OPTIMAL CLASS ASSIGNMENT PROBLEM: A CASE STUDY AT GUNMA UNIVERSITY

Kira, Akifumi
Kyushu University

Nagano, Kiyohito
Gunma University

Sugiyama, Manabu
Gunma University

Kamiyama, Naoyuki
Kyushu University

出版情報：日本オペレーションズ・リサーチ学会論文誌. 69 (1), pp.23-38, 2026-01-31. 日本オペレーションズ・リサーチ学会
バージョン：

Journal of the Operations Research Society of Japan
Vol. 69, No. 1, January 2026, pp. 23–38

OPTIMAL CLASS ASSIGNMENT PROBLEM: A CASE STUDY AT GUNMA UNIVERSITY

Akifumi Kira *Kyushu University*
Kiyohito Nagano *Gunma University*
Manabu Sugiyama *Gunma University*
Naoyuki Kamiyama *Kyushu University*

(Received October 11, 2024; Revised September 30, 2025)

Abstract In this study, we consider the real-world problem of assigning students to classes, where each student has a preference list, ranking a subset of classes in order of preference. We use existing successful approaches to contribute the work of Gunma University; however, in practice, one method does not always yield the best results, and new concepts and adjustments are required to find improved results depending on real instances in the field. Thus, we propose minimax-rank constrained maximum-utility matchings and a compromise between maximum-utility matchings and fair matchings, where a matching is said to be fair if it lexicographically minimizes the number of students assigned to classes not included in their choices, that assigned to their last choices, that assigned to their penultimate choices, and so on. In addition, we compare our methods with the Boston mechanism and the deferred acceptance mechanism. We also observe the potential inefficiency of the student proposing deferred acceptance mechanism with single tie-breaking, which is a hot topic in the literature on the school choice problem.

Keywords: Optimization, OR practice, matching, linear programming, network flow, lexicographic minimization

1. Introduction

In all schools and colleges, students are offered a plethora of subjects. Depending on their preferences and availability of seats, they are assigned to various classes. This is the classic assignment problem. At Gunma University, all approximately 1100 first-year students must take a subject called “Academic Literacy II” in the second semester. This subject is a compulsory elective subject in liberal arts education and consists of about 50 classes each year; each student must take one of these classes. Since the number of students in each class is limited, a questionnaire survey is conducted in advance asking students which classes they would like to take. Each student submits a preference list, ranking a subset of classes in order of preference. The university then assigns students to classes based on the preference lists so that the students can be satisfied. Until Academic Year 2017 (AY2017), the university outsourced this class assignment procedure to a system development company every year. Approximately 80 percent of the students were assigned to classes up to their third choices; however, the remaining 20 percent of students were assigned to lower preferred classes (fourth to sixth choices), and many of the students were unsatisfied with the results. Thus, the authors were asked to improve this system.

This type of matching problem, known as the class assignment problem, was introduced by Konno and Zhu [19] as a successful application of optimization theory. Their case study at the Tokyo Institute of Technology (now Institute of Science Tokyo) dealt with the problem of assigning approximately 1200 students to 12 to 15 classes. They set the maximization of the sum of students’ utilities as a criterion, and formulated the problem as a linear programming problem. It corresponds to the maximum-weight matching problem for a weighted bipartite graph. Konno [18] improved the model annually through discussions with students.

In this study, we apply the optimization approach used by Konno to our class assignment problem. However, new concepts and adjustments are required to find improved results depending on real instances in the field. Hence, we propose minimax-rank constrained maximum-utility matchings to forcibly prohibit assignment to less preferred classes. We also propose a compromise between maximum-utility matching and fair matching. Specifically, a fair matching, first studied by Huang et al. [15] as a derivative of rank-maximal matching [14, 16, 17, 20, 21], is a matching that first minimizes the number of students assigned to classes not included in their choices. Subject to that constraint, a fair matching minimizes the number of students assigned to their last choices, and so on. We apply these methods to real instance and compare them with other well-known matching mechanisms. From these experiments, we also observe the potential inefficiency of the student proposing deferred acceptance mechanism with single tie-breaking, which has been the hottest topic on the school choice problem since Erdil and Ergin [6] pointed it out. In this study, the suitability of the optimization approach for class assignments will be confirmed.

2. Basic Model

“Academic Literacy II” is a compulsory elective subject in liberal arts education at Gunma University. This subject consists of about 50 classes each year (see Table 1), and each first-year student must take one class in the second semester.

Table 1: Classes and capacities

Class	Title	Quota	
		Min	Max
c_1	Reading Soseki Natsume	7	40
c_2	Introduction to Physics	7	25
c_3	Invitation to Modern Mathematics	7	25
c_4	Let’s Think about Environmental Issues.	7	30
c_5	Learn ICT/IoT	7	25
⋮	⋮	⋮	⋮
c_{54}	History and Culture	7	25

For example, we assume the utility of each student as shown in Figure 1.

Figure 1: Utility of each student

Our goal is to find a matching that maximizes the sum of utilities (see Table 2).

Table 2: Total utility for a feasible matching

Student	Preference list						Assigned	Utility
	1st	2nd	3rd	4th	5th	6th		
s_1	c_{30}	c_{25}	c_{51}	c_{42}	c_{32}	c_{35}	c_{30}	100
s_2	c_{39}	c_{15}	c_{16}	c_{02}	c_{01}	c_{33}	c_{16}	45
s_3	c_{33}	c_{16}	c_{05}	c_{37}	c_{03}	c_{34}	c_{33}	100
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_{1138}	c_{51}	c_{33}	c_{31}	c_{29}	c_{17}	c_{19}	c_{33}	67
							Total utility	97684

Let $\mathcal{S}$ be the set of all students, and let $\mathcal{C}$ be the set of all classes. Each class c has a capacity u_c, which refers to the maximum number of students that this class can accept. Furthermore, to prevent depopulated classes, all classes must accommodate a minimum of seven students. Mathematically, a matching is a function $\mu : \mathcal{S} \to \mathcal{C}$ that satisfies

$$7 \leqq |\mu^{-1}(c)| \leqq u_c,$$

for all $c \in \mathcal{C}$, where,

$$\mu^{-1}(c) := \{s \in \mathcal{S} \mid \mu(s) = c\}.$$

We generalize the utility vector $(100, 67, 45, 30, 20, 0, -10^6)$ in Figure 1 as follows:

$$\boldsymbol{p} = (p_1, p_2, \dots, p_6, p_{\text{others}}) \in \mathbb{R}^7, \quad p_1 > p_2 > \cdots > p_6 > p_{\text{others}}.$$

For each $s \in \mathcal{S}$ and each $c \in \mathcal{C}$, we define an optimization variable x_{sc} such that

$$x_{sc} = \begin{cases} 1 & \text{if student } s \text{ is assigned to class } c \\ 0 & \text{otherwise,} \end{cases}$$

and define a constant p_{sc} in the following manner:

$$p_{sc} = \begin{cases} p_1 & \text{if class } c \text{ is the 1st choice of student } s \\ p_2 & \text{if class } c \text{ is the 2nd choice of student } s \\ p_3 & \text{if class } c \text{ is the 3rd choice of student } s \\ p_4 & \text{if class } c \text{ is the 4th choice of student } s \\ p_5 & \text{if class } c \text{ is the 5th choice of student } s \\ p_6 & \text{if class } c \text{ is the 6th choice of student } s \\ p_{\text{others}} & \text{otherwise.} \end{cases}$$

Our problem can be expressed by the following integer programming problem.

$$\text{CA}(\boldsymbol{p}) \quad \left| \begin{array}{lll} \text{Maximize} & \displaystyle\sum_{s\in\mathcal{S}}\sum_{c\in\mathcal{C}} p_{sc}x_{sc} & \text{(2.1a)} \\ \text{subject to} & \displaystyle\sum_{c\in\mathcal{C}} x_{sc} = 1, \quad \forall s \in \mathcal{S} & \text{(2.1b)} \\ & \displaystyle\sum_{s\in\mathcal{S}} x_{sc} \geqq 7, \quad \forall c \in \mathcal{C} & \text{(2.1c)} \\ & \displaystyle\sum_{s\in\mathcal{S}} x_{sc} \leqq u_c, \quad \forall c \in \mathcal{C} & \text{(2.1d)} \\ & x_{sc} \in \{0,1\}, \quad \forall s \in \mathcal{S},\ \forall c \in \mathcal{C}. & \text{(2.1e)} \end{array} \right.$$

Because the coefficients of the constraints form a totally unimodular matrix [22], we can relax the constraints (2.1e) to

$$0 \leqq x_{sc} \leqq 1, \quad \forall s \in \mathcal{S},\ \forall c \in \mathcal{C},$$

and guarantee zero relaxation gap (e.g., see Conforti et al. [5, Theorem 4.5]). We note that this linear programming (LP) problem is the so-called Hitchcock transportation problem.

The class assignment problem is closely related to the school choice problem introduced by Abdulkadiroğlu and Sönmez [3] as an application of matching theory. The important difference between the school choice problem and the class assignment problem is that in the school choice problem, each school has a priority ranking over students; in the class assignment problem, each class has no preference for the students.

3. Network Flow Formulation

We can solve CA($\boldsymbol{p}$) by using a general purpose LP solver. Alternatively, since CA($\boldsymbol{p}$) is a Hitchcock transportation problem, we can also convert it into a minimum-cost flow problem. We define the corresponding network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with capacities $u : \mathcal{E} \to \mathbb{R}_{\geq 0}$, costs $h : \mathcal{E} \to \mathbb{R}$, and supplies/demands $b : \mathcal{V} \to \mathbb{R}$ as follows: For each student s in $\mathcal{S}$, we create a node s. For each class c in $\mathcal{C}$, we create a node c. Moreover, we add a super source o and a super sink t. Namely,

$$\mathcal{V} = \mathcal{S} \cup \mathcal{C} \cup \{o, t\}.$$

The super source o connects all student nodes and the super sink t is connected by each class node. For each s in $\mathcal{S}$ and each c in $\mathcal{C}$, there exists an arc that connects s to c. Namely,

$$\mathcal{E} = (\{o\} \times \mathcal{S}) \cup (\mathcal{S} \times \mathcal{C}) \cup (\mathcal{C} \times \{t\}).$$

The capacity function u, the cost function h, and the supply/demand function b are given by

$$u(e) = \begin{cases} u_c - 7 & \text{if } e = (c, t) \in \mathcal{C} \times \{t\}, \\ 1 & \text{otherwise.} \end{cases}$$

$$h(e) = \begin{cases} -p_{sc} & \text{if } e = (s, c) \in \mathcal{S} \times \mathcal{C}, \\ 0 & \text{otherwise.} \end{cases}$$

$$b(v) = \begin{cases} |\mathcal{S}| & \text{if } v = o, \\ 0 & \text{if } v \in \mathcal{S}, \\ -7 & \text{if } v \in \mathcal{C}, \\ 7|\mathcal{C}| - |\mathcal{S}| & \text{if } v = t. \end{cases}$$

Figure 2 illustrates the network.

An o-t b-flow is a function $f : \mathcal{E} \to \mathbb{R}$ satisfying

$$\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b(e), \quad \forall v \in \mathcal{V},$$
$$0 \leqq f(e) \leqq u(e), \quad \forall e \in \mathcal{E},$$

where $\delta^+(v)$ and $\delta^-(v)$ denote the set of arcs $e \in \mathcal{E}$ leaving node v and entering node v, respectively. For each o-t b-flow f, the cost is given as follows:

$$\sum_{e \in \mathcal{E}} h(e) f(e).$$

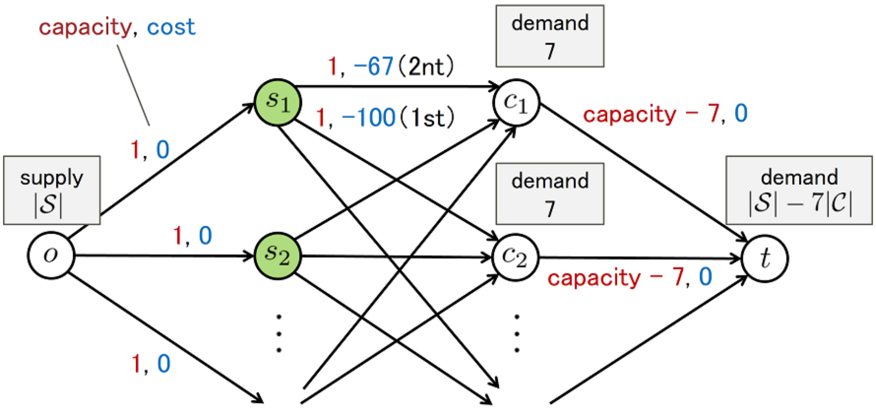

Figure 2: Minimum-cost flow formulation of CA($\boldsymbol{p}$)

The goal of the minimum-cost flow problem is to find the o-t b-flow that minimizes this cost. It is well known that this setting of the minimum-cost flow problem has an integer flow solution. Namely, we can obtain the solution f^* such that $f^*(e) \in \{0,1\}$ for every $e \in \mathcal{S} \times \mathcal{C}$.

4. Minimax-Rank Constrained Matching

We assign 1138 students to 54 classes in AY2018. Each student can choose only six classes they want to take and rank them. However, two of the classes were canceled after all the students submitted their preferences due to unforeseen circumstances. First, we solve the problem CA($\boldsymbol{p}$) for the four different utility vectors shown in Table 3, where M is a sufficiently large integer and we set

$$M := 100 \times |\mathcal{S}| + 1.$$

In Opt80, $\boldsymbol{p}$ is set so that $\frac{p_{i+1}}{p_i} \approx \frac{4}{5}$ for $i = 1,2,3,4$. Similarly, in Opt75, Opt67, and Opt50, $\boldsymbol{p}$ are set so that $\frac{p_{i+1}}{p_i} \approx \frac{3}{4}$, $\frac{2}{3}$, and $\frac{1}{2}$ for $i = 1,2,3,4$, respectively.

Table 3: Four models with different $\boldsymbol{p}$

Model	Utility vector $\boldsymbol{p}$
Opt80	$(100, 80, 64, 51, 41, 0, -M)$
Opt75	$(100, 75, 56, 42, 32, 0, -M)$
Opt67	$(100, 67, 45, 30, 20, 0, -M)$
Opt50	$(100, 50, 25, 13, 6, 0, -M)$

In the case study at the Tokyo Institute of technology, Konno [18] finally proposes the procedure for each student to declare her/his utility vector. Applying this method to our case, each student must submit $\boldsymbol{p}_s = (p_1, p_2, \ldots, p_6, p_{\text{others}})$ satisfying

$$p_1 = 100, \quad p_i \in [0, 100], \;\; i = 2, 3, \ldots, 6, \quad p_{\text{others}} = -M.$$

However, when discussing this with staffs of the Liberal Arts Education Section, they were reluctant to follow this approach. They believed that students may submit their choices without a full understanding of the procedure despite it being explained in advance. Therefore, we determined the value of $\boldsymbol{p}$ without input from the students.

We implemented our solver using Python and NetworkX [11], and executed it on a laptop PC equipped with an Intel® Core™ i7-8565U processor and 16GB memory. The computation of the minimum-cost flow for each $\boldsymbol{p}$ was completed in five seconds. Before solving the four problems, we shuffled the order of students in the input data once. This is equivalent to randomly selecting one solution from among the optimal solutions, which is like a lottery for the students. The results are shown in Table 4 and Figure 3.

Table 4: Matching results by the four models with different $\boldsymbol{p}$ in AY2018

Model	# of students (Total = 1138)							Average	
	1st	2nd	3rd	4th	5th	6th	Others	Utility	Rank
Opt80	743	266	110	14	5	0	0	90.983	1.482
Opt75	750	259	103	14	12	0	0	88.897	1.488
Opt67	758	247	103	16	11	3	0	85.838	1.492
Opt50	776	222	85	28	17	10	0	80.221	1.522

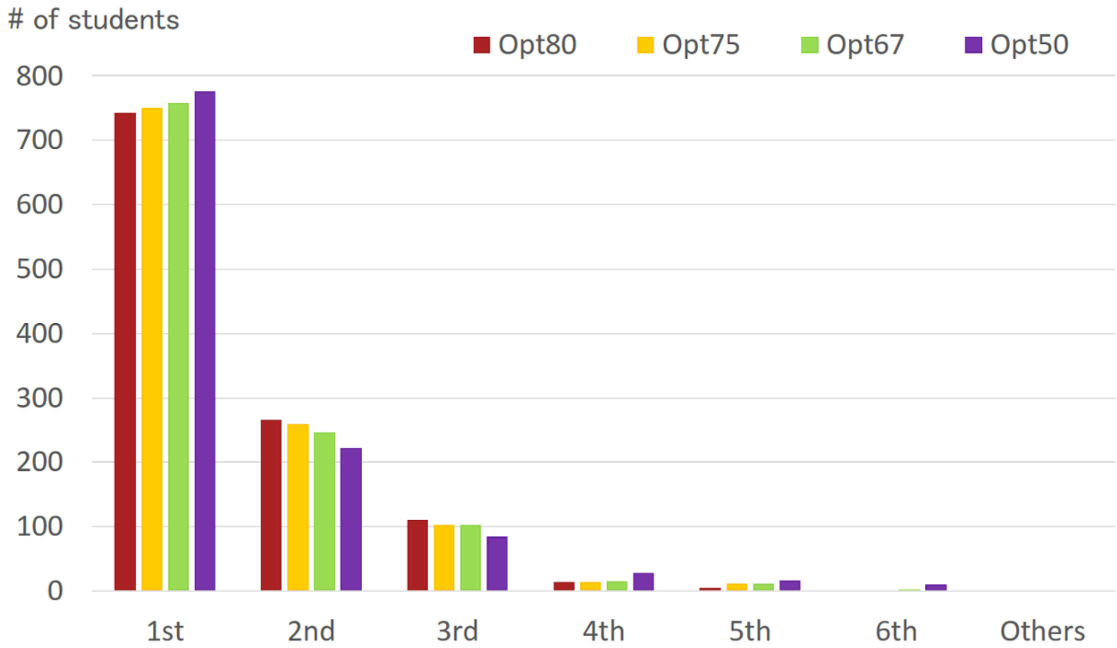

Figure 3: Matching results by the four models with different $\boldsymbol{p}$ in AY2018

The results for Opt80 and Opt75 are better than that of the other models in that no students are assigned to their sixth choices. However, in these cases, the difference in utility between the ith and $(i+1)$st choices is small. Thus, Opt67 and Opt50 may be preferable in that we can reflect the wishes of students who strongly desire to be assigned to higher preferred classes. Therefore, while adopting the utility vector of Opt67, we attempt to forcibly prohibit assignment to lower preferred classes. We can easily accomplish this by replacing the utilities of the classes with values of $-M$ or less, as in Table 5. We note that, for $i = 2, 3, 4, 5$, all students can be assigned to classes up to their ith choices if and only if Opt67 (1st to ith) has a positive optimal value. Thus, in this study, when Opt67 (1st to ith) has a negative optimal value, we will say that Opt67 (1st to ith) is infeasible. The results are shown in Table 6 and Figure 4. The final assignments are shown in Table 7.

Table 5: Restricted versions of Opt67

Model	Utility vector $\boldsymbol{p}$
Opt67 (1st to 5th)	$(100, 67, 45, 30, 20, -M, -M-1)$
Opt67 (1st to 4th)	$(100, 67, 45, 30, -M, -M-1, -M-2)$
Opt67 (1st to 3rd)	$(100, 67, 45, -M, -M-1, -M-2, -M-3)$
Opt67 (1st to 2nd)	$(100, 67, -M, -M-1, -M-2, -M-3, -M-4)$

Table 6: Matching results by the restricted versions of Opt67 in AY2018

Model	# of students (Total = 1138)							Average	
	1st	2nd	3rd	4th	5th	6th	Others	Utility	Rank
Opt67	758	247	103	16	11	3	0	85.838	1.492
Opt67 (1st to 5th)	754	252	103	16	13	0	0	85.816	1.490
Opt67 (1st to 4th)	748	256	106	28	0	0	0	85.731	1.485
Opt67 (1st to 3rd)	727	271	140	0	0	0	0	85.375	1.484
Opt67 (1st to 2nd)	infeasible								

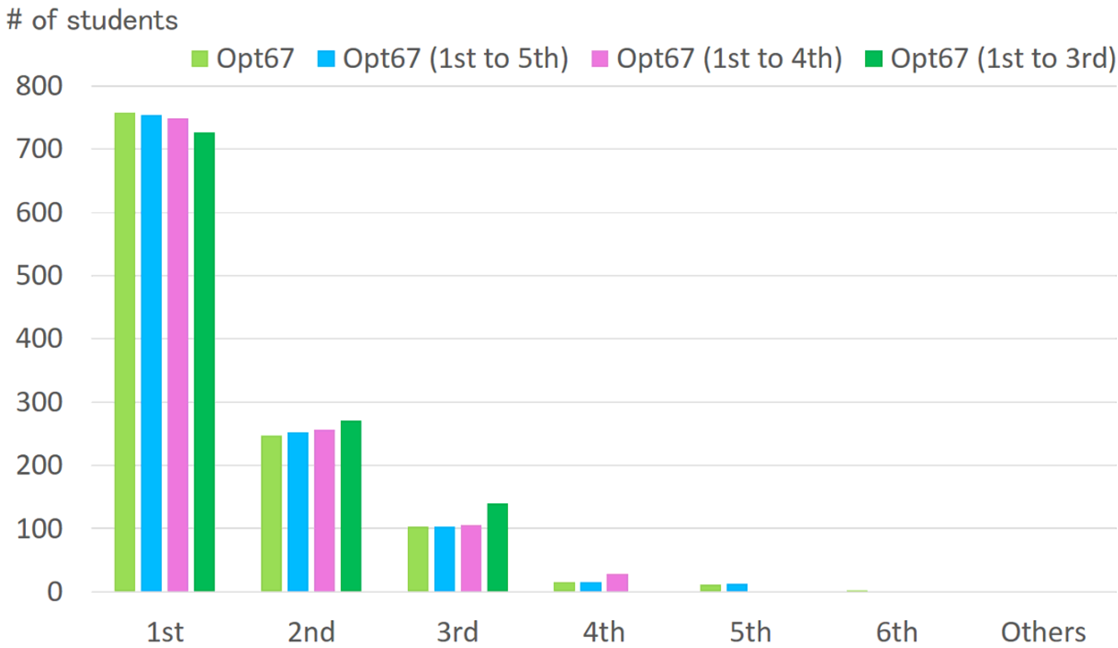

Figure 4: Matching results by the restricted versions of Opt67 in AY2018

As a result, we succeed in assigning all students to classes up to their third choice. Compared with the conventional way of outsourcing to a system development company, in which approximately 20 percent of students were assigned to lower preferred classes (fourth to sixth choices), this result shows a significant improvement. The staffs of the Liberal Arts Education Section were deeply impressed by the efficacy of the optimization theory and decided to adopt Opt67 (1st to 3rd) as the final class assignment.

Table 7: Final assignment by Opt67 (1st to 3rd) in AY2018

Class	Quota		# of students assigned					# of students ranking the class						
	Lower	Upper	Total	E	M	SS	ST	Total	1st	2nd	3rd	4th	5th	6th
1	7	25	25	11	8	2	4	93	4	8	22	15	22	22
2	7	40	40	22	12	3	3	198	94	22	13	20	21	28
3	7	30	30	6	5	6	13	147	20	25	27	18	26	31
4	Canceled		0	0	0	0	0	69	17	10	10	14	9	9
5	7	25	25	2	0	8	15	90	20	17	14	19	5	15
6	7	20	7	0	2	1	4	19	1	4	3	2	1	8
7	7	16	7	0	1	0	6	56	3	7	6	11	11	18
8	7	26	26	1	6	1	18	123	15	25	17	19	20	27
9	7	25	25	10	4	5	6	164	14	33	28	35	31	23
10	7	25	25	4	0	5	16	200	22	30	42	37	31	38
11	7	25	7	1	2	0	4	39	2	8	8	9	8	4
12	7	25	25	9	0	2	14	175	17	43	20	34	34	27
13	7	25	25	11	2	3	9	147	20	30	17	22	32	26
14	7	25	25	10	0	0	15	270	80	60	39	40	28	23
15	7	25	12	4	0	4	4	44	4	7	9	6	10	8
16	7	25	24	0	7	0	17	95	13	15	6	18	16	27
17	7	30	24	3	0	0	21	88	16	9	16	19	10	18
18	7	20	20	3	4	9	4	101	15	19	15	16	18	18
19	7	25	22	11	1	2	8	138	12	7	31	41	23	24
20	7	40	40	1	18	2	19	256	69	49	56	30	28	24
21	7	25	9	2	3	1	3	57	0	7	8	6	18	18
22	7	30	30	8	3	5	14	202	15	29	36	35	58	29
23	7	30	30	10	4	1	15	127	18	20	21	25	24	19
24	7	30	30	20	3	2	5	105	18	20	16	10	22	19
25	Canceled		0	0	0	0	0	176	41	34	37	18	24	22
26	7	40	40	0	25	0	15	263	50	59	45	42	38	29
27	7	25	10	2	2	0	6	78	2	7	10	14	21	24
28	7	25	25	1	5	0	19	280	88	47	51	36	34	24
29	7	40	30	4	8	3	15	118	4	19	19	22	22	32
30	7	18	16	5	3	6	2	78	10	13	9	10	19	17
31	7	25	25	0	19	1	5	269	36	51	46	41	45	50
32	7	24	24	0	8	1	15	120	6	20	24	23	28	19
33	7	25	25	1	19	0	5	182	43	22	34	33	17	33
34	7	25	25	4	14	1	6	175	15	30	26	38	42	24
35	7	30	13	7	2	3	1	55	5	6	14	9	7	14
36	7	15	14	8	2	2	2	53	3	9	6	11	15	9
37	7	25	10	1	2	2	5	56	2	7	11	15	10	11
38	7	20	12	1	7	0	4	66	4	5	15	12	15	15
39	7	35	35	7	12	1	15	216	28	32	34	48	34	40
40	7	20	7	2	3	0	2	31	3	1	4	10	4	9
41	7	25	14	0	1	0	13	50	8	10	7	8	8	9
42	7	25	25	12	4	6	3	99	23	16	12	13	11	24
43	7	20	20	2	3	0	15	169	28	18	40	29	25	29
44	7	25	24	1	0	3	20	102	16	12	14	14	15	31
45	7	40	40	7	16	1	16	167	43	32	29	26	20	17
46	7	16	16	3	5	0	8	53	12	7	5	11	10	8
47	7	20	20	8	4	0	8	107	26	17	19	23	8	14
48	7	40	29	1	3	4	21	87	12	16	15	15	20	9
49	7	30	7	0	0	4	3	52	3	6	11	9	9	14
50	7	25	25	0	8	0	17	272	53	51	47	45	45	31
51	7	20	20	11	3	0	6	212	32	48	33	31	32	36
52	7	25	25	0	1	1	23	133	18	20	24	18	32	21
53	7	20	13	2	1	2	8	52	7	11	10	5	8	11
54	7	25	16	2	2	0	12	54	8	8	7	8	14	9

- E = Faculty of Education, M = Faculty of Medicine, SS = Faculty of Social and Information Studies, ST = School of Science and Technology
- The order of the classes has been shuffled to ensure anonymity.

5. Optimization v.s. Boston and DA

We succeeded in assigning all students to classes up to their third choices. Nevertheless, several critical opinions were received from outside the university, stating that it is wrong to use the mathematical optimization approach and that the Gale–Shapley deferred acceptance (DA) algorithm [10] should be used instead. Although DA is a promising mechanism for the school choice problem, as mentioned in Section 2, the settings for the school choice problem and the class assignment problem are different. If we ignore the lower quota constraints (2.1c), and if we use a single tie-breaking procedure (STB), that is, if we shuffle the student list and regard it as a priority ranking over students, then we obtain a situation in which the student-proposing DA can be applied.* Therefore, we conducted additional experiments. The results of applying the student proposing DA and the Boston mechanisms,† both with STB to our problem are shown in Table 8 and Figure 5. As a result of STB, we used the same shuffled student list that was input into Opt67 (1st to 3rd).

Table 8: Matching results by the DA and Boston mechanisms in AY2018

Model	# of students (Total = 1138)							Average	
	1st	2nd	3rd	4th	5th	6th	Others	Utility	Rank
Boston with STB	783	145	64	34	21	26	65	–	$\geqq$ 1.860
DA with STB	688	196	102	47	40	22	43	–	$\geqq$ 1.939
Opt67 (1st to 3rd)	727	271	140	0	0	0	0	85.375	1.484

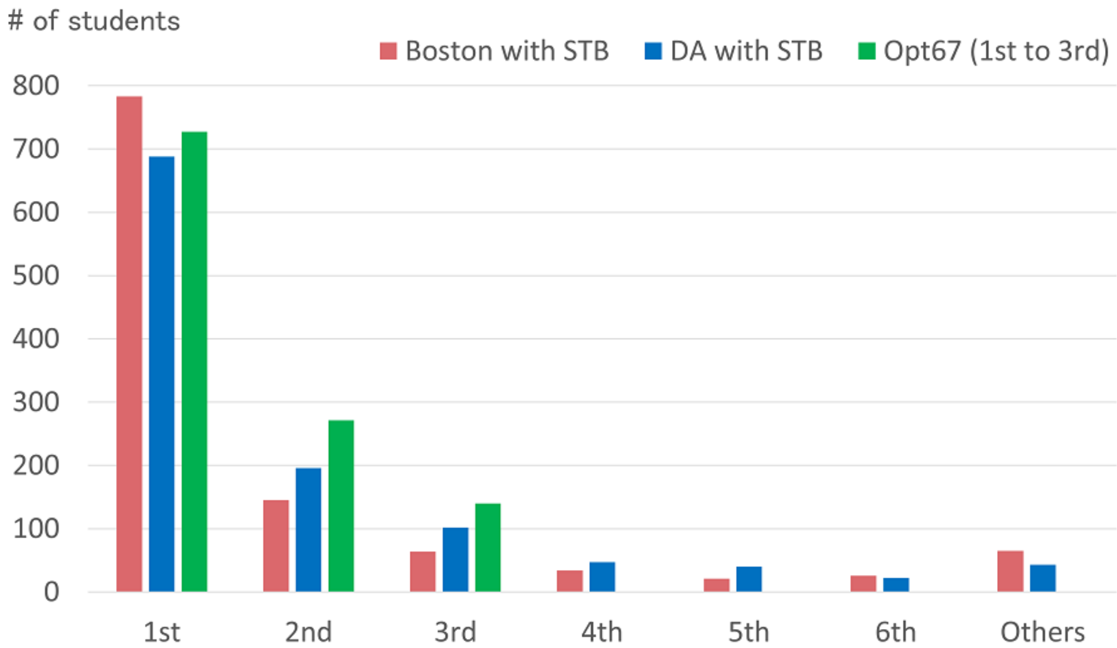

Figure 5: Matching results by the DA and Boston mechanisms in AY2018

*For stable matchings with lower quotas, see e.g., Birò et al. [4], Fleiner and Kamiyama [8], Fragiadakis et al. [9], Hamada et al. [12], Huang [13], and Yokoi [23].

†The Boston mechanism is a popular student-placement mechanism in school-choice programs around the world. In Faculty of Informatics, Gunma University, to which two of authors belong, the Boston mechanism is used when students select seminars for their third year.

Since the lower quotas are not considered, five and four classes do not reach seven students in Boston with STB and DA with STB, respectively. Nevertheless, a considerable number of students are assigned to classes not included in their choices. Thus, these results could never be considered satisfactory. Furthermore, we repeated the experiment by changing the seed used for randomization in the shuffle. Although the matching output itself changed (i.e., the pairs of students and classes changed) depending on the random seed in Opt67 (1st to 3rd), its profile (i.e. the vector whose ith component indicates the number of students assigned to their ith choices) remained completely unchanged. In contrast, in Boston with STB and DA with STB, those profiles also changed slightly. However, there was no improvement in the unfavorable situation where many students were assigned to classes not included in their choices.

Certainly, in the school choice problem, the student-proposing DA is strategy-proof for students, meaning that no student has an incentive to misstate her/his true preference list. Furthermore, it yields a student optimal stable matching within all stable matchings; that is, it yields a stable matching that is not Pareto dominated for students by any other stable matching. By contrast, the optimization approach is not strategy-proof for students, but it yields a student optimal matching in all (not necessarily stable) matchings; that is, it yields a matching that is not Pareto dominated for students by any other matching. Indeed, assuming that there exists a matching M_{PD} that Pareto dominates a matching M_{opt} obtained by solving the optimization problem CA($\boldsymbol{p}$) for some $\boldsymbol{p}$, then M_{PD} would exceed M_{opt} in terms of the objective value of the optimization problem, which contradicts the fact that M_{opt} is the optimal solution. This is the theoretical justification for our approach using optimization.

In fact, the inefficiency of the DA mechanism with random tie-breaking is known in the literature on the school choice problem. According to Erdil and Ergin [6], versions of the DA algorithm are being adopted in several school choice districts in the U.S. To their knowledge, all of them employ a random tie-breaking rule when faced with indifferences in the priority orders. Erdil and Ergin [6] note that this may cause a significant loss of efficiency, because such a random tie-breaking rule introduces artificial stability constraints. Abdulkadiroğlu et al. [2] prove that there exists no strategy-proof mechanism (stable or otherwise) which Pareto improves on the DA with single tie-breaking, and state that the potential inefficiency of the DA with single tie-breaking is the cost of strategy-proofness. Furthermore, Abdulkadiroğlu et al. [1] show a problem setting in which the Boston mechanism with random tie-breaking Pareto dominates the DA mechanism with random tie-breaking in ex ante welfare (i.e., in the sense of expected utility). This is interesting because it is in contrast to the fact shown by Ergin and Sönmez [7] that the Boston mechanism is (weakly) Pareto dominated by the DA mechanism when priorities are strict. Based on these findings from previous studies and the results of our additional experiments applying it to the real instance, we conclude that DA with STB should not be used for the class assignment procedure at Gunma University.

6. Maximum-Utility Matching and Fair Matching

We assigned 1123 students to 50 classes in AY2019. Based on the successful results in AY2018, the authors expected that, despite a reduction of two classes in the AY2019, it would still be possible to assign all students to classes up to their third choices. Even in the most pessimistic scenario, we were confident that students would be assigned to classes up to their fifth choices. Therefore, the staffs of the Liberal Arts Education Section decided to reduce the number of classes ranked by each student from six to five. As in AY2018, we solved the restricted versions of Opt67 and compared the results with the matching results of the Boston mechanism with STB and the student proposing DA mechanism with STB. The results are shown in Table 9 and Figure 6.

Table 9: Matching results by the restricted versions of Opt67 in AY2019

Model	# of students (Total = 1123)						Average	
	1st	2nd	3rd	4th	5th	Others	Utility	Rank
Boston with STB	742	148	74	40	29	90	–	≧ 1.874
DA with STB	659	193	111	61	29	70	–	≧ 1.947
Opt67 (1st to 4th)	711	257	116	39	0	0	84.336	1.540
Opt67 (1st to 3rd)	infeasible							

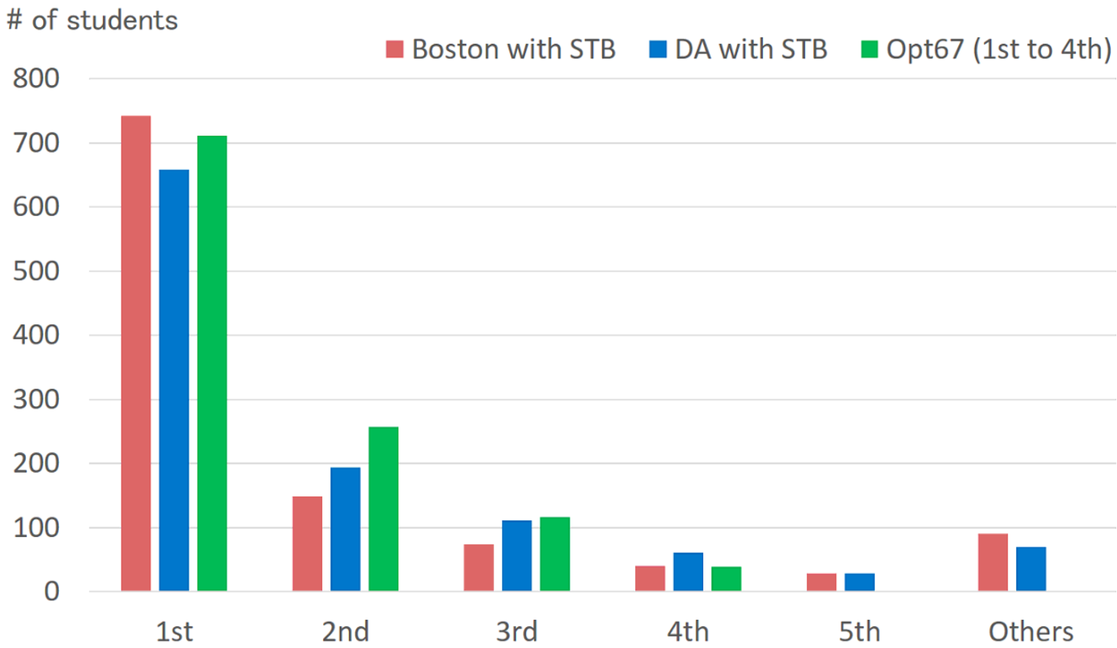

Figure 6: Matching results by the restricted versions of Opt67 in AY2019

We note again that the lower quota constraints were not considered both in Boston with STB and DA with STB. As a result, four classes and three classes did not reach seven students in each scenario, respectively.

Contrary to our expectations, Opt67 (1st to 3rd) turned out to be infeasible. This was caused by the lower quotas (2.1c). If we remove the lower quota constraints, we can assign all students to classes up to their third choices. As a necessary condition for the existence of a feasible solution, it must satisfy

$$\sum_{i=1}^{3} n_{ci} \geq 7, \tag{6.1}$$

for all $c \in \mathcal{C}$, where n_{ci} represents the number of students whose i-th choices are c. Nevertheless, two classes break the condition (6.1). That is, the reason why some students are assigned to their fourth choice is the policy of the university, which wants to prevent the occurrence of depopulated classes.

In order to reduce the number of students who receive inopportune results, we applied well-known approaches in the study of profile-based matchings (see Table 10). The first is the rank-maximal matching approach [14, 16, 17, 20, 21]. A matching is said to be rank-maximal if it lexicographically maximizes the number of students assigned to their first choices, then second choices, and so on. Let $\mathcal{M}^{\min}_{\text{others}}$ be the set of all maximum cardinality matchings that minimize the number of students assigned to classes not included in their choices. We are here concerned with rank-maximal matchings within $\mathcal{M}^{\min}_{\text{others}}$. We can obtain such a constrained rank-maximal matching by solving CA($\boldsymbol{p}$) for $\boldsymbol{p} = (N^3, N^2, N, 1, 0, -L)$, where N and L are sufficiently large integers and we set

$$N := |\mathcal{S}| + 1,$$
$$L := N^3|\mathcal{S}| + 1.$$

Hereinafter, we will refer to rank-maximal matchings within $\mathcal{M}^{\min}_{\text{others}}$ as simply rank-maximal matchings.

The second is the fair matching approach [15]. A (maximum-cardinality) matching is said to be fair if it lexicographically minimizes the number of students assigned to classes not included in their choices, the number of students assigned to their last choices, and so on. We can obtain a fair matching by solving CA($\boldsymbol{p}$) for $\boldsymbol{p} = (0, -1, -N, -N^2, -N^3, -N^4)$.

The third approach is our own method. We propose a compromise between maximum-utility matchings and fair matchings. Specifically, we lexicographically minimize the number of students assigned to classes not included in their choices, the number of students assigned to their fifth choices, and the number of students assigned to their fourth choices. In addition, we maximize the sum of the utilities of students assigned to their first to third choices with respect to the utility vector of Opt67. We can realize this by solving CA($\boldsymbol{p}$) for $\boldsymbol{p} = (100, 67, 45, -M, -MN, -MN^2)$. We denote this model as Opt67×Fair.

Table 10: Three profile-based models

Model	Utility vector $\boldsymbol{p}$
Rank-maximal	$(N^3, N^2, N, 1, 0, -L)$
Fair	$(0, -1, -N, -N^2, -N^3, -N^4)$
Opt67×Fair	$(100, 67, 45, -M, -MN, -MN^2)$

In all three profile-based models, the corresponding utility vectors still satisfy $p_1 > p_2 > \cdots > p_5 > p_{\text{others}}$. Therefore, these models also yield student optimal matchings that are not Pareto dominated for students by any other matching. The results are shown in Table 11 and Figure 7. The final assignment is shown in Table 12.

Table 11: Matching results by the profile-based models in AY2019

Model	# of students (Total = 1123)						Average	
	1st	2nd	3rd	4th	5th	Others	Utility (Opt67)	Rank
Opt67 (1st to 4th)	711	257	116	39	0	0	84.336	1.540
Rank-maximal	736	216	109	39	23	0	84.245	1.573
Fair	584	435	97	7	0	0	82.030	1.579
Opt67×Fair	682	278	156	7	0	0	83.754	1.544

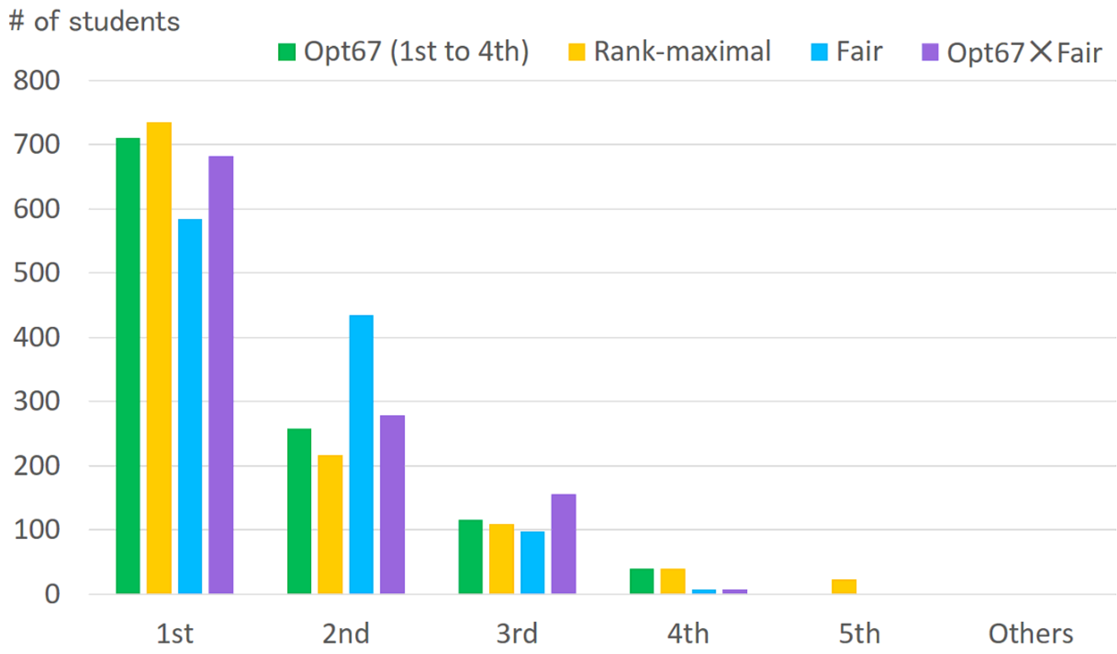

Figure 7: Matching results by the profile-based models in AY2019

Finally, we could reduce the number of students assigned to their fourth choices to seven. After discussing these results with the members concerned, they agreed that either the Fair category or Opt67×Fair can be adopted; however it is difficult to determine which of the two is preferable. Thus, Opt67×Fair was adopted by a majority vote among the authors and some of the faculty members involved.

Since AY2018, the authors have continued this initiative every year at Gunma University. Table 13 shows the models adopted in each academic year. From AY2022 to AY2024, due to the lower quotas (2.1c), a very small number of students were assigned to their fourth choices. In AY2022 and AY2024, Opt67×Fair was adopted by majority vote, as in AY2019. However, in AY2023, Fair matching was adopted by majority vote. This indicates that whether Opt67×Fair or Fair matching is preferred depends on the real instance of that year.

Table 13: Adopted models from AY2018 to AY2024

Academic year	Adopted model
2018	Opt67 (1st to 3rd)
2019	Opt67×Fair
2020	Opt67 (1st to 3rd)
2021	Opt67 (1st to 3rd)
2022	Opt67×Fair
2023	Fair
2024	Opt67×Fair

Table 12: Final assignment by Opt67×Fair in AY2019

Class	Quota		# of students assigned					# of students ranking the class					
	Lower	Upper	Total	E	M	SS	ST	Total	1st	2nd	3rd	4th	5th
1	7	25	25	15	2	4	4	143	15	29	27	41	31
2	7	25	18	0	2	4	12	58	8	6	11	19	14
3	7	30	11	2	2	1	6	31	1	8	8	8	6
4	7	25	25	4	6	2	13	104	14	19	12	21	38
5	7	25	13	0	1	0	12	54	2	6	16	11	19
6	7	36	36	20	2	0	14	90	18	24	18	18	12
7	7	40	40	2	18	1	19	123	9	23	31	31	29
8	7	32	18	2	0	0	16	43	5	6	9	9	14
9	7	25	16	0	0	14	2	44	10	8	9	9	8
10	7	25	25	1	2	0	22	70	13	11	17	18	11
11	7	20	13	0	1	2	10	70	6	11	11	12	30
12	7	40	40	0	14	1	25	203	28	51	43	46	35
13	7	40	40	2	29	0	9	241	63	65	38	42	33
14	7	25	25	0	10	0	15	339	71	68	96	57	47
15	7	25	25	3	4	0	18	155	37	35	29	23	31
16	7	30	18	2	0	5	11	55	8	15	9	17	6
17	7	16	16	0	1	0	15	70	16	12	11	14	17
18	7	40	33	6	4	2	21	104	11	19	21	16	37
19	7	18	18	11	1	2	4	45	9	8	12	8	8
20	7	25	25	2	2	7	14	82	15	16	14	25	12
21	7	25	17	12	4	1	0	67	7	8	20	20	12
22	7	25	25	1	2	1	21	190	61	38	23	19	49
23	7	25	24	10	2	9	3	56	12	9	16	9	10
24	7	12	7	1	3	0	3	13	0	4	4	4	1
25	7	25	25	1	21	0	3	171	46	43	29	32	21
26	7	25	7	3	0	2	2	48	3	9	6	18	12
27	7	20	20	9	1	3	7	116	16	21	29	22	28
28	7	30	30	10	7	0	13	140	29	46	23	21	21
29	7	25	8	3	0	0	5	45	2	6	9	11	17
30	7	40	40	12	17	2	9	252	97	40	36	44	35
31	7	20	20	4	12	0	4	164	41	29	31	28	35
32	7	16	9	1	0	0	8	39	2	8	9	8	12
33	7	25	25	9	7	3	6	84	9	12	17	15	31
34	7	25	23	8	7	1	7	78	5	19	17	17	20
35	7	25	25	6	9	1	9	199	17	25	49	65	43
36	7	15	15	3	2	1	9	98	18	19	21	21	19
37	7	20	7	1	2	3	1	25	1	3	1	9	11
38	7	25	25	1	12	0	12	375	131	92	52	50	50
39	7	30	13	1	0	1	11	61	9	9	10	15	18
40	7	25	25	1	2	5	17	72	21	10	21	10	10
41	7	30	30	18	2	2	8	87	16	26	17	13	15
42	7	25	25	13	0	2	10	112	21	25	20	28	18
43	7	20	20	2	2	0	16	206	19	37	50	40	60
44	7	20	7	0	3	1	3	9	1	0	1	5	2
45	7	35	35	1	22	2	10	185	13	34	49	44	45
46	7	40	40	0	8	1	31	214	77	48	46	26	17
47	7	30	11	0	0	7	4	41	7	3	13	6	12
48	7	20	20	4	3	4	9	83	7	13	15	27	21
49	7	40	40	10	14	3	13	141	32	28	30	34	17
50	7	25	25	12	2	1	10	120	44	19	17	17	23

- E = Faculty of Education, M = Faculty of Medicine, SS = Faculty of Social and Information Studies, ST = School of Science and Technology
- The order of the classes has been shuffled to ensure anonymity.

7. Concluding Remarks

Through a seven-year case study of the class assignment problem at Gunma University, we have augmented the optimization approach introduced by Konno [18] and applied them to real field data. Our own versions include the minimax-rank constrained maximum-utility matching and a compromise between maximum-utility matchings and fair matchings. Our results show that we have succeeded in reducing the percentage of students not assigned to classes up to their third choices from 20 percent to almost zero (zero percent in AY2018 and 0.62 percent in AY2019). In addition, we also could observe the potential inefficiency of the student proposing deferred acceptance mechanism with single tie-breaking, which is well known in the literature on the school choice problem. We believe that there are no perfect model for the class assignment problem, and there is still scope for discussion and further adjustments depending on real field data provided.

Acknowledgments

The authors would like to thank Dr. Kenji Kubota, Executive Vice President of Gunma University at that time, who asked us to improve the class assignment procedure. The authors are also deeply grateful to all the staffs of the Liberal Arts Education Section, Gunma University. Without their cooperation, the optimization techniques would never have been used in university, and this investigation would not have been possible. We would like to thank Editage (www.editage.com) for English language editing.

Akifumi Kira was supported in part by JSPS KAKENHI Grant Numbers 17K12644, 21K11766, 23H01030, and 25K15005, Japan. Kiyohito Nagano was supported by JSPS KAKENHI Grant Numbers 17K00036 and 23H01030, Japan. Manabu Sugiyama was supported by JSPS KAKENHI Grant Number 20K01847, Japan. Naoyuki Kamiyama was supported in part by JST, PRESTO Grant Number JPMJPR1753, Japan.

References

[1] A. Abdulkadiroğlu, Y.-K. Che, and Y. Yasuda: Resolving conflicting preferences in school choice: The “Boston mechanism” reconsidered. *American Economic Review*, **101-1** (2011), 399–410.

[2] A. Abdulkadiroğlu, P. Pathak, and A.E. Roth: Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. *American Economic Review*, **99-5** (2009), 1954–1978.

[3] A. Abdulkadiroğlu and T. Sönmez: School choice: A mechanism design approach. *American Economic Review*, **93-3** (2003), 729–747.

[4] P. Birò, T. Fleiner, R.W. Irving, and D. Manlove: The college admissions problem with lower and common quotas. *Theoretical Computer Science*, **411-34–36** (2010), 3136–3153.

[5] M. Conforti, G. Cornuéjols, and G. Zambelli: *Integer Programming* (Springer, 2014).

[6] A. Erdil and H. Ergin: What’s the matter with tie-breaking? Improving efficiency in school choice. *American Economic Review*, **98-3** (2008), 669–689.

[7] H. Ergin and T. Sönmez: Games of school choice under the Boston mechanism. *Journal of Public Economics*, **90-1–2** (2006), 215–237.

[8] T. Fleiner and N. Kamiyama: A matroid approach to stable matchings with lower quotas. *Mathematics of Operations Research*, **41-2** (2016), 734–744.

[9] D. Fragiadakis, A. Iwasaki, P. Troyan, S. Ueda, and M. Yokoo: Strategyproof matching with minimum quotas. *ACM Transactions on Economics and Computation*, **4-1** (2016), 2167–8375.

[10] D. Gale and L.S. Shapley: College admissions and the stability of marriage. *The American Mathematical Monthly*, **69-2** (1962), 9–15.

[11] A.A. Hagberg, D.A. Schult, and P.J. Swart: Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, and J. Millman (eds.): *Proceedings of the 7th Python in Science Conference (SciPy2008)* (Pasadena, CA USA, 2008), 11–15.

[12] K. Hamada, K. Iwama, and S. Miyazaki: The hospitals/residents problem with lower quotas. *Algorithmica*, **74-1** (2016), 440–465.

[13] C.-C. Huang: Classified stable matching. *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, (Society for Industrial and Applied Mathematics, 2010), 1235–1253.

[14] C.-C. Huang, N. Kakimura, and N. Kamiyama: Exact and approximation algorithms for weighted matroid intersection. *Mathematical Programming*, **177-1–2** (2019), 85–112.

[15] C.-C. Huang, T. Kavitha, K. Mehlhorn, and D. Michail: Fair matchings and related problems. *Algorithmica*, **74-3** (2016), 1184–1203.

[16] R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K.E. Paluch: Rank-maximal matchings. *ACM Transactions on Algorithms*, **2-4** (2006), 602–610.

[17] T. Kavitha and C.D. Shah: Effcient algorithms for weighted rank-maximal matchings and related problems. In *Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC)*, (Springer, 2006), 153–162.

[18] H. Konno: *Introduction to Mathematical Decision Making: Campus OR* (Asakura Publishing, 1992). (in Japanese)

[19] H. Konno and Z. Zhu: The optimal class assignment problem: a case study at Tokyo Institute of Technology. *Communications of Operations Research of Japan*, **36** (1991), 85–89. (in Japanese)

[20] D. Michail: Reducing rank-maximal to maximum weight matching. *Theoretical Computer Science*, **389-1–2** (2007), 125–132.

[21] K. Paluch: Capacitated rank-maximal matchings. In *Proceedings of the 8th International Conference on Algorithms and Complexity (CIAC)*, (Springer, 2013) 324–335.

[22] A. Schrijver: *Theory of Linear and Integer Programming* (John Wiley & Sons, 1998).

[23] Y. Yokoi: A generalized polymatroid approach to stable matchings with lower quotas. *Mathematics of Operations Research*, **42-1** (2017), 238–255.

Akifumi Kira
Institute of Mathematics for Industry
Kyushu University
744 Motooka, Nishi-ku
Fukuoka 819-0395, Japan
E-mail: kira@imi.kyushu-u.ac.jp