

## Adaptive Balanced Allocation for Peer Assessments\*

Hideaki OHASHI<sup>†a)</sup>, Nonmember, Yasuhito ASANO<sup>††</sup>, Toshiyuki SHIMIZU<sup>†</sup>,  
and Masatoshi YOSHIKAWA<sup>†</sup>, Members

**SUMMARY** Peer assessments, in which people review the works of peers and have their own works reviewed by peers, are useful for assessing homework. In conventional peer assessment systems, works are usually allocated to people before the assessment begins; therefore, if people drop out (abandoning reviews) during an assessment period, an imbalance occurs between the number of works a person reviews and that of peers who have reviewed the work. When the total imbalance increases, some people who diligently complete reviews may suffer from a lack of reviews and be discouraged to participate in future peer assessments. Therefore, in this study, we adopt a new adaptive allocation approach in which people are allocated review works only when requested and propose an algorithm for allocating works to people, which reduces the total imbalance. To show the effectiveness of the proposed algorithm, we provide an upper bound of the total imbalance that the proposed algorithm yields. In addition, we extend the above algorithm to consider reviewing ability. The extended algorithm avoids the problem that only unskilled (or skilled) reviewers are allocated to a given work. We show the effectiveness of the proposed two algorithms compared to the existing algorithms through experiments using simulation data.

**key words:** peer assessment, task allocation, allocation algorithm

## 1. Introduction

Peer assessments, in which people review the works of peers and have their own works reviewed by peers, are useful for reviewing homework. Especially, peer assessments are effective when assessing essay-type homework which is difficult to review automatically, and when the number of participants is large, such as in a massive open online course (MOOC), in which people can attend various lectures on the Internet. Lecturers and teaching assistants (TAs) alone are unable to review large volumes of works [1]–[3].

However, some reports indicate that people are not willing to participate in peer assessments; one reason is that people are disheartened by the lack of reviews [4], [5]. Therefore, we need to develop methods of peer assessment that allow people to receive sufficient feedback based on the number of reviews to increase the number of people who

participate in peer assessments.

A major reason for the existence of insufficient review numbers is that peers dropout without reviewing allocated works [4], [6]. In existing peer assessment systems, each person is usually asked to review a predefined number of works, and works are allocated to people before the peer assessments start. If a certain number of people drop out of the review process, an imbalance occurs between the number of works a person reviews (termed the “reviewing number”) and the number of peers who review the work of the same person (termed the “reviewed number”). When the total imbalance increases, people who diligently finish reviews may suffer from a lack of reviews and be discouraged to participate in future peer assessments. Note that we focus on the assignment tasks that is not so difficult and time-consuming to review for the people who complete the task. We assume dropout occurs because people don’t have motivation. To resolve dropout due to the task difficulty is out of our scope.

To address the above problem, we develop a new adaptive allocation approach in which people are allocated works only when requested. People can request one work to review at any time; they can request second and subsequent works to review only after they have finished the review of the previously requested work. This rule is more suitable for a realistic situation in which some people 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 person, 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 work of one person will be reviewed as many times as that same person reviews the works of others. It can be expected that this algorithm resolves dissatisfaction about the lack of reviews and incentivizes people to review the works 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 people; instead, it depends on the maximum reviewing number among people. In practical situations, the maximum reviewing number usually does not increase, even if the number of people grows. Therefore, our results show that the average difference between the re-

Manuscript received June 26, 2019.

Manuscript revised November 4, 2019.

Manuscript publicized December 26, 2019.

<sup>†</sup>The authors are with Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.

<sup>††</sup>The author is with Faculty of Information Networking for Innovation and Design, Toyo University, Tokyo, 115–0053 Japan.

\*A previous version of this paper was accepted in the The 25th International Computing and Combinatorics Conference (COCOON 2019), July, 2019.

a) E-mail: ohashi@db.soc.i.kyoto-u.ac.jp  
DOI: 10.1587/transinf.2019DAP0004

viewing number of each person and the reviewed number decreases as the number of people increases. This property is desirable in MOOC settings from the viewpoint of fairness among people.

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 people 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 the lecturers and TAs need to perform.

In addition, we also consider people's reviewing ability in addition to the RR imbalance. We assume that a scalar reviewing ability value for each person are given in advance, similar to the existing research [7]. When the average reviewing ability value of the reviewers allocated to a work varies, it means that the least (or most) skillful reviewers concentrate on only one work. Hence, we want to make this average value be balanced among works. 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 simulation data.

The remainder of this paper is organized as follows. In Sect. 2, we introduce the related works. We describe the problem definitions in this research in Sect. 3. In Sect. 4, we describe the RRB algorithm and ARRB algorithm. In Sect. 5, we prove the upper bound of the RR imbalance by the RRB algorithm. We present the experimental results in Sect. 6, and finally, we conclude this work and suggest future work in Sect. 7.

## 2. Related Work

### 2.1 Allocation Methods in Peer Assessment

Crowdsourcing has attracted much attention, and studies on crowdsourcing and peer assessment are closely related [8]. Many task allocation methods have been proposed for crowdsourcing [9]–[12]; 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 provided; 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.

Est'vez-Ayres et al. [13] proposed an allocation mechanism to avoid lack of reviews due to dropout and con-

firmed its usefulness through a simulation. They assumed that some people were willing to review other works even when their reviewing number exceeded their reviewed number. We do not assume such optimistic person characteristics in this study.

Han et al. [7] 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 person's reviewing ability value is given and, like our work, aimed to find allocations to achieve fair reviews. However, they did not consider dropout.

### 2.2 Methods for Improving the Quality of Reviews in Peer Assessments

A method of automatically assessing review content (automated meta-reviewing) that prompts the reviewer to correct and improve review content has been proposed [14]. In addition, another method was proposed in which the reviewer scores the reviewer on his or her review content [15]. Increasing the quality of the rubrics (reviewing standards) used in peer assessments leads directly to improved review quality; therefore, some studies have verified the effect of rubrics [2], [16]. In addition, many studies exist that aggregate reviewer scores in peer assessments; these studies apply quality control research in the context of crowdsourcing [3], [17]–[20]. The above studies are orthogonal to our study; hence, we can combine their methodologies and results with ours.

## 3. Problem Setting

Initially, we explain our problem setting intuitively through Fig. 1. In this research, to deal with realistic situations in which some people drop out during the peer assessment process, we propose an allocation algorithm that uses an adaptive allocation approach. Under this approach, a new work is allocated to a person only when he or she requests one, and he or she can request an additional work to review only after he or she has finished the review of the previous work. In addition, we assume that people always complete the requested review. This assumption is considered to be valid because people who are not willing to review do not request a work in the first place.

In Fig. 1, we assume that there are five people,  $a, b, c, d$ , and  $e$ , and each vertex represents a person. First,  $a$  requests a work; then, the work 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 work and that each person can review a given work only once. After the first allocation, the next allocation occurs when another person requests a work, and then, a directed edge is drawn. These steps are repeated under an adaptive allocation approach.

Let  $V$  be a set of people,  $E_i$  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)

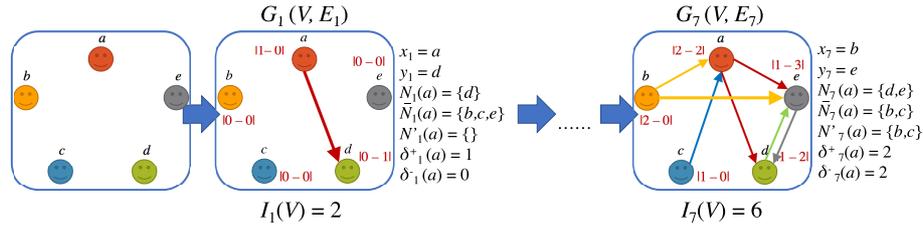


Fig. 1 Example of adaptive allocation behavior.

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 person and allocating works to all people 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 person doing the  $i$ -th request under the adaptive allocation approach be  $x_i \in V$ . A work by a person  $y_i (\neq x_i) \in V$  is allocated to  $x_i$  before a person  $x_{i+1}$  can request a work. This allocation is represented by a directed edge from  $x_i$  to  $y_i$ . In the graph  $G_i$ , let the set of people whose works are allocated to person  $v \in V$  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 people who review the work of person  $v \in V$ . The reviewing number (outdegree) of person  $v$  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 Fig. 1. In Fig. 1, we assume five people,  $a, b, c, d$ , and  $e$ ; thus,  $V = \{a, b, c, d, e\}$ . Initially, person  $a$  requests a work, and the work of person  $d$  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 person  $v \in V$  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 [7]. 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 people. 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 people who review  $v \in V$ 's work as  $W_t(v)$  and the average value of  $W_t(v)$  of all people 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 people. The ARR imbalance  $I_t'(V)$  when the  $t$ -th allocation is finished is given by the following equation.

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

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

#### 4. 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 Sect. 5, and experiments to evaluate the performance of the RRB and ARRB algorithms are presented in Sect. 6.

##### 4.1 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.

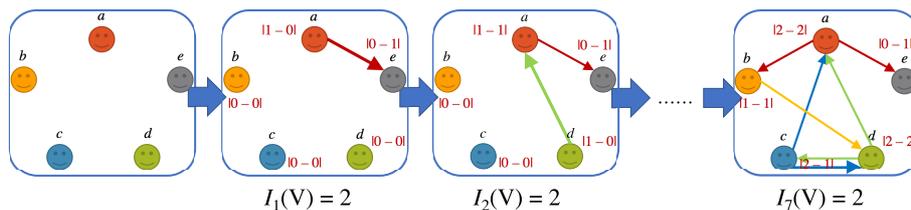


Fig. 2 Example of RRB behavior.

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

We provide an intuitive explanation of the above algorithm using Fig. 2. In this figure, it is assumed that there are five people,  $a, b, c, d$ , 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 person is 0; therefore, the work is randomly allocated to  $a$ . Let us assume that the work of  $e$  is randomly selected. Next, because the difference between the reviewing number and reviewed number of  $a$  is the maximum, the work 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 work of  $d$  is preferentially allocated in the next step. In Fig. 2, the above allocation is repeated, showing intuitively how the RRB algorithm aims to reduce the RR imbalance.

#### 4.2 ARRB (Ability-Aware Reviewing-Reviewed Balanced Allocation Algorithm)

Next, we extend the RRB to an algorithm that reduces the ARR imbalance. Here,  $W'_i(v)$  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 people. 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.

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

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

### 5. 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 people 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 people  $n$ ; it depends only on the maximum number of reviews performed by any one reviewer. When an enormous number of people exist, such as in an MOOC,  $k$  is expected to be considerably smaller than  $n$  because one person cannot review works by everyone. In other words, the proposed algorithm should be extremely effective on MOOCs. Although we assume that the number of people is larger than  $k^2 + k + 1$ , this is equivalent to the assumption that the total number of people is larger than the square of the reviewing number of any one person. It is natural to use this assumption when many people 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,  $I_i(V') = (|E'| + |E_I|) - (|E'| + |E_O|) = |E_I| - |E_O|$   $\square$

**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.

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

**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:

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

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:

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

From the above two inequalities (1 and 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.  $\square$

**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:

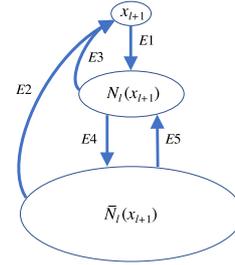
$$I_i(V) \leq 4k_i^2 - 4k_i + 2$$

**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 person 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:



**Fig. 3** Grouping for proof of Theorem 1.

$$\delta_l^+(x_{l+1}) - \delta_l^-(x_{l+1}) \geq 0 \quad \& \quad \delta_l^+(y_{l+1}) - \delta_l^-(y_{l+1}) = 0 \quad (3)$$

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:

$$\delta_l^+(x_{l+1}) \leq k_l - 1 \quad (4)$$

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 Fig. 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.

- 1  $I_l(\{x_{l+1}\})$ : We consider the edge sets  $E1, E2$ , and  $E3$  in Fig. 3. From conditions (3) and (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 Fig. 3. From condition (3), the RRB algorithm selects a  $y_{l+1}$  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:

$$\forall v \in \bar{N}_l(x_{l+1}), \delta_l^+(v) - \delta_l^-(v) \leq 0 \quad (5)$$

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 (4),  $|N_l(x_{l+1})| \leq k_l - 1$  holds. Then, because the maximum outdegree is  $k_l$ , the following is satisfied:

$$I_l(\bar{N}_l(x_{l+1})) \leq |E4| \leq k_l(k_l - 1) \quad (6)$$

- 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 Fig. 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 (4), because  $|N_l(x_{l+1})| \leq k_l - 1$ , the out-degree 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  $\bar{N}_l(x_{l+1})$ .

- Edges from  $\{x_{l+1}\}$  ( $E1$ ): From condition (4), the number of edges is less than  $k_l - 1$ .
- Edges between  $N_l(x_{l+1})$ : From condition (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.
- Edges from  $\bar{N}_l(x_{l+1})$  ( $E5$ ): From condition (5) and Lemma 1, the following is satisfied:

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

Therefore, from condition (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 (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.  $\square$

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

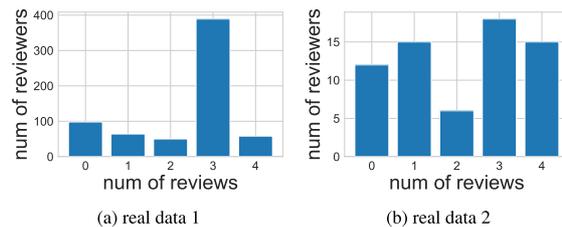
## 6. Experiments

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

### 6.1 Simulation Data Based on Canvas Network Dataset

We use the simulation data based on the data published by Canvas Network<sup>†</sup>. This data is comprised of de-identified

<sup>†</sup><https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/XB2TLU>



**Fig. 4** The number of reviewers for each number of reviews from the real data.

data from March 2014 - September 2015 of Canvas Network open courses.

In our experiments, we utilize those data whose class ID is 770000832960949 and whose assignment ID is 770000832930436 (denoted as CN data 1) and those data whose class ID is 770000832945340 and assignment ID is 770000832960431 (denoted as CN 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*.

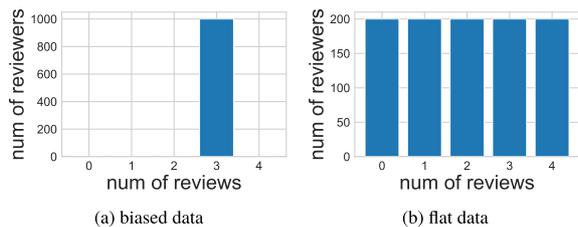
In CN data 1, one reviewer performed 25 reviews just before the end of peer assessment. We consider this value as representing a lecturer or TA who reviewed the student submissions whose reviewed number is insufficient. Thus, we replaced the reviewing number of this reviewer with 3, the mode value of the reviewing number from the CN 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.

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 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.

We call the CN data 1 complemented as mentioned above as real data 1, and the complemented CN data 2 as real data 2. Figure 4 shows a plot of the number of reviewers for each number of reviews from the datasets real data 1 and real data 2.

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.

In addition, it is not possible to read the strict reviewing



**Fig. 5** The number of reviewers for each number of reviews from the synthetic data.

order from the Canvas Network data. Therefore, we construct the reviewing order using the following two methods:

### 6.1.1 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.

### 6.1.2 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  $x_i$ .

## 6.2 Simulation Data Based on Synthetic Dataset

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 Fig. 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 generated based on the reviewer transition model described in Sect. 6.1.2.

## 6.3 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.

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. Note that we assume that students can request additional works after they complete reviewing the allocated works like the existing work [13]. Therefore, we suppose that a student whose reviewing number is 4 requests an additional work, and an additional work is randomly allocated for simulation.

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. 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.

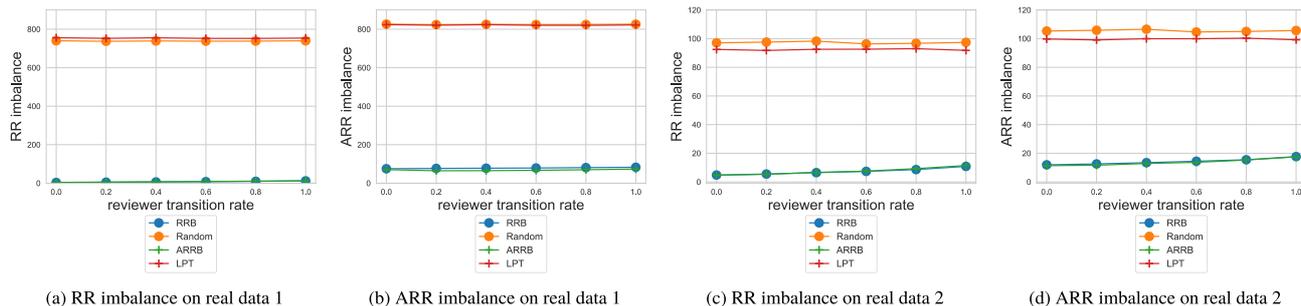
The detail of the comparison methods are as follows:

### 6.3.1 Naive Allocation Algorithm in Nonadaptive Approach

This algorithm adopts random allocations assuming both the reviewing number and the reviewed number for all students are 3. Under the simulation, when a student's actual reviewing number is smaller than 3, works are randomly selected from the works allocated in advance. In addition, for students whose reviewing number is larger than 3, an additional work is randomly selected from the works which are not allocated beforehand. We denote this algorithm as Random.

### 6.3.2 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 supposing 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 [7]; hence, we denote this algorithm as LPT.



**Fig. 6** Experimental results on the simulation based on real data complemented by the reviewer transition model when  $\lambda = 1$ .

**Table 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 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

viewing 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  $\lambda$ , 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 Fig. 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 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 Sect. 5 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. Note that we cannot find any tendency as  $P$  changes because two nonadaptive comparison algorithms do not use the reviewing order for simulation. 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 Fig. 6 (a)(c)(d). However, we can observe that ARRB is superior to RRB in Fig. 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.

## 6.4 Experimental Results

### 6.4.1 Simulation Based on Real Data with the Reviewing Order in Sect. 6.1.1

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  $\lambda$ , which is used for the ARR imbalance and ARRB, is set to 1. The results are shown in Tables 1 and 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 Sect. 5) is 50, and the results are satisfied with this upper bound.

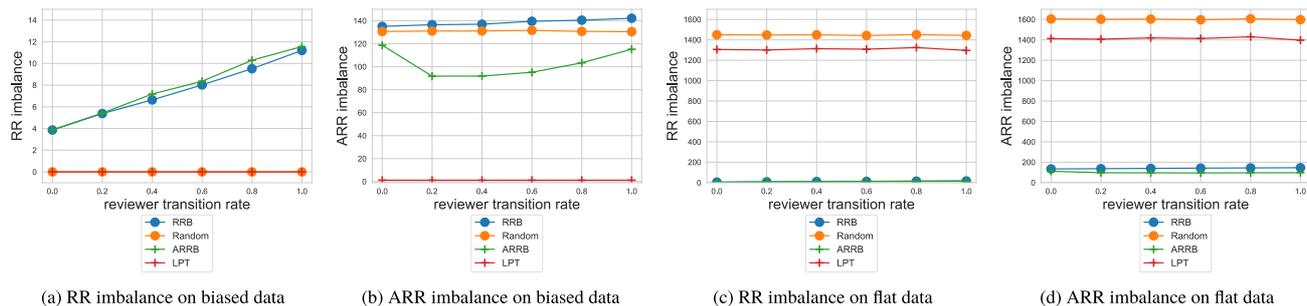
ARRB obtains results that are superior to RRB regarding both RR imbalance and ARR imbalance on the real data 1 dataset. In contrast, RRB is superior to ARRB regarding both RR imbalance and ARR imbalance on the real data 2 dataset. These results do not consist with the aim of RRB and ARRB. We compare and examine RRB and ARRB in more detail in subsequent experiments.

### 6.4.2 Simulation Based on Real Data with the Reviewing Order in Sect. 6.1.2

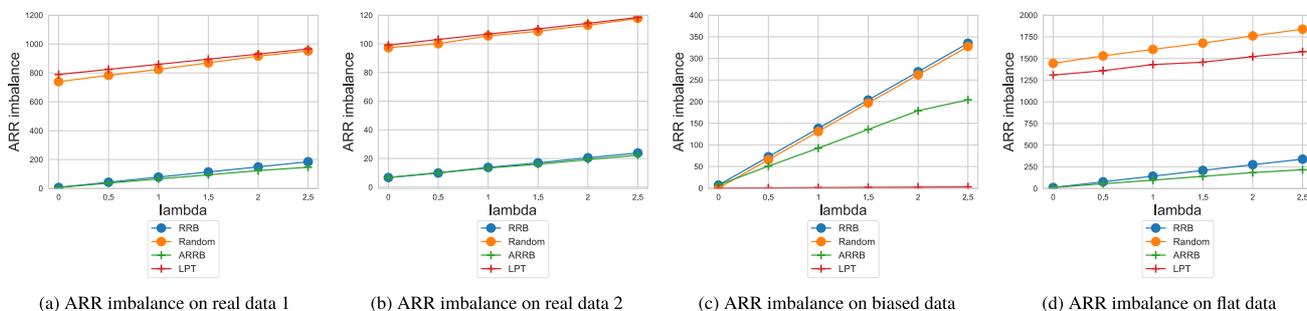
We use the real data 1 and real data 2 datasets whose re-

### 6.4.3 Simulation Based on Synthetic Data with the Reviewing Order in Sect. 6.1.2

We conducted an experiment similar to the second experiment but on the biased data and the flat data. The result is shown in Fig. 7. We can confirm that the upper bound of the RR imbalance described in Sect. 5 is established. In the case of flat data, the proposed algorithm greatly outperforms the existing algorithm, but in the biased data the



**Fig. 7** Experimental results on the simulation based on synthetic data complemented by the reviewer transition model when  $\lambda = 1$ .



**Fig. 8** Experimental results on the simulation based on real data and synthetic data complemented by the reviewer transition model when the transition rate is  $P = 0.5$ .

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 Fig. 7(a) that the RR imbalance remains at 0 with the Random algorithm. In addition, as shown in Fig. 7(b), the ARR imbalance is the smallest with LPT. However, when many students have the same number of reviews and only a few students with different reviewing numbers exist, as in real data 1 (see Fig. 4(a)), the results using Random and LPT become worse (see Fig. 6(a)(b)). Therefore, under nonadaptive allocation, the Random and LPT algorithms work effectively only in certain special situations. In addition, in Fig. 7(a)(c), little performance difference between the two proposed algorithms can be observed, and as Fig. 7(b) and (d) show, ARRB is superior to RRB.

#### 6.4.4 Experiments for $\lambda$ in ARR Imbalance

We fixed the transition rate  $P$  to 0.5 and varied  $\lambda$  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 synthetic datasets. The results are shown in Fig. 8. In Fig. 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 Fig. 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  $\lambda$  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  $\lambda$  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 performance with respect to ARR imbalance.

## 7. 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 simulation data. In future work, we plan to study how to estimate the reviewing ability values from the students' past behavioral data and estimate the final evaluation scores using the scoring results of each submission by students. Subsequently, we hope to propose a framework that could be useful throughout the peer assessment process.

## References

- [1] L. Pappano, "The year of the mooc," *The New York Times*, vol.2, no.12, p.2012, 2012.
- [2] E.F. Gehringer, "A survey of methods for improving review quality," *International Conference on Web-Based Learning*, vol.8699,

- pp.92–97, 2014.
- [3] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller, “Tuned models of peer assessment in moocs,” *Educational Data Mining 2013*, pp.153–160, 2013.
  - [4] E.S. Acosta, J.J.E. Otero, and G.C. Toletti, “Peer review experiences for mooc. development and testing of a peer review system for a massive online course,” *The New Educational Review*, vol.37, no.3, pp.66–79, 2014.
  - [5] D.F. Onah, J. Sinclair, and R. Boyatt, “Dropout rates of massive open online courses: behavioural patterns,” *International Conference on Education and New Learning Technologies*, pp.5825–5834, 2014.
  - [6] E. Er, B. Lorenzo, L. Miguel, E. Gómez Sánchez, Y.A. Dimitriadis, A. Pérez, and J. Ignacio, “Predicting student participation in peer reviews in moocs,” *Proceedings of the Second European MOOCs Stakeholder Summit 2017*, 2017.
  - [7] Y. Han, W. Wu, and Y. Pu, “Task assignment of peer grading in moocs,” *International Conference on Database Systems for Advanced Applications*, vol.10179, pp.352–363, 2017.
  - [8] D.S. Weld, E. Adar, L. Chilton, R. Hoffmann, E. Horvitz, M. Koch, J. Landay, C.H. Lin, and M. Mausam, “Personalized online education - a crowdsourcing challenge,” *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp.1–31, 2012.
  - [9] I. Abraham, O. Alonso, V. Kandylas, and A. Slivkins, “Adaptive crowdsourcing algorithms for the bandit survey problem,” *Conference on learning theory*, pp.882–910, 2013.
  - [10] D.R. Karger, S. Oh, and D. Shah, “Budget-optimal task allocation for reliable crowdsourcing systems,” *Operations Research*, vol.62, no.1, pp.1–24, 2014.
  - [11] X. Chen, Q. Lin, and D. Zhou, “Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing,” *International Conference on Machine Learning*, pp.64–72, 2013.
  - [12] Y. Yan, R. Rosales, G. Fung, and J.G. Dy, “Active learning from crowds,” *International Conference on Machine Learning*, pp.1161–1168, 2011.
  - [13] I. Estévez-Ayres, R.M.C. García, J.A. Fisteus, and C.D. Kloos, “An algorithm for peer review matching in massive courses for minimising students’ frustration,” *J. UCS*, vol.19, no.15, pp.2173–2197, 2013.
  - [14] L. Ramachandran, *Automated assessment of reviews*, North Carolina State University, 2013.
  - [15] L. de Alfaro and M. Shavlovsky, “Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments,” *Proceedings of the 45th ACM technical symposium on Computer science education*, pp.415–420, 2014.
  - [16] D. Babik, E.F. Gehringer, J. Kidd, F. Pramudianto, and D. Tinapple, “Probing the landscape: Toward a systematic taxonomy of online peer assessment systems in education,” *Educational Data Mining (Workshops)*, 2016.
  - [17] N.B. Shah, J.K. Bradley, A. Parekh, M. Wainwright, and K. Ramchandran, “A case for ordinal peer-evaluation in moocs,” *NIPS Workshop on Data Driven Education*, 2013.
  - [18] K. Raman and T. Joachims, “Methods for ordinal peer grading,” *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.1037–1046, 2014.
  - [19] J. Díez Peláez, Ó. Luaces Rodríguez, A. Alonso Betanzos, A. Troncoso, and A. Bahamonde Rionda, “Peer assessment in moocs using preference learning via matrix factorization,” *NIPS Workshop on Data Driven Education*, 2013.
  - [20] H.P. Chan and I. King, “Leveraging social connections to improve peer assessment in moocs,” *Proceedings of the 26th International Conference on World Wide Web Companion*, pp.341–349, 2017.



**Hideaki Ohashi** received his B.E. degree in 2015, and M.E. degree in Informatics from Kyoto University in 2017. He is currently a doctoral course student at the Graduate School of Informatics, Kyoto University. His research interest is educational data mining. He is a member of DBSJ.



**Yasuhito Asano** received the B.S., M.S., and D.S. degrees in information science from the University of Tokyo in 1998, 2000, and 2003 respectively. In 2003–2005, he was a research associate in the Graduate School of Information Sciences, Tohoku University. In 2006–2007, he was an assistant professor in the Department of Information Sciences, Tokyo Denki University. He joined Kyoto University in 2008 as an assistant professor. In 2009–2018, he was an associate professor in the Graduate School of Informatics, Kyoto University. Since 2019, he has been a professor at Faculty of Information Networking for Innovation and Design, Toyo University. His research interests include web mining and network algorithms. He is a member of the IEICE, IPSJ, DBSJ and OR Soc. Japan.



**Toshiyuki Shimizu** received his B.E. degree in 2003, M.S. degree in Information Science from Nagoya University in 2005, and Ph.D. degree in Informatics from Kyoto University in 2008. He is currently an assistant professor at the Graduate School of Informatics, Kyoto University. His research interests include handling semi-structured data, science data management, and scholarly databases. He is a member of ACM, IPSJ, and DBSJ.



**Masatoshi Yoshikawa** received his BE, ME, and Dr. Eng. degrees from the Department of Information Science, Kyoto University, in 1980, 1982, and 1985, respectively. From 1985 to 1993, he was with Kyoto Sangyo University. In 1993, he joined the Nara Institute of Science and Technology as an associate professor in the Graduate School of Information Science. From June 2002 to March 2006, he served as a professor at Nagoya University. Since April 2006, he has been a professor at Kyoto University. His general research interests are in the area of databases. His current research interest includes privacy protection technologies, personal data market, and multi-user routing algorithms and services. He is a member of the ACM and IPSJ.