Strategic Issues in College Admissions with Score-Limits

Kamiyama, Naoyuki
Institute of Mathematics for Industry, Kyushu University

出版情報：Operations Research Letters. 45 (2), pp.105-108, 2017-01-11. Elsevier
バージョン：

Strategic Issues in College Admissions with Score-Limits

Naoyuki Kamiyama[a,b]

[a] *Institute of Mathematics for Industry, Kyushu University, Japan.*
[b] *JST, PRESTO, Japan.*

Abstract

We consider strategic problems in college admissions with score-limits introduced by Biró and Kiselgof. We first consider the problem of deciding whether a given applicant can cheat the algorithm of Biró and Kiselgof so that this applicant is assigned to a more preferable college. We prove its polynomial-time solvability. In addition, we consider the situation in which all applicants strategically behave. We prove that a Nash equilibrium always exists, and we can find one in polynomial time.

Keywords: college admission, score-limit, weakly stable matching

1. Introduction

The stable matching model introduced by Gale and Shapley [1] is one of the most important matching models from both the theoretical and practical viewpoints. Gale and Shapley [1] proved that there always exists a stable matching, and they proposed a polynomial-time algorithm for finding a stable matching. One of the most notable properties of the algorithm proposed by Gale and Shapley [1] is the strategy-proofness for the proposing side [2]. In this paper, we consider a problem in which we assign applicants to colleges based on their scores. In such a problem, it is desirable to treat equally applicants with the same score, i.e., we accept/reject all applicants with the same score. Based on a real system used in Hungary, Biró and Kiselgof [3] proposed a variant of the stable matching problem taking such a constraint into consideration. They proved that there always exists a stable assignment in this problem, and we can find a stable assignment in polynomial time. However, they also proved that their algorithm is not strategy-proof for applicants.

Email address: kamiyama@imi.kyushu-u.ac.jp (Naoyuki Kamiyama)

Preprint submitted to Operations Research Letters *January 5, 2017*

If an algorithm is not strategy-proof, then there exists a possibility that we can cheat this algorithm. However, it is reasonable to think that if finding a cheating strategy for this algorithm is **NP**-hard (and the size of an instance is sufficiently large), then it is not easy to manipulate it. Thus, it is important to reveal the computational complexity of the problem of finding a cheating strategy for an algorithm that is not strategy-proof.

In this paper, we consider the following strategic problem related to the model proposed by Biró and Kiselgof [3]. In this problem, we are given some applicant. Then, the goal is to decide whether this applicant can cheat the algorithm of [3] so that this applicant is assigned to a more preferable college. We prove that this problem can be solved in polynomial time (Section 3). Furthermore, we consider the situation in which all applicants strategically behave. We prove that a Nash equilibrium always exists in this situation, and we can find a Nash equilibrium in polynomial time (Section 4).

1.1. Related Work

Recently, computational problems related to manipulation of matching algorithms have been widely studied. In [4, 5, 6, 7, 8, 9, 10], the authors considered cheating strategies for the Gale–Shapley algorithm in the (classical) stable matching problem. Huang [11] considered a cheating strategy in the stable roommate problem. In [12], the authors considered a cheating strategy for the probabilistic serial rule. Nasre [13] considered a cheating strategy in the popular matching problem. Pini, Rossi, Venable, and Walsh [14] proposed a mechanism such that the problem of finding a cheating strategy for this mechanism is **NP**-hard. Matsui [15] considered a game related to cheating strategies for the Gale–Shapley algorithm. In [16], the authors considered a game related to cheating strategies for the probabilistic serial rule.

It should be noted that Fleiner and Jankó [17] proposed a choice function-based approach for the model proposed by Biró and Kiselgof [3].

2. Preliminaries

We denote by $\mathbb{Z}_+$ the set of non-negative integers. For each pair of sets X, Y, each mapping $\mu\colon X \to Y$, and each element y in Y, we define $\mu^{-1}(y)$ as the set of elements x in X such that $\mu(x) = y$. Assume that we are given a set $X = \{x_1, x_2, \ldots, x_k\}$ and a strict total order $\rhd$ on X. In addition, we assume that $x_i \rhd x_j$ for every pair of integers i, j in $\{1, 2, \ldots, k\}$ such that $i < j$. Then, we write $\rhd\colon x_1, x_2, \ldots, x_k$ for representing this strict total order $\rhd$. For each subset Y of X, an element x in Y is said to be *maximal in* Y *with respect to* $\rhd$, if $x \rhd y$ for every element y in $Y \setminus \{x\}$.

In college admissions with score-limits introduced by Biró and Kiselgof [3], we are given a set $[n] = \{1, 2, \ldots, n\}$ of *applicants* and a set C of *colleges*. Define $m := |C|$. For each applicant i in $[n]$, we are given a strict total order $\succ_i$ on $C \cup \{i\}$. For each applicant i in $[n]$, the strict total order $\succ_i$ represents the preference list of i over colleges. For each applicant i in $[n]$ and each pair of colleges c_1, c_2 in C, if $c_1 \succ_i c_2$, then i preferes c_1 to c_2. Define $\succ := (\succ_1, \succ_2, \ldots, \succ_n)$. Furthermore, we are given a capacity function $q \colon C \to \mathbb{Z}_+$. For each applicant i in $[n]$ and each college c in C, we are given a non-negative integer $s_i(c)$ that represents the *score* of i for c.

For each applicant i in $[n]$, we denote by S_i the set of strict total orders on $C \cup \{i\}$. Define $\mathbf{S} := S_1 \times S_2 \times \cdots \times S_n$. An element in $\mathbf{S}$ is called a *profile*. It should be noted that $\succ$ is a profile. A function from C to $\mathbb{Z}_+$ is called a *score-limit*. Furthermore, a mapping from $[n]$ to $[n] \cup C$ is called a *matching*, if $\mu(i) \in C \cup \{i\}$ for every applicant i in $[n]$.

Assume that we are given a profile $\rhd = (\rhd_1, \rhd_2, \ldots, \rhd_n)$ in $\mathbf{S}$. For each score-limit ℓ and each applicant i in $[n]$, we define $F_{i,\rhd}(\ell)$ as the set of colleges c in C such that $c \rhd_i i$ and $s_i(c) \geq \ell(c)$. In addition, for each score-limit ℓ and each applicant i in $[n]$ such that $F_{i,\rhd}(\ell) \neq \emptyset$, we define $f_{i,\rhd}(\ell)$ as the maximal college in $F_{i,\rhd}(\ell)$ with respect to $\rhd_i$. For each score-limit ℓ and each applicant i in $[n]$ such that $F_{i,\rhd}(\ell) = \emptyset$, we define $f_{i,\rhd}(\ell) := i$. For each score-limit ℓ and each college c in C, we define $G_{c,\rhd}(\ell)$ as the set of applicants i in $[n]$ such that $f_{i,\rhd}(\ell) = c$. For each score-limit ℓ and each college c in C such that $\ell(c) > 0$, we define a score-limit ℓ_c^- by $\ell_c^-(c) := \ell(c) - 1$ and $\ell_c^-(c') := \ell(c')$ for each college c' in $C \setminus \{c\}$. We call a score-limit ℓ an *H-feasible* score-limit with respect to $\rhd$, if for every college c in C, $|G_{c,\rhd}(\ell)| \leq q(c)$. In addition, an H-feasible score-limit ℓ with respect to $\rhd$ is called an *H-stable* score-limit with respect to $\rhd$, if for every college c in C, at least one of the following conditions holds.

1. $\ell(c) = 0$.
2. $\ell(c) > 0$ and ℓ_c^- is not an H-feasible score-limit with respect to $\rhd$.

This concept is motivated by a real system used in Hungary. (Biró and Kiselgof [3] introduced another stability concept called the *L-stability*. In this paper, we do not consider this stability concept.) Biró and Kiselgof [3] proved that there always exists an H-stable score-limit. Furthermore, they propose a polynomial-time algorithm for finding an H-stable score-limit (see the next subsection).

2.1. Algorithm of Biró and Kiselgof

Here we explain the algorithm of Biró and Kiselgof [3] for finding an H-stable score-limit. We call this algorithm the *BK-algorithm*. (Precisely speaking, Biró and Kiselgof [3] proposed this algorithm as an *applicant-oriented algorithm*. They also proposed another algorithm, called a *college-oriented algorithm*. Since an applicant-oriented algorithm is the best for applicants in some sense (see Theorem 2.1), we adopt this algorithm.) The input of the BK-algorithm is a profile $\rhd = (\rhd_1, \rhd_2, \dots, \rhd_n)$ in $\mathbf{S}$. For computing an H-stable score-limit with respect to $\succ$, we set $\rhd := \succ$.

Step 1: Define a score-limit δ_0 by $\delta_0(c) := 0$. Define a matching σ_0 by $\sigma_0(i) := i$. For each applicant i in $[n]$, set $L_0(i)$ to be the set of colleges c in C such that $c \rhd_i i$. Furthermore, set $R_0 := \{i \in [n] \mid L_0(i) \neq \emptyset\}$ and $t := 0$.

Step 2: If $R_t = \emptyset$, then output δ_t and halt. Otherwise, set i_t to be an applicant in R_t, and find the maximal college $\overline{c}_t$ in $L_t(i_t)$ with respect to $\rhd_{i_t}$. Furthermore, set π_t to be the same matching as σ_t except that $\pi_t(i_t) = \overline{c}_t$.

Step 3: If $|\pi_t^{-1}(\overline{c}_t)| \leq q(\overline{c}_t)$, then go to **(a)**. Otherwise, go to **(b)**.

(a) Set $\delta_{t+1} := \delta_t$ and $\sigma_{t+1} := \pi_t$. In addition, set $L_{t+1}(i_t) := L_t(i_t) \setminus \{\overline{c}_t\}$, and $L_{t+1}(i) := L_t(i)$ for each applicant i in $[n] \setminus \{i_t\}$.

(b) Set $\Delta_t := \min\{s_i(\overline{c}_t) \mid i \in \pi_t^{-1}(\overline{c}_t)\} + 1$. Furthermore, set δ_{t+1} to be the same score-limit as δ_t except that $\delta_{t+1}(\overline{c}_t) = \Delta_t$. Set σ_{t+1} to be a matching such that

$$\sigma_{t+1}(i) = \begin{cases} i & \text{if } i \in \pi_t^{-1}(\overline{c}_t) \text{ and } s_i(\overline{c}_t) < \Delta_t \\ \sigma_t(i) & \text{otherwise.} \end{cases}$$

For each applicant i in $[n]$, set

$$L_{t+1}(i) := \begin{cases} L_t(i) \setminus \{\overline{c}_t\} & \text{if (i) } i = i_t, \text{ or (ii) } i \neq i_t,\ s_i(\overline{c}_t) < \Delta_t \\ L_t(i) & \text{otherwise.} \end{cases}$$

Set $R_{t+1} := \{i \in [n] \mid L_{t+1}(i) \neq \emptyset, \sigma_{t+1}(i) = i\}$, and $t := t + 1$. Then, go back to **Step 2**.

The BK-algorithm is clearly a polynomial-time algorithm (we assume that for every applicant i in $[n]$ and every pair of elements d_1, d_2 in $C \cup \{i\}$, we can check in $O(1)$ time whether $d_1 \rhd_i d_2$). It is known [3, Theorem 3.1] that an output ℓ of the BK-algorithm with an input profile $\rhd$ in $\mathbf{S}$ is an H-stable score-limit with respect to $\rhd$. The following property of this algorithm is known.

Theorem 2.1 (Biró and Kiselgof [3, Theorem 4.1])**.** *Assume that we are given an output ℓ of the BK-algorithm with an input profile $\rhd$ in $\mathbf{S}$. Then, for every H-stable score-limit ℓ' with respect to $\rhd$ and every college c in C, we have $\ell(c) \le \ell'(c)$.*

In **Step 2** of the BK-algorithm, there exists a freedom in the choice of i_t. However, as proved below, this does not affect an output of this algorithm. Although this fact was not explicitly stated in [3], it immediately follows from Theorem 2.1.

Corollary 2.2. *An output of the BK-algorithm with an input profile $\rhd$ in* $\mathbf{S}$ *does not depend on the choice of i_t in* **Step 2***.*

Proof. Theorem 2.1 implies that for every college c in C and every pair of outputs ℓ_1, ℓ_2 of the BK-algorithm, we have $\ell_1(c) \le \ell_2(c)$ and $\ell_2(c) \le \ell_1(c)$, i.e., $\ell_1(c) = \ell_2(c)$. This completes the proof. □

In what follows, for each profile $\rhd$ in $\mathbf{S}$, we denote by $\ell_\rhd$ the output of the BK-algorithm with an input profile $\rhd$. In addition, for each profile $\rhd$ in $\mathbf{S}$, we define a matching $\mu_\rhd$ by $\mu_\rhd(i) := f_{i,\rhd}(\ell_\rhd)$ for each applicant i. If the BK-algorithm with an input profile $\rhd$ in $\mathbf{S}$ halts when $t = T$, then it is not difficult to see that $\mu_\rhd = \sigma_T$.

3. Finding a Cheating Strategy

In this section, we consider the Cheating Score-Limit Algorithm problem defined as follows. For each applicant i in $[n]$, a strict total order $\rhd_i$ in S_i is called a *cheating strategy of* i, if $\mu_\rhd(i) \succ_i \mu_\succ(i)$ holds, where $\rhd$ is the profile in $\mathbf{S}$ obtained from $\succ$ by replacing $\succ_i$ by $\rhd_i$. For each applicant i in $[n]$, we denote by CS_i the set of cheating strategies of i. Then, Cheating Score-Limit Algorithm is formally defined as follows.

Input: An applicant a in $[n]$.

Goal: Decide whether $CS_a = \emptyset$. If $CS_a \neq \emptyset$, then find a strict total order $\rhd_a$ in CS_a such that $\mu_\rhd(a) \succ_a \mu_{\rhd'}(a)$ or $\mu_\rhd(a) = \mu_{\rhd'}(a)$ for every strict total order $\rhd'_a$ in CS_a, where $\rhd$ and $\rhd'$ are the profiles in $\mathbf{S}$ obtained from $\succ$ by replacing $\succ_a$ by $\rhd_a$ and $\rhd'_a$, respectively.

In other words, the goal of this problem is to decide whether there exists an incentive for a to misreport his/her true preference.

In this section, we prove that this problem can be solved in polynomial time. We first prove necessary lemmas.

Lemma 3.1. *Assume that we are given a profile* $\rhd$ *in* $\mathbf{S}$ *and an H-feasible score-limit* ℓ *with respect to* $\rhd$*. Then, for every college* c *in* C*, we have* $\ell_{\rhd}(c) \leq \ell(c)$*.*

Proof. If ℓ is an H-stable score-limit with respect to $\rhd$, then this lemma follows from Theorem 2.1. If ℓ is not an H-stable score-limit with respect to $\rhd$, then there exists a college c in C such that $\ell(c) > 0$ and ℓ_c^- is an H-feasible score-limit with respect to $\rhd$. By repeating this, we can see that there exists an H-stable score-limit ℓ' with respect to $\rhd$ such that $\ell'(c) \leq \ell(c)$ for every college c in C. Theorem 2.1 implies that $\ell_{\rhd}(c) \leq \ell'(c)$ for every college c in C. Thus, since $\ell'(c) \leq \ell(c)$ for every college c in C, this implies that $\ell_{\rhd}(c) \leq \ell(c)$ for every college c in C. This completes the proof. □

Lemma 3.2. *Assume that we are given a profile* $\rhd = (\rhd_1, \rhd_2, \ldots, \rhd_n)$ *in* $\mathbf{S}$ *and an applicant* a *in* $[n]$*. In addition, we assume that* $\rhd_a \colon d_1, d_2, \ldots, d_{m+1}$*,* $\mu_{\rhd}(a) = d_k$*, and* $d_k \neq a$*. Let* $\rhd'_a$ *be a strict total order in* S_a *such that*

$$\rhd'_a \colon d_1, d_2, \ldots, d_k, d'_1, d'_2, \ldots, d'_{m+1-k}, \text{ and}$$
$$\{d'_1, d'_2, \ldots, d'_{m+1-k}\} = \{d_{k+1}, d_{k+2}, \ldots, d_{m+1}\}.$$

Then, we have $\ell_{\rhd}(c) = \ell_{\rhd'}(c)$ *for every college* c *in* C *(i.e.,* $\mu_{\rhd}(i) = \mu_{\rhd'}(i)$ *for every applicant* i *in* $[n]$*), where* $\rhd'$ *is the profile in* $\mathbf{S}$ *obtained from* $\rhd$ *by replacing* $\rhd_a$ *by* $\rhd'_a$*.*

Proof. Define D as the set of colleges c in C such that $\bar{c}_t = c$ for some integer t with $i_t = a$ in the BK-algorithm. Then, it is not difficult to see that the definition of the BK-algorithm implies that $D \subseteq \{d_1, d_2, \ldots, d_k\}$. Thus, the output of the BK-algorithm does not change even if we change the order of $d_{k+1}, d_{k+2}, \ldots, d_{m+1}$. This completes the proof. □

Lemma 3.3. *Assume that we are given a profile* $\rhd = (\rhd_1, \rhd_2, \ldots, \rhd_n)$ *in* $\mathbf{S}$ *and an applicant* a *in* $[n]$*. In addition, we assume that* $\rhd_a \colon d_1, d_2, \ldots, d_{m+1}$*,* $\mu_{\rhd}(a) = d_k$*, and* $d_k \neq a$*. Define a strict total order* $\rhd'_a$ *in* S_a *by*

$$\rhd'_a \colon d_k, d_{k+1}, \ldots, d_{m+1}, d_1, d_2, \ldots, d_{k-1}.$$

Then, we have $\ell_{\rhd}(c) \geq \ell_{\rhd'}(c)$ *for every college* c *in* C*, where* $\rhd'$ *is the profile in* $\mathbf{S}$ *obtained from* $\rhd$ *by replacing* $\rhd_a$ *by* $\rhd'_a$*.*

Proof. If we can prove that $\ell_{\rhd}$ is an H-feasible score-limit with respect to $\rhd'$, then this lemma follows from Lemma 3.1. If we can prove that $f_{a,\rhd'}(\ell_{\rhd}) = f_{a,\rhd}(\ell_{\rhd})$, then we have $G_{c,\rhd'}(\ell_{\rhd}) = G_{c,\rhd}(\ell_{\rhd})$ for every college c in C. Since $\ell_{\rhd}$ is an H-feasible score-limit with respect to $\rhd$, we have $|G_{c,\rhd}(\ell_{\rhd})| \leq q(c)$ for every college c in C. Thus, this completes the proof.

It follows from $d_k \in F_{a,\rhd}(\ell_{\rhd})$ that $s_a(d_k) \geq \ell_{\rhd}(d_k)$. This implies that $d_k \in F_{a,\rhd'}(\ell_{\rhd})$. Thus, we have $f_{a,\rhd'}(\ell_{\rhd}) = d_k$. This completes the proof. □

Lemma 3.4. *Assume that we are a profile $\rhd = (\rhd_1, \rhd_2, \ldots, \rhd_n)$ in $\mathbf{S}$, an applicant a in $[n]$, and a college d in C. Furthermore, we assume that there exists a strict total order $\rhd'_a$ in S_a such that $\mu_{\rhd'}(a) = d$, where $\rhd'$ is the profile in $\mathbf{S}$ obtained from $\rhd$ by replacing $\rhd_a$ by $\rhd'_a$. Then, we have $\mu_{a,\rhd''}(a) = d$, where we define a strict total order $\rhd''_a$ in S_a by $\rhd''_a \colon d, a, \ldots$, and $\rhd''$ is the profile in $\mathbf{S}$ obtained from $\rhd$ by replacing $\rhd_a$ by $\rhd''_a$.*

Proof. Assume that $\rhd'_a \colon d_1, d_2, \ldots, d_{m+1}$ and $d_k = d$. Then, we define a strict total order $\rhd^{\circ}_a$ in S_a by

$$\rhd^{\circ}_a \colon d_1, d_2, \ldots, d_{k-1}, d, a, \ldots.$$

Lemma 3.2 implies that $\ell_{\rhd'}(c) = \ell_{\rhd^{\circ}}(c)$ for every college c in C and $\mu_{\rhd'}(a) = \mu_{\rhd^{\circ}}(a) = d$. Thus, Lemma 3.3 implies that $\ell_{\rhd^{\circ}}(c) \geq \ell_{\rhd''}(c)$ for every college c in C. Thus, since $d \in F_{a,\rhd'}(\ell_{\rhd'})$, we have $d \in F_{a,\rhd''}(\ell_{\rhd''})$, which implies that $f_{a,\rhd''}(\ell_{\rhd''}) = d$. This completes the proof. □

Lemma 3.4 naturally leads to the following algorithm for Cheating Score-Limit Algorithm, called **Algorithm CSLA**.

Step 1: Compute a matching $\mu_{\succ}$ by using the BK-algorithm.

Step 2: Set $C' := \{c \in C \mid c \succ_a \mu_{\succ}(a)\}$.

Step 3: If $C' = \emptyset$, then output **null** and halt (i.e., there exists no cheating strategy of a). Otherwise, set c to be the maximal college in C' with respect to $\succ$, and then do the following steps.

- **(a)** Compute a matching $\mu_{\rhd}$, where we define a strict total order $\rhd_a$ in S_a by $\rhd_a \colon c, a, \ldots$, and $\rhd$ is the profile in $\mathbf{S}$ obtained from $\succ$ by replacing $\succ_a$ by $\rhd_a$.
- **(b)** If $\mu_{\rhd} = c$, then output $\rhd_a$ and halt. Otherwise, set $C' := C' \setminus \{c\}$, and then go back to the beginning of **Step 3**.

It is not difficult to see that **Algorithm CSLA** is a polynomial-time algorithm. The following main result of this section follows from Lemma 3.4.

Theorem 3.5. Algorithm CSLA *can correctly solve* CHEATING SCORE-LIMIT ALGORITHM.

4. Finding a Nash Equilibrium

In this section, we consider the following (strategic form) game. The set of players is $[n]$. For each applicant i in $[n]$, the set of strategies of an applicant i in $[n]$ is S_i. A profile $\rhd = (\rhd_1, \rhd_2, \ldots, \rhd_n)$ in $\mathbf{S}$ is called a *Nash equilibrium* [18], if for every applicant i in $[n]$ and every strict total order $\rhd'_i$ in S_i, we have $\mu_\rhd(i) \succ_i \mu_{\rhd'}(i)$ or $\mu_\rhd(i) = \mu_{\rhd'}(i)$, where $\rhd'$ is the profile in $\mathbf{S}$ obtained from $\rhd$ by replacing $\rhd_i$ by $\rhd'_i$. We will prove that a Nash equilibrium always exist, and we can find a Nash equilibrium in polynomial time.

A Nash equilibrium of the above game is closely related to weakly stable matching [19] defined as follows. A matching μ such that $\mu(i) \succ_i i$ or $\mu(i) = i$ for every applicant i in $[n]$ is called a *weakly stable matching*, if for every pair (i, c) in $[n] \times C$ such that $\mu(i) \neq c$ and $c \succ_i i$, at least one of the following conditions holds.

1. $\mu(i) \succ_i c$.
2. $|\mu^{-1}(c)| = q(c)$ and $s_j(c) \geq s_i(c)$ for every applicant j in $\mu^{-1}(c)$.

Theorem 4.1 (Irving [19])**.** *There always exists a weakly stable matching.*

In addition, it is known [19] that we can find a weakly stable matching in polynomial time.

Lemma 4.2. *Assume that we are given a weakly stable matching μ. For each applicant i in $[n]$, we define a strategy $\rhd_i$ as follows.*

$$\begin{cases} \rhd_i \colon \mu(i), i, \ldots & \text{if } \mu(i) \neq i \\ \rhd_i \colon i, \ldots & \text{if } \mu(i) = i. \end{cases} \tag{1}$$

Then, the profile $\rhd = (\rhd_1, \rhd_2, \ldots, \rhd_n)$ is a Nash equilibrium.

Proof. It is not difficult to see that $\mu_\rhd(i) = \mu(i)$ for every applicant i in $[n]$. Let us fix an applicant i in $[n]$. We prove that for any college c in C such that $c \succ_i \mu(i)$, there does not exists a strict total order $\rhd'_i$ in S_i such that $\mu_{\rhd'}(i) = c$, where $\rhd'$ is the profile in $\mathbf{S}$ obtained from $\rhd$ by replacing

$\rhd_i$ by $\rhd'_i$. Let us fix a college c in C such that $c \succ_i \mu(i)$. Define a strict total order $\rhd'_i$ by $\rhd'_i$: $c, i, \ldots$, and $\rhd'$ as the profile in $\mathbf{S}$ obtained from $\rhd$ by replacing $\rhd_i$ by $\rhd'_i$. Lemma 3.4 implies that if $\mu_{\rhd'}(i) \neq c$, then the proof is done. For this, it suffices to prove that $c \notin F_{i,\rhd'}(\ell_{\rhd'})$. In order to prove this, we prove that $s_i(c) < \ell_{\rhd'}(c)$. Since μ is a weakly stable matching, we have $|\mu^{-1}(c)| = q(c)$ and $s_j(c) \geq s_i(c)$ for every applicant j in $\mu^{-1}(c)$. Notice that since $c \succ_i \mu(i)$, we have $i \notin \mu^{-1}(c)$. If $s_i(c) \geq \ell_{\rhd'}(c)$, then (1) implies that $G_{c,\rhd'}(\ell_{\rhd'}) = \mu^{-1}(c) \cup \{i\}$, i.e., $|G_{c,\rhd'}(\ell_{\rhd'})| > q(c)$. This contradicts the fact that $\ell_{\rhd'}$ is H-feasible, which completes the proof. □

Theorem 4.3. *There always exists a Nash equilibrium.*

Proof. This theorem follows from Theorem 4.1 and Lemma 4.2. □

Furthermore, since we can find a weakly stable matching in polynomial time [19], we can find a Nash equilibrium in polynomial time.

Lemma 4.2 implies that for every weakly stable matching μ, there exists a Nash equilibrium $\rhd$ such that $\mu_{\rhd} = \mu$. However, the other direction does not necessarily hold.

Theorem 4.4. *In some instance, there exists a Nash equilibrium $\rhd$ such that $\mu_{\rhd}$ is not a weakly stable matching.*

Proof. We give a concrete example for proving this theorem. The set of applicants is $\{1, 2\}$, and the set of colleges is $\{c_1, c_2\}$. Define the total orders $\succ_1, \succ_2$ by $\succ_1$: $c_1, c_2, 1$ and $\succ_2$: $c_1, c_2, 2$. Define $q(c_1) = 1$ and $q(c_2) = 2$. In addition, we assume that $s_1(c_1) = s_2(c_1)$ and $s_1(c_2) = s_2(c_2)$. Then, $\mu_{\succ}(1) = \mu_{\succ}(2) = c_2$. It is not difficult to see that $\succ$ is a Nash equilibrium. However, $\mu_{\succ}$ is not a weakly stable matching. This completes the proof. □

Acknowledgements. This research was supported by JST, PRESTO.

[1] D. Gale, L. S. Shapley, College admissions and the stability of marriage, The American Mathematical Monthly 69 (1) (1962) 9–15.

[2] A. E. Roth, The economics of matching: Stability and incentives, Mathematics of Operations Research 7 (1982) 617–628.

[3] P. Biró, S. Kiselgof, College admissions with stable score-limits, Central European Journal of Operations Research 23 (2015) 727–741.

[4] T. Inoshita, R. W. Irving, K. Iwama, S. Miyazaki, T. Nagase, Improving man-optimal stable matchings by minimum change of preference lists, Algorithms 6 (2) (2013) 371–382.

[5] C.-P. Teo, J. Sethuraman, W.-P. Tan, Gale–Shapley stable marriage problem revisited: Strategic issues and applications, Management Science 47 (9) (2001) 1252–1267.

[6] H. Aziz, H. G. Seeding, J. K. von Wedel, On the susceptibility of the deferred acceptance algorithm, in: Proceedings of the fourteenth International Conference on Autonomous Agents and Multiagent Systems, 2015, pp. 939–947.

[7] H. Kobayashi, T. Matsui, Successful manipulation in stable marriage model with complete preference lists, IEICE Transactions on Information and Systems E92-D (2009) 116–119.

[8] H. Kobayashi, T. Matsui, Cheating strategies for the Gale–Shapley algorithm, Algorithmica 58 (2010) 151–169.

[9] N. Sukegawa, Y. Yamamoto, Preference profiles determining the proposals in the Gale–Shapley algorithm for stable matching problems, Japan Journal of Industrial and Applied Mathematics 29 (3) (2012) 547–560.

[10] C.-C. Huang, Cheating by men in the Gale–Shapley stable matching algorithm, in: Proceedings of the 14th Annual European Symposium on Algorithms, Vol. 4168 of Lecture Notes in Computer Science, 2006, pp. 418–431.

[11] C.-C. Huang, Cheating to get better roommates in a random stable matching, in: Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science, Vol. 4393 of Lecture Notes in Computer Science, 2007, pp. 453–464.

[12] H. Aziz, S. Gaspers, S. Mackenzie, N. Mattei, N. Narodytska, T. Walsh, Manipulating the probabilistic serial rule, in: Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems, 2015, pp. 1451–1459.

[13] M. Nasre, Popular matchings: Structure and strategic issues, SIAM Journal on Discrete Mathematics 28 (3) (2014) 1423–1448.

[14] M. S. Pini, F. Rossi, K. B. Venable, T. Walsh, Manipulation complexity and gender neutrality in stable marriage procedures, Autonomous Agents and Multi-Agent Systems 22 (1) (2011) 183–199.

[15] T. Matsui, Algorithmic aspects of equilibria of stable marriage model with complete preference lists, in: Operations Research Proceedings 2010, 2010, pp. 47–52.

[16] H. Aziz, S. Gaspers, S. Mackenzie, N. Mattei, N. Narodytska, T. Walsh, Equilibria under the probabilistic serial rule, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 1105–1112.

[17] T. Fleiner, Z. Jankó, Choice function-based two-sided markets: Stability, lattice property, path independence and algorithms, Algorithms 7 (2014) 32–59.

[18] J. Nash, Equilibrium points in n-person games, Proceedings of the National Academy of Science 36 (1) (1950) 48–49.

[19] R. W. Irving, Stable marriage and indifference, Discrete Applied Mathematics 48 (3) (1994) 261–272.