Fairness Analysis of Traditional and Internet Supported Mass Enrollment to High Schools

Andrzej P. Urbański
Institute of Computer Science,
Department of Computer Science and Management,
Poznań University of Technology, ul.Piotrowo 3a, 60-965 Poznań, Poland,
andrzej.urbanski@cs.put.poznan.pl

Abstract

Fairness in educational enrollment means that better previous educational results are preferred. In this paper we show that in the case of mass enrollment it is difficult to make it entirely fair with traditionally distributed decision making. This has led us to a client-server enrollment system working on the Internet.

Keywords: algorithms, combinatorial problems, e-Government, e-Education, educational enrollment, enrollment fairness

1. Introduction

Fairness in educational enrollment means that a student with better previous educational results is preferred to a worse student in every school they apply to.

However, in this paper we show that in the case of mass enrollment it is difficult to make it entirely fair. This observation made in practice in 2002 (see sec. 3.1) has lead us to the design and implementation of a client-server enrollment system working over the Internet (see sec.3.2). In this paper we make an attempt to formally prove that computer support is necessary to achieve fair mass enrollment (see sect.3.3). In sec.4 we formally define enrollment and its fairness. In sec.5 we formally criticize mass enrollment performed with the use of traditional methods, while in sec.6 we introduce algorithms devoted to computerized mass enrollment coordination and prove their correctness.

2. Basic background

In Poland we have the following types of schools:

- primary school (6 years)
- junior high school (3 years) – called "gimnazjum"
- high school or vocational school (usually 3 years).

In this paper we focus on high school or vocational school enrollment, which in Poland has a unified framework. For this reason, we will use the term 'schools' to denote both high schools and vocational schools.

Moreover, we will use the term 'students' to denote schoolchildren who graduated from junior high school and apply to a high school or a vocational school.

3. Motivation for this work

3.1. Dramatic event as a motivation

Since the early seventies of the previous century, Polish schools have been enrolling new students on the basis of their school certificates only. Each school can have slightly different rules to compute the number of points from the grades on the certificates. From among all applying candidates each school admits no more students than the number of free places and chooses the candidates who have the highest number of points.

In 2002 the Polish Department of Education changed the rules of school enrollment by allowing the candidates to apply to an unlimited number of schools at the same time (before that the applicant could choose just one school). In that year it led to total chaos in the enrollment process in many schools in Poland, especially in big cities, where thousands of students applied to hundreds of schools and many of them tried to enroll to as many as ten schools simultaneously. The best students were admitted to every school they applied to and consequently blocked places, which prevented the rest from being accepted at all. It took long weeks full of frustration before they finally found a school while better students gradually unblocked places by deciding which school to choose.

3.2. Our first solution at a glance

This event has made us realize that a centralized information processing system for all enrolling schools (or at least all the schools from the whole metropolitan area) would solve the problem.

Contrary to the chaos in 2002, the subsequent enrollment in Poznań in 2003 was a great organizational success. Students could submit their certificates and preferences until Friday afternoon and received the final results the following week - on Tuesday morning. On Sunday before the Tuesday a super-committee of all school principals worked together and modified their school's offers in order to improve compliance with students' preferences.

As a result, each student was admitted to one school at most. More than 90% of students found schools in this stage of enrollment. The rest did not because they had provided too short preference lists.

3.3. Need for a theoretical justification

Our enrollment coordination algorithms and systems are mostly viewed as a solution to a specialized problem in a case-specific organizational environment. However, we believe that it can have several more global applications.

For this reason we decided to focus on objective formal methods, criticize traditional methods used in mass enrollment and prove that computer-oriented algorithms overcome the difficulties which arise while using the traditional methods.

4. The enrollment problem and its fairness

In order to define the enrollment problem, we assume that each candidate provides an ordered list of schools starting from the most desired one.

Definition 1. The enrollment problem.

Input

p – preferences; $p_{i,j}$=k means that k-th school was ranked j-th by candidate i

q – quality of a student; $q_{i,k}$=v means that student i in the opinion of school number k is valued at v (in this paper we assume that a different quality value is assigned to each student); if the value does not depend on the school (each school has the same opinion) we omit k

s – maximum school capacity; s_k=l means that school k can take in not more than l new students

n – number of students

m – number of schools

Output

r – enrollment result; r_i=j means that student i is assigned to a school number j on his preference list; 0 means that there is no school for student i

Problem

Find integer r_i for i=1,2,…n such that:

1. $\underset{i\in<1,n>}{\forall} 0 \le r_i \le m$
2. $\underset{k\in<1,m>}{\forall} s_k \ge \left|\left\{p_{i,r_j} = k \mid i \in <1,n>\right\}\right|$
3. If student i is admitted to school r_i and student j is not, despite his or her preferences, student i should have a higher value of the quality function than j. $\underset{i,j\in<1,n>}{\forall} \underset{k<r_j}{\exists} \left(p_{j,k} = p_{i,r_i} \Rightarrow q_i > q_j\right)$

Point (3) in **Definition 1** describes a fairness rule, which is intuitive and could be easily checked on publicly available lists of admitted candidates.

5. Traditional distributed enrollment

Traditionally, each school performs the enrollment independently from the others, and the process is not coordinated by anyone. We distinguish two variants of such enrollment: with strongly limited and unlimited number of schools the student can apply to at the same time.

5.1. Limited registrations

The most common limit set by different countries throughout the years is between one and three.

ALGORITHM 1

Each school performs its individual enrollment process at the same time. No school can recruit more students than the number of available places.

THEOREM 1

ALGORITHM 1 is not fair.

PROOF

We can easily imagine a situation in which a student with fairly good educational results applies to very popular schools and is not admitted anywhere in the first stage of the enrollment process because of too many better candidates. Had he or she chosen a less besieged school, he or she would have been admitted straight away. The algorithm is not fair because the results of enrollment depend on the student's ability to predict school's popularity in a given year.

5.2. Unlimited registrations

The great inconvenience and disadvantage of this method lies in the fact that the organizers of enrollment have to force students to make hasty decisions. Moreover, in practice the process appears to be very long and inefficient. However, in this paper we discuss its fairness only.

ALGORITHM 2

Each school starts its enrollment at the same time.

After the applications' submission deadline, students who are potentially admitted are required to confirm their enrollment within a few days or else they are deleted from the list. Candidates who were rejected at first are now added to the list of admitted students to fill in the vacancies but again they are given some time to confirm their enrollment. The process continues until the lists contain confirmed enrollments only. What is more, each student can confirm his or her enrollment in one school only, which means that the decision has to be final.

THEOREM 2

ALGORITHM 2 is not fair.

PROOF

Not fair, because it requires students to make risky decisions in a short period of time. Good but nervous students will probably confirm the first offer, although they had a chance to get to a better school, provided that they waited patiently until better students confirmed their enrollments elsewhere and unblocked places.

6. Centralized coordination of mass enrollment

In the previous section we have proved that traditional distributed mass enrollment is not fair. When we discovered this in practice (see sect.3) we introduced computerized central mass enrollment coordination, the fairness of which we will prove in this section. We will separately discuss several variants applied in different circumstances.

6.1. Coordination algorithm for the unified criteria problem

The unified criteria problem is the simplest one, but still useful in practice (see [3] for examples). It assumes that all schools use the same student ranking methods and therefore it is enough to compute one quality value for each student.

ALGORITHM 3

Sort the student records by their educational results starting from the best student. Take the first student from the list and let him choose any school which still has vacancies. Remove the student from the list, and repeat the procedure for the next student.

THEOREM 3

The sorting algorithm is fair for the unified criteria problem

PROOF

Fairness rule from **Definition 1** will be fulfilled, because only a candidate with better results can steal another candidate's dream place in a given school.

6.2. Cloned applications algorithm for different ranking criteria in every school

The following algorithm was first implemented on a notebook in 2002 and used to build the first enrollment system in 2003 (see sect.3). It is almost the same as the traditional ALGORITHM 2, but all decisions are made "inside the computer" (student preference lists are used instead of asking students for confirmation in the course of the enrollment process), which makes work much faster, and the results – fair because it does not require risky and hasty decisions from students.

ALGORITHM 4

Step 1: For each school there is a queue of candidates:

$$\underset{k\in<1,m>}{\forall} C_k = \{i \in <1,n>| \ p_{i,j} = k\}$$

$$\underset{k\in<1,m>,i,j\in<1,n>}{\forall} |C_k| \geq i > j \Rightarrow q_{c_k^i,k} < q_{c_k^j,k}$$

Each student is cloned to appear in the queue of every school he included in his or her preference list.

Step 2: Queue L of potentially admitted students is created and initially assigned: $L := \left\{ i \in <1,n>| \underset{l\in<1,s_k>}{\exists} c_k^l = i \right\}$

Step 3: Subsequent student number "i" is taken from L and removed from it.

Step 4: We are looking for a minimum value of "mj" where

$$p_{i,mj} = k \wedge \underset{l\in<1,|C_k|>}{\exists} (c_k^l = i \wedge l \leq s_k)$$

Step 5: For all j greater than mj we remove "i" from a corresponding C_k

Step 6: Add to L students who moved above the admittance border in any C_k

Step 7: If L is not empty then go to Step3.

Step 8: All C_k contain a list of admitted students (on positions from 1 to s_k).

THEOREM 4

Cloned applications algorithm is fair.

PROOF

1. The property of fairness is guaranteed in Step1, which enforces the right order of students in each school. This order is not changed in subsequent steps and only superfluous entities are removed in Step5.
2. Algorithm finds a solution in a finite number of steps, because each entry from C is added to and removed from L only once.

However, it was found that this algorithm behaves strangely in some cases and thus is not entirely fair [3].

DEF. 2

For at least two students i and j of which one has $p_{i,1}=1$ and another $p_{j,1}=2$. If a given algorithm assigns them in a reversed order i.e. $r_i=2$ and $r_j=1$ we call it a paradox of crossed preferences.

THEOREM 5

ALGORITHM 4 allows for the occurrence of the paradox of crossed preferences.

PROOF

Assume we have only two students and two schools with only one free place in each one. Preferences are as follows: $p_{i,1}=1$, $p_{j,1}=2$, $p_{i,2}=2$, $p_{j,2}=1$; and quality factors: $q_{i,1}=100$, $q_{j,1}=200$, $q_{i,2}=200$, $q_{j,2}=100$. It is easy to notice that ALGORITHM 4 will find result $r_i=2$ and $r_j=1$.

6.3. Moving applications algorithm for different ranking criteria in every school

When one thinks about a new generation of enrollment systems one thinks about something extremely flexible. When we analyzed the enrollment problem for the first time, we thought of agent-based systems [1,2]. However, existing environments are not ready to be used in such responsible applications. Today's agent-based environments do not implement persistent systems [2]. Nevertheless, our experiments on agent-based systems resulted in creating a simple and lucid algorithm, which could also be implemented traditionally:

ALGORITHM 5

proc AddToSchool(t:Student,k:School);

For a school k queue of candidates is updated:

$$C_k' = \{i \in C_k \cup \{t\}\}$$

$$\underset{k\in<1,m>,i,j\in<1,n>}{\forall} \left|C_k'\right| \geq i > j \Rightarrow q_{c'^i_k,k} < q_{c'^j_k,k}$$

if $|C_k'| > S_k$

then begin

AddToSchool("last student from C_k", "next school on his/her preference list");

RemoveLastFrom C'_k;

end;

end (* AddToSchool *);

```
begin
   for all t from all students
     AddToSchool(t,pt,1);
end;
```

Of course, this algorithm is fair:

THEOREM 6

Moving applications algorithm is fair

PROOF

ALGORITHM 5 works on the basis of a rule “better removes worse”. Each student tries to apply to the first school on their preference list. If there are already no free places there, the student is moved to their next preference, but even if the algorithm initially assigns a student to the first or next school on their preference list, the place is not guaranteed and can change in subsequent steps. This is a direct application of the fair enrollment rule (**Definition 1**).

What is more interesting is that it does not allow for the paradox of crossed preferences:

THEOREM 7

ALGORITHM 5 does not allow for the paradox of crossed preferences.

PROOF

Let us assume we have only two students and two schools with only one free place in every school. Preferences are as follows: $p_{i,1}=1$, $p_{j,1}=2$, $p_{i,2}=2$, $p_{j,2}=1$; and quality factors: $q_{i,1}=100$, $q_{j,1}=200$, $q_{i,2}=200$, $q_{j,2}=100$. It is easy to notice that ALGORITHM 5 will find the correct result: $r_i=1$ and $r_j=2$.

7. Conclusions

In our first paper [3] we have sketched a much wider area of applications for enrollment systems including internal (already implemented and used in practice at Poznań University of Technology since 1998) and external enrollment at Universities. Out next step should be to refine our algorithms or even find more sophisticated ones, all to make the enrollment process as efficient and fair as possible.

8. Acknowledgements

This work was partially supported by grant DS-91-425 from Poznań University of Technology.

The author wishes to thank Aleksandra Wojnowska for her comments on an earlier version of this paper, which allowed me to make significant improvements in it.

9. References

1. Mazurek C., Stroiński M., Urbański A.P., "A Client-Server System for Mass Recruitment to Secondary Schools in a Metropolitan Area", eAdoption and Knowledge Economy(eds. P.Cunningham and M.Cunningham), Vienna, IOS Press, 2004, pp.1801-1808.
2. Płóciennik M., "Solving recruitment problems in agent-based environment", M.Sc. Thesis, Dept. of Computer Science and Management, Poznań University of technology, September, 2004(in Polish).
3. Urbański A.P., Nawrocki J., „Algorithms for fair and efficient educational recruitment", Conference on EDI-EC, Łódź, Acta Universitas Lodziensis, FOLIA OECONOMICA 167, 2003, pp.429-442 (in Polish).