



Strategic issues in college admissions with score-limits



Naoyuki Kamiyama*

*Institute of Mathematics for Industry, Kyushu University, Japan
JST, PRESTO, Japan*

ARTICLE INFO

Article history:

Received 17 June 2016
Received in revised form
5 January 2017
Accepted 5 January 2017
Available online 11 January 2017

Keywords:

College admission
Score-limit
Weakly stable matching

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.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

The stable matching model introduced by Gale and Shapley [6] is one of the most important matching models from both the theoretical and practical viewpoints. Gale and Shapley [6] 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 [6] is the strategy-proofness for the proposing side [17]. 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 [4] 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.

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 [4]. In this problem, we are given some applicant. Then, the goal is to decide whether this applicant can cheat the algorithm of [4] 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 [9,19,3,11,12,18,7], the authors considered cheating strategies for the Gale–Shapley algorithm in the (classical) stable matching problem. Huang [8] considered a cheating strategy in the stable roommate problem. In [2], the authors considered a cheating strategy for the probabilistic serial rule. Nasre [15] considered a cheating strategy in the popular matching problem. Pini, Rossi, Venable, and Walsh [16] proposed a mechanism such that the problem of finding a cheating strategy for this mechanism is **NP-hard**. Matsui [13] considered a game related to cheating strategies for the Gale–Shapley algorithm. In [1], the authors considered a game related to cheating strategies for the probabilistic serial rule.

It should be noted that Fleiner and Jankó [5] proposed a choice function-based approach for the model proposed by Biró and Kiselgof [4].

* Correspondence to: Institute of Mathematics for Industry, Kyushu University, Japan.

E-mail address: kamiyama@imi.kyushu-u.ac.jp.

2. Preliminaries

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

In college admissions with score-limits introduced by Biró and Kiselgof [4], we are given a set $[n] = \{1, 2, \dots, n\}$ 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 prefers c_1 to c_2 . Define $\succ := (\succ_1, \succ_2, \dots, \succ_n)$. Furthermore, we are given a capacity function $q: C \rightarrow \mathbb{Z}_+$. 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 $\mathbf{S} := S_1 \times S_2 \times \dots \times S_n$. 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 $\triangleright = (\triangleright_1, \triangleright_2, \dots, \triangleright_n)$ in \mathbf{S} . For each score-limit ℓ and each applicant i in $[n]$, we define $F_{i,\triangleright}(\ell)$ as the set of colleges c in C such that $c \triangleright_i i$ and $s_i(c) \geq \ell(c)$. In addition, for each score-limit ℓ and each applicant i in $[n]$ such that $F_{i,\triangleright}(\ell) \neq \emptyset$, we define $f_{i,\triangleright}(\ell)$ as the maximal college in $F_{i,\triangleright}(\ell)$ with respect to \triangleright_i . For each score-limit ℓ and each applicant i in $[n]$ such that $F_{i,\triangleright}(\ell) = \emptyset$, we define $f_{i,\triangleright}(\ell) := i$. For each score-limit ℓ and each college c in C , we define $G_{c,\triangleright}(\ell)$ as the set of applicants i in $[n]$ such that $f_{i,\triangleright}(\ell) = c$. 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 \triangleright , if for every college c in C , $|G_{c,\triangleright}(\ell)| \leq q(c)$. In addition, an H-feasible score-limit ℓ with respect to \triangleright is called an *H-stable score-limit* with respect to \triangleright , 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 \triangleright .

This concept is motivated by a real system used in Hungary. (Biró and Kiselgof [4] introduced another stability concept called the *L-stability*. In this paper, we do not consider this stability concept.) Biró and Kiselgof [4] 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 [4] for finding an H-stable score-limit. We call this algorithm the *BK-algorithm*. (Precisely speaking, Biró and Kiselgof [4] 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 $\triangleright = (\triangleright_1, \triangleright_2, \dots, \triangleright_n)$ in \mathbf{S} . For computing an H-stable score-limit with respect to \triangleright , we set $\triangleright := \triangleright$.

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 $c \triangleright_i i$. 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 \bar{c}_t in $L_t(i_t)$ with respect to \triangleright_{i_t} . Furthermore, set π_t to be the same matching as σ_t except that $\pi_t(i_t) = \bar{c}_t$.

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

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

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

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

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

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

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 $d_1 \triangleright_i d_2$). It is known [4, Theorem 3.1] that an output ℓ of the BK-algorithm with an input profile \triangleright in \mathbf{S} is an H-stable score-limit with respect to \triangleright . The following property of this algorithm is known.

Theorem 2.1 (Biró and Kiselgof [4, Theorem 4.1]). *Assume that we are given an output ℓ of the BK-algorithm with an input profile \triangleright in \mathbf{S} . Then, for every H-stable score-limit ℓ' with respect to \triangleright and every college c in C , we have $\ell(c) \leq \ell'(c)$.*

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 [4], it immediately follows from Theorem 2.1.

Corollary 2.2. *An output of the BK-algorithm with an input profile \triangleright 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 $\ell_1(c) \leq \ell_2(c)$ and $\ell_2(c) \leq \ell_1(c)$, i.e., $\ell_1(c) = \ell_2(c)$. This completes the proof. \square

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

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 \triangleright_i in S_i is called a *cheating strategy* of i , if $\mu_\triangleright(i) \succ_i \mu_\triangleright(i)$ holds, where \triangleright is the profile in \mathbf{S} obtained from \succ by replacing \succ_i by \triangleright_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.

Download English Version:

<https://daneshyari.com/en/article/5128428>

Download Persian Version:

<https://daneshyari.com/article/5128428>

[Daneshyari.com](https://daneshyari.com)