 2014.

[14] Sarah EM Meek, Louise Blakemore, and Leah Marks. Is peer review an appropriate form of assessment in a mooc? student participation and performance in formative peer review. *Assessment & Evaluation in Higher Education*, 42(6):1000–1013, 2017.

[15] Karthik Raman and Thorsten Joachims. Methods for ordinal peer grading. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1037–1046, 2014.

[16] UDACITY. https://www.udacity.com.

[17] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. Studying learning in the worldwide classroom research into edx's first mooc. *Research & Practice in Assessment*, 8:13–25, 2013.

[18] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 687–698, 2014.

[19] Jihyun Park, Renzhe Yu, Fernando Rodriguez, Rachel Baker, Padhraic Smyth, and Mark Warschauer. Understanding student procrastination via mixture models. In *Educational Data Mining*, 2018.

[20] Justin Reich. Rebooting mooc research. *Science*, 347(6217):34–35, 2015.

[21] Justin Reich and José A Ruipérez-Valiente. The mooc pivot. *Science*, 363(6423):130–131, 2019.

[22] Kwangsu Cho, Christian D Schunn, and Roy W Wilson. Validity and reliability of scaffolded peer assessment of writing from instructor and student perspectives. *Journal of Educational Psychology*, 98(4):891, 2006.

[23] Katy Jordan. Massive open online course completion rates revisited: Assessment, length and attrition. *The International Review of Research in Open and Distributed Learning*, 16(3), 2015.

[24] Thomas Staubitz, Dominic Petrick, Matthias Bauer, Jan Renz, and Christoph Meinel. Improving the peer assessment experience on mooc platforms. In *Proceedings of the third ACM conference on Learning@ Scale*, pages 389–398, 2016.

[25] Sunny SJ Lin, Eric Zhi-Feng Liu, and Shyan-Ming Yuan. Web-based peer assessment: feedback for students with various thinking-styles. *Journal of computer assisted Learning*, 17(4):420–432, 2001.

[26] Sarah Gielen, Elien Peeters, Filip Dochy, Patrick Onghena, and Katrien Struyven. Improving the effectiveness of peer feedback for learning. *Learning and instruction*, 20(4):304–315, 2010.

[27] Philip M Sadler and Eddie Good. The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31, 2006.

[28] Maria De Marsico, Luca Moschella, Andrea Sterbini, and Marco Temperini. Effects of network topology on the openanswer ' s bayesian model of peer assessment. In *European Conference on Technology Enhanced Learning*, pages 385–390, 2017.

[29] Andrea Sterbini and Marco Temperini. Openanswer, a framework to support teacher's management of open answers through peer assessment. In *IEEE Frontiers in Education Conference (FIE)*, pages 164–170, 2013.

[30] Michael Mogessie Ashenafi, Marco Ronchetti, and Giuseppe Riccardi. Predicting student progress from peer-assessment data. In *Educational Data Mining*, 2016.

[31] Erkan Er, Bote Lorenzo, L Miguel, Eduardo Gómez Sánchez, Yannis A Dimitriadis, Asensio Pérez, and Juan Ignacio. Predicting student participation in peer reviews in moocs. In *Proceedings of EMOOCs 2017*, 2017.

[32] Luca De Alfaro and Michael Shavlovsky. Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 415–420, 2014.

[33] James R Wright, Chris Thornton, and Kevin Leyton-Brown. Mechanical ta: Partially automated high-stakes peer grading. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 96–101, 2015.

[34] Chinmay E Kulkarni, Michael S Bernstein, and Scott R Klemmer. Peerstudio: rapid peer feedback emphasizes revision and improves performance. In *Proceedings of the second ACM conference on Learning@ Scale*, pages 75–84, 2015.

[35] Kwangsu Cho and Christian D Schunn. Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education*, 48(3):409–426, 2007.

[36] Stephen P Balfour. Assessing writing in moocs: Automated essay scoring and calibrated peer review. *Research & Practice in Assessment*, 8:40–48, 2013.

[37] Dave Clarke, Tony Clear, Kathi Fisler, Matthias Hauswirth, Shriram Krishnamurthi, Joe Gibbs Politz, Ville Tirronen, and Tobias Wrigstad. In-flow peer review. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, pages 59–79, 2014.

[38] John Hamer, Kenneth TK Ma, and Hugh HF Kwong. A method of automatic grade calibration in peer assessment. In *Proceedings of the 7th Australasian conference on Computing education-Volume 42*, pages 67–72, 2005.

[39] Fei Mi and Dit-Yan Yeung. Probabilistic graphical models for boosting cardinal and ordinal peer grading in moocs. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[40] Nihar B Shah, Joseph K Bradley, Abhay Parekh, Martin Wainwright, and Kannan Ramchandran. A case for ordinal peer-evaluation in moocs. In *NIPS Workshop on Data Driven Education*, pages 1–8, 2013.

[41] Karthik Raman and Thorsten Joachims. Bayesian ordinal peer grading. In *Proceedings of the second ACM conference on Learning@ Scale*, pages 149–156, 2015.

[42] Tianqi Wang, Qi Li, and Jing Gao. Improving peer assessment accuracy by incorporating relative peer grades. In *Educational Data Mining*, 2019.

[43] Jorge Díez Peláez, Óscar Luaces Rodríguez, Amparo Alonso Betanzos, Alicia Troncoso, and Antonio Bahamonde Rionda. Peer assessment in moocs using preference learning via matrix factorization. In *NIPS Workshop on Data Driven Education*, 2013.

[44] Toby Walsh. The peerrank method for peer assessment. *Frontiers in Artificial Intelligence and Applications*, 263, 2014.

[45] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

[46] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

[47] Qiang Liu, Jian Peng, and Alexander T Ihler. Variational inference for crowdsourcing. In *Advances in neural information processing systems*, pages 692–700, 2012.

[48] Mehdi SM Sajjadi, Morteza Alamgir, and Ulrike von Luxburg. Peer grading in a course on algorithms and data structures: Machine learning algorithms do not improve over simple baselines. In *Proceedings of the third ACM conference on Learning@ Scale*, pages 369–378, 2016.

[49] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

[50] Hou Pong Chan and Irwin King. Leveraging social connections to improve peer assessment in moocs. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 341–349, 2017.

[51] Yong Han, Wenjun Wu, Suozhao Ji, Lijun Zhang, and Hui Zhang. A human-machine hybrid peer grading framework for spocs. In *Educational Data Mining*, 2019.

[52] Iria Estévez-Ayres, Raquel M Crespo García, Jesús A Fisteus, and Carlos Delgado Kloos. An algorithm for peer review matching in massive courses for minimising students' frustration. *J. UCS*, 19(15):2173–2197, 2013.

[53] Yong Han, Wenjun Wu, and Yanjun Pu. Task assignment of peer grading in moocs. In *International Conference on Database Systems for Advanced Applications*, pages 352–363, 2017.

[54] Hou Pong Chan, Tong Zhao, and Irwin King. Trust-aware peer assessment using multi-armed bandit algorithms. In *Proceedings of the 25th International Conference on World Wide Web*, pages 899–903, 2016.

[55] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[56] Edward F Gehringer and Yun Cui. An effective strategy for the dynamic mapping of peer reviewers. In *ASEE Annual Conference and Exposition*, 2002.

[57] Raquel M Crespo, Abelardo Pardo, and C Delgado Kloos. An adaptive strategy for peer review. In *IEEE Frontiers in Education Conference (FIE)*, pages F3F–7, 2004.

[58] Raquel M Crespo, Abelardo Pardo, Juan Pedro Somolinos Pérez, and Carlos Delgado Kloos. An algorithm for peer review matching using student profiles based on fuzzy classification and genetic algorithms. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 685–694, 2005.

[59] Raquel M Crespo Garcia and Abelardo Pardo. A supporting system for adaptive peer review based on learners' profiles. In *Proceedings of Computer Supported Peer Review in Education Workshop*, pages 22–31, 2010.

[60] Daniel S Weld, Eytan Adar, Lydia Chilton, Raphael Hoffmann, Eric Horvitz, Mitchell Koch, James Landay, Christopher H Lin, and Mausam Mausam. Personalized online education—a crowdsourcing challenge. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[61] Ittai Abraham, Omar Alonso, Vasilis Kandylas, and Aleksandrs Slivkins. Adaptive crowdsourcing algorithms for the bandit survey problem. In *Conference on learning theory*, pages 882–910, 2013.

[62] Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *International Conference on Machine Learning*, pages 64–72, 2013.

[63] Yan Yan, Romer Rosales, Glenn Fung, and Jennifer G Dy. Active learning from crowds. In *International Conference on Machine Learning*, volume 11, pages 1161–1168, 2011.

[64] Lakshmi Ramachandran et al. *Automated Assessment of Reviews.* PhD thesis, North Carolina State University, Raleigh, NC.

[65] Gabriel Zingle, Balaji Radhakrishnan, Yunkai Xiao, Edward Gehringer, Zhongcan Xiao, Ferry Pramudianto, Gauraang Khurana, and Ayush Arnav. Detecting suggestions in peer assessments. In *Educational Data Mining*, 2019.

[66] Juan Ramón Rico-Juan, Antonio-Javier Gallego, and Jorge Calvo-Zaragoza. Automatic detection of inconsistencies between numerical scores and textual feedback in peer-assessment processes with machine learning. *Computers & Education*, page 103609, 2019.

[67] Haddadi Lynda, Bouarab-Dahmani Farida, Berkane Tassadit, and Lazib Samia. Peer assessment in moocs based on learners' profiles clustering. In *8th International Conference on Information Technology (ICIT)*, pages 532–536, 2017.

[68] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263, 2008.

[69] Leonard Springer, Mary Elizabeth Stanne, and Samuel S Donovan. Effects of small-group learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis. *Review of educational research*, 69(1):21–51, 1999.