Towards a Fair and Comprehensible Course Allocation System

Term Paper, ECO426: Market Design

Department of Economics, University of Toronto

Tianyu Du*

Instructor: Professor Anne-Katrin Roesler

April 10, 2020

1 Introduction

The main goal of this paper is to propose a course allocation system for universities. Currently, most universities are allocating courses on a first-come-first-serve basis with enrolment priorities. A student typically has priorities for courses in her program requirement. An university determines two start dates for course enrolment: one priority enrolment date, and one regular enrolment date. After the priority enrolment date, students are eligible to enrol in courses they have priority access to. After the regular enrolment date, all students are eligible to enrol in all courses. Moreover, the university gave students different priority enrolment date based on their years of study, students of higher years have earlier priority enrolment dates. In contrast, the regular enrolment dates are the same for all students.

Ideally, a course allocation system has two merit properties on the student side: **fairness** and **comprehensibility**. A fair allocation mechanism should ensure students who need the course most have a higher chance of getting enrolled. For instance, given the limited seats in a course, someone in her last semester and needs this course for degree completion should have a higher chance of enrolling in this course than another first-year student. A comprehensible allocation system needs to be understandable and minimizes students' effort to use this system. For instance, asking students to submit their preference over all possible combinations of courses involves prohibitively high effort: even choosing 5 courses from 10 courses requires a student to order 252 course bundles.

From the technical point of view, letting all students enrol courses via an online portal (e.g., acorn) immediately after the start date (called concurrent requests) induces great pressure on the school's infrastructure (e.g., web servers). Due to the high volume of concurrent requests from the student within the first few minutes after the starting date, the server could become irresponsible and the student portal goes down. Moreover, students accessing the portal from another country would have significant disadvantages due to

*tianyu.du@mail.utoronto.ca

internet lags and time zone issues. The current allocation system is highly time-sensitive for students and students tend to be nervous before the course enrolment date. To alleviate the pressure on both the server and students, the allocation outcome of an effective system should not depend on the exact time when a student submits the enrolment request.

Lastly, while improving the current system, the new system should only involve the minimized amount of modification. The new system is ideally similar to the original system as much as possible so that students and instructors can adapt to the new system quickly.

This paper aims to introduce a fair and comprehensible course allocation system for students. At the same time, the proposed system eases the pressure on students and the school's infrastructure.

2 The Course Allocation Problem

2.1 Agents

Agents in the course allocation problem are students and the school/course-offering department. The school is effectively an auctioneer allocating commodities (courses) to buyers (students). For the fairness issue, no monetary transaction should be involved in this allocation system and the final allocation should be independent of students' financial background.

For the school side, there are multiple types of commodities (courses), this paper uses c to denote courses and C to denote the total number of course. For each course c, there is a limited number of seats, denoted as q_c. In very rare cases, the course-offering department would expand the class size, however, an effective course allocation system should treat q_c as constant constraints. For simplicity, we assume that there is only one section of one course, that is, all q_c seats of course c are homogenous. Therefore, there are $\sum_{c=1}^{C} q_c$ items in total to be sold.

Let S denote the total number of students, for a student s, not all courses are acceptable for this student, she would not consider most courses outside her program and those courses with conflicts (if there is a timetable conflict or the student does not have the prerequisite). This paper assumes each student has a strict and responsive preference over all subsets of her acceptable courses. In particular, she is capable of ordering any set of courses from the most preferred one to the least preferred one and then unacceptable courses.

2.2 Allocating Courses

A typical undergraduate student at the University of Toronto enrolls in four to six (the maximum) courses each semester among which around four courses are required by their program of study and one is for the breadth requirement (i.e., elective courses). Traditionally, students can enroll up to five courses before the regular enrolment date and choose the sixth course afterward.

In huge universities like the University of Toronto, the demands for many courses are much higher than the number of available spots. Every year, many students fail to enroll in courses in their program entry/completion requirements and these students' academic plans are messed up. The proposed system aims to minimize the number of students suffering from this issue.

3 The Allocation Mechanism

The introduced allocation system (the 'system') consists of two stages. The first stage (priority enrolment round) allows students to enroll up to five courses in this round. In the second stage, students can choose to enroll in the sixth course (regular enrolment round).

3.1 Priority enrolment Round: Bidding Stage

Prior to the course enrolment, the system gives each student a fixed number of **tokens**. The token has no monetary value and cannot be transferred. Moreover, tokens are designed to allocate courses during the current priority enrolment period only. Therefore, these tokens are only usable for the current semester and expire immediately after the regular enrolment date. Students with higher years of study receive a higher amount of tokens, and these students in their last year receive substantially more tokens.

At the bidding stage, each student firstly chooses *up to* ten courses and order them according to her strict preference. The system does not allow tie at this stage, students are required to provide a strict order over these selected courses. Then the student allocates her tokens among these courses to indicate her bids on these courses. Note that the student is allowed to assign the same amount of tokens on more than one course, however, the strict order over these courses breaks any possible tie on the second step. For example, one student chooses three courses and orders them as ECO101 $\succ$ MAT137 $\succ$ MAT223. Then, she allocates 20, 10 and 10 tokens on these courses. Given the strict preference submitted, this bidding is effectively bidding 20, $10 + \varepsilon$ and $10 - \varepsilon$, and the tie in token allocation is broken. For any course c in the list of student s, we say that student s **requests** for course c.

3.2 Priority enrolment Round: Allocating Courses using Auctions

After all students submit their preference orders and biddings, for each student-course pair (s, c) (student s bids positive for course c), the system assigns a **fitness score** on this bid. Unlike the number of tokens assigned on c, which measures how much student s wants course c *subjectively*, the fitness score proxies how student s *objectively* needs course c.

For example, there is a first-year student in math major requesting for MAT137 (first-year calculus) and this course is required by the student's program of study and is the prerequisite of many subsequent courses (i.e., most second-year math courses). Therefore, the system will assign a high fitness score to this student-course pair. In contrast, when the course is purely elective (i.e., none of this student's programs requires this course) for this student, the fitness score is one. The eventual fitness score is determined by multiple factors and table 1 enumerates several factors affecting the student's fitness score.

Table 1: Rules on Determining the Fitness Score

Fitness Score	Condition
1.0	The baseline fitness score.
+0.5	The course is in the student's program requirement.
+0.5	The course is in the student's program requirement, the specified year of study matched (i.e., second-year course, etc).
+0.2	This course is offered by this student's departmental affinity, which is determined by this student's primary specialist program.
+1.0	This student is in her last year.
−0.5	This student is a part-time / non-degree student.
×0.0	This course is blocked for this student during the priority enrolment period.
×0.0	This student has taken this course before (i.e., repeated course)

Note that these above-mentioned factors are completely flexible and subject to change, the department may choose to add more rules to this system.

The bidding price, in terms of tokens, for each student-course pair is then multiplied by the corresponding fitness score, the product constitutes a **weighted bid** on this course by this student. Students' weighted bids on courses reflect both how they subjectively want the course and how they objectively need the course. The system is then allocating courses based on students' weighted bids.

The first round enrolment is determined by an iterative process consists of auctions. Note that all tokens expire after the current semester and tokens are not real expenses, therefore, there is no difference between using first-price and second-price auctions here.

First Round Allocation (**Step 0**) Let $n_s = \max\{5, \text{num of courses requested}\}$ denote the maximum number of courses student s may enrol in during the priority enrolment round. After the initialization step, each step $h \geq 1$ consists of three sub-steps.

Step h.1: For each course c with capacity q_c, the system assigns **candidacy of enrolment** for this course to the q_c students with the highest weighted bids, ties are broken randomly.

Step h.2: For now, each student has up to ten candidacies of enrolment, depending on her bids and popularities of courses. Let Φ_s denote the set of candidacies received by student s, and $|\Phi_s|$ is the number of candidacies received. For a student with $|\Phi_s| \leq 5$, she is then enrolled in all courses with candidacies and updated as $n_s = n_s - |\Phi_s|$. For a student with $|\Phi_s| > 5$, she is enrolled in the five courses she prefers the most according to her submitted preference list. For these students, n_s is set to zero and we say these students' requests are fulfilled.

Step h.3 Then, course capacities are updated to their original capacities minus current numbers of enrolments. Courses with $q_c = 0$ and students with $n_s = 0$ are removed from this problem. Then step $h.1$ to step $h.3$ run again in the next iteration. Note that preference lists do not change during the algorithm.

The algorithm stops when there is no more new candidacy assignment. After the algorithm stops, outcomes are announced to students and all tokens (spent or not) expire.

It is not necessarily that all students are fulfilled after this algorithm stops. This could happen to students who request courses with extremely imbalanced demand and supply but bid too few tokens. However, the new system could fulfill more students' requests than the original system.

3.3 Regular enrolment Round

The second round allocation is basically the same as before: all students log into the portal and enrol courses (or join the waitlist) online after the starting date, and courses are distributed on a first-come-first-serve basis. Students are allowed to enrol in up to five courses (including waitlist) before the regular enrolment date, after which students can enrol in the sixth course (including waitlist).

As mentioned before, a comprehensible allocation system minimizes student's time cost of using it. In the priority enrolment round, all students need to do is to provide a list of up to ten courses and assign tokens on those courses. The rest of the mechanism completely runs on the server-side.

Moreover, the proposed system helps relieve server-side traffic pressure as well. This system reduces the maximum of concurrent requests. In the original system, a wave of requests arrives shortly after the beginning of the priority enrolment period: students log into the online portal and request for enrolment immediately after the portal is open (about $S \times 5$ requests). In contrast, in the proposed system, students can submit their lists of requested courses and bids at any time before the deadline and the spike of internet traffic is smoothed out. There are still two but much smaller waves of requests in the new system. The first occurs one day after the matching results are announced, students enrolled in less than five courses may request to enrol in additional courses (much less than $S \times 2$ requests). As argued before, most of the students will have their requests fulfilled in the auction, only a few students need to submit enrolment requests on the portal after the auction. The second wave arrives after the regular enrolment date when students choose the sixth course (much less than S requests). The new system significantly reduces the number of concurrent requests to the school's server.

4 Digression: Dynamic Adjustment of Class Size

Even though course capacities are taken as constraints in this allocation system, departments can still add new sections to highly demanded courses. For example, the department may choose to expand the course capacity if there is a long waitlist for a course and the course is a strict program entry/completion requirement. This can be done better in the proposed system: the department does not have to wait until the enrolment period ends and modify class capacities according to waitlist sizes. After the system receives preference lists and bids from students, it can use the total number of tokens bid to each course as a proxy of the course's popularity and demand. Referring to historical data, the department can choose to modify the capacity of highly demanded courses even before the auction-based course allocation.

5 Concluding Remarks

This paper presents a new course allocation system based the current system used at the University of Toronto. The new system aims to minimize students' time cost of using it. Moreover, while allocating courses, the proposed system take both subjective and objective aspects into consideration using an auction-based system. Students who subjectively want to take the course and objectively need the course for their programs are more likely to get seats. Last but not least, the new system helps reduce the maximum concurrent requests to the school's server significantly and alleviates the pressures on infrastructures. However, there are still several improvements and extensions to the proposed system. Firstly, the course-offering department can refine fitness score rules in table 1 to better differentiate students and avoid ties in the auction. Secondly, applying the auction-based allocation system on all courses might be overshooting. The school can apply this system on only a portion of seats of the course: students in the program have guaranteed seats, and the rest of seats are distributed among students outside the program.