Submitted to *Operations Research*
manuscript 1

Improving the Chilean College Admissions System

Ignacio Rios
Graduate School of Business, Stanford University, iriosu@stanford.edu

Tomas Larroucau
Department of Economics, University of Pennsylvania

Giorgiogiulio Parra
Departamento de Evaluación, Medición y Registro Educacional (DEMRE), Santiago, Chile

Roberto Cominetti
Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Santiago, Chile

In this paper we present the design and implementation of a new system to solve the Chilean college admissions problem. We develop an algorithm that obtains all stable allocations when preferences are not strict and when all tied students in the last seat of a program (if any) must be allocated, and we used this algorithm to determine which mechanism was used to perform the allocation. In addition, we propose a new method to incorporate the affirmative action that is part of the system and correct the inefficiencies that arise from having double-assigned students. By unifying the regular admission with the affirmative action, we have improved the allocation of approximately 3% of students every year since 2016. From a theoretical standpoint, we introduce a new concept of stability and we show that some desired properties, such as strategy-proofness and monotonicity, cannot be guaranteed under flexible quotas. Nevertheless, we show that the mechanism is strategy-proof in the large, and therefore truthful reporting is approximately optimal.

Key words: college admissions, stable assignment, flexible quotas, non-strict preferences.

History: This paper was first submitted on XXX, 20XX and has been with the authors for X years for 0 revisions.

1. Introduction

Centralized admission systems have been increasingly used in recent years to carry out the assignment of students to schools and colleges. A variety of mechanisms have been studied, including the celebrated Deferred Acceptance (DA) algorithm (Gale and Shapley (1962)), the Immediate Acceptance (Boston) algorithm (Abdulkadiroglu et al. (2005) and Ergin and Somnez (2006)), the Top-Trading Cycles algorithm (Shapley and Scarf (1974)), among many others. An important part of the literature in market design has been devoted to characterize these mechanisms, mostly focusing on canonical examples that illustrate their properties. Another important body of the literature studies real life applications, by combining the aforementioned mechanisms with specific rules such as restrictions in the length of preferences, tie breaking rules, affirmative actions, among many others. In this paper we try to contribute to both by studying the Chilean college admissions problem.

A centralized mechanism to match students to programs[1] has been used in Chile since the late 1960's by the *Departamento de Evaluación, Medición y Registro Educacional* (DEMRE), the analogue of the American College Board. Every year more than 250,000 students participate in the system, which includes more than 1,400 programs in 33 universities. This system has two main[2] components: a regular admission track, where all students that graduated from high-school can participate; and an affirmative action policy, that aims to benefit underrepresented groups by offering them reserved seats and economic support. More specifically, to be considered for the reserve seats and the scholarship – which is called *"Beca de Excelencia Academica"*, or simply BEA – a student must belong to the top 10% of his class, must graduate from a public/voucher school and his family income must be among the lower four quintiles.

When the affirmative action was introduced in 2007, the procedure to match students to programs relied on a black-box software that could not be updated to incorporate this new feature. Hence, the authorities decided that the admission of BEA students would be run after the admission of Regular students. Since BEA students can apply to both regular and reserve seats[3], running the process sequentially introduces a series of inefficiencies. For instance, a BEA student can be assigned to two different programs, and the seat that this student decides not to take cannot be re-allocated to another student. Moreover, the double-assignment generates confusion among students and increases the chances of them enrolling in a less preferred program.

In this paper we provide a "reverse-engineering" approach to correct these inefficiencies. The reason why we start from the current system and we don't simply propose a re-design is that DEMRE wanted to keep the current rules and incorporate the reserved seats while keeping the system as close as possible to its current state. Hence, there were two main questions to be answered: (1) what was the mechanism that was currently being used, and (2) how could this mechanism be modified to unify the admissions for regular and reserve seats. To answer these questions and address the aforementioned inefficiencies, our first goal was to determine the mechanism "inside the black-box". Based on the rules of the system, we had enough evidence to think that the desired outcome was a stable matching. In addition, we realized that unlike other systems all students tied in the last seat had to be admitted, so quotas must be *flexible* in order to allocate them. With these features in mind, we implemented an algorithm based on Baïou and Balinski (2004) that obtains all stable allocations satisfying the rules of the system. By comparing the results of our algorithm with the actual assignment of past years we found that the mechanism used is a university-proposing deferred acceptance, with the special feature of flexible quotas to allocate all tied students in the last seat. Furthermore, we show that, unlike the classic DA, the Chilean mechanism is not strategy-proof nor monotone. Nevertheless, we show that strategic behavior is not a major concern since our mechanism is strategy-proof in the large.

After determining the algorithm that was being used, our next goal was to integrate both systems in order to maximize the utilization of vacancies. To solve this problem we introduce a new approach where each type of seat (regular or reserve) is assumed to belong to a different program with its own capacity and requirements, and students benefited by the affirmative action can apply to both.

This research is the outcome of an ongoing multiyear collaboration with DEMRE (2012-2017), aiming to improve the Chilean college admissions system. All the solutions described in this paper were adopted and implemented starting in 2014 with a pilot phase. In 2015 the system switched to a student-optimal mechanism with flexible quotas, and in 2016 the unified allocation was finally adopted. Based on simulations in 2014-2015 and actual data in 2016, we find that our implementation has improved the allocation of approximately 3% of the students participating each year. Furthermore, our "white box" implementation made the admission process fully transparent and reduced the execution time from over 5 hours to a couple of minutes. This improvement in transparency and performance has allowed the evaluation and introduction of different policies (e.g. the inclusion of the high-school class rank as admission factor; see Larroucau et al. (2015), among others) that otherwise could not have been included.

The reminder of the paper is organized as follows. Section 2 provides a background on the Chilean college admissions system and introduces the main features of the problem. In Section 3 we discuss the closest related literature to our setting. In Section 4 we develop a model that formalizes our problem, we describe our mechanisms and present their properties. We discuss the implementation in Section 5. Finally, we provide concluding remarks in Section 6.

2. The Chilean College Admissions System

In order to apply to any program in the 33 universities that are part of the centralized system run by CRUCH[4], each applicant undergoes a series of standardized tests (*Prueba de Selección Universitaria* or PSU). These tests include Math, Language, and a choice between Science or History, providing a score for each of them. The performance of students during high-school gives two additional scores, one obtained from the average grade along high-school (*Notas de Enseñanza Media* or NEM) and a second that depends on the relative position of the student among his/her cohort (*Ranking de Notas* or Rank)[5].

Once they have their scores, each student can apply to at most 10 different programs at no cost. These programs must be listed in strict order of preference. Each program defines a set of specific requirements that must be met by applicants to be acceptable, such as minimum application score or minimum tests scores. In addition, each program freely[6] defines the weights assigned to each score and also the vacancies offered for (i) the Regular process, where all students compete, and for (ii) the special admission track related to the affirmative action policy (BEA process/track).

Each program's preference list is defined by first filtering all applicants that do not meet the special requirements. Then, students are ordered in terms of their application scores, which are computed as the weighted sum of the applicants' scores and the weights pre-defined by each program. Note that two candidates can obtain the same application score, and therefore programs' preferences need not be strict.

Considering the preference lists of the applicants and programs, as well as the vacancies offered in both admission tracks, DEMRE runs an assignment algorithm to match students and programs. Specifically, the Regular process is first solved by considering all the applications and the regular vacancies. Once the Regular process is done, the BEA process is run considering only the applications of students shortlisted for the scholarship and those programs in which they were wait-listed in the Regular process. Thus, some students who are assigned a regular vacancy might later decline it if admitted to a preferred program through the BEA admission process. The declined positions can not be re-assigned, and therefore are lost[7].

DEMRE performs the matching for both processes using a black-box software for which no information is available regarding the specific algorithm used. Instead, the following description is provided[8]:

> *"SORTING OF APPLICANTS PER PROGRAM AND ELIMINATION OF MULTIPLE ALLOCATIONS:*
>
> *(a) Once the final application score is computed, candidates will be ordered in strict decreasing order based on their scores in each program.*
>
> *(b) Programs complete their vacancies starting with the applicant that is first in the list of candidates, and continue in order of precedence until seats are full.*
>
> *(c) If an applicant is selected in his first choice, then he is erased from the lists of his 2nd, 3th, 4th, until his last preference. If he is not selected in his first choice, he is wait-listed and moves on to compete for his 2nd preference. If he is selected in this preference, he is dropped from the list of his 3th to this 10th choice, and so on. In this way, it is possible that a student is selected in his 6th preference and wait-listed in his top five preferences; however, he will be dropped from the lists of his preferences 7th to 10th.*

(d) This procedure to select candidates is the result of an agreement between the universities of having a unified and integrated process, so that no student is admitted by more than one program. Nevertheless, a student can be wait-listed in more than one program if his score is not enough to be admitted.

(e) All candidates that apply and satisfy the requirements of the corresponding program and institution will be wait-listed.

THEREFORE, IT IS FUNDAMENTAL THAT APPLICANTS SELECT THEIR PROGRAMS IN THE SAME ORDER AS THEIR PREFERENCES".

This description suggests that the final allocation must be stable, in the sense that there is no pair student/program who simultaneously prefers to be matched together rather than to their matches in the proposed assignment. Indeed, as the results are public and students can easily check if their scores are higher than that of the last admitted in a program they prefer, legal problems may arise if the resulting matching is unstable. However, it is unclear from the description which specific stable assignment is implemented. Moreover, by analyzing the results of previous assignments we realized that, in case of a tie in the last seat of a program, the number of seats increased in order to include all tied applicants. This feature of the system was confirmed by DEMRE, and applies to both admission tracks (Regular and BEA). From now on we refer to this as *flexible quotas*.

Another important feature of the system is that the admission tracks are processed sequentially. In particular, DEMRE begins solving the Regular process by considering all students (Regular and BEA), applications, and the regular seats. Then, using the results of their black-box they update the applications of BEA students in order to consider only those preferences where they were wait-listed in the Regular process. Finally, considering only BEA students, their updated preferences and the reserve seats, DEMRE runs the same black-box to obtain the assignment for the BEA process. Notice that a BEA student can be double-assigned[9], and that BEA students always prefer their allocation in the BEA process. Nevertheless, DEMRE reports both allocations and allows double-assigned students to choose where to enroll. This introduces large inefficiencies, since the seats that are assigned but not taken by double-assigned students are not re-allocated, and therefore could lost if they are not re-allocated during the enrollment process[10].

3. Literature review

This paper is related to several strands of the literature. Regarding the mechanism, Biró and Kiselgof (2013) analyze the college admissions system in Hungary, where all students tied in the lowest rank group of a program are rejected if their admission would exceed the quota. This mechanism is opposed to the Chilean case, where the quota is increased just enough so that all students in the tie can be admitted. Biró and Kiselgof (2013) formalize these ideas by introducing the concepts of H-stability and L-stability, that correspond to the rules in Hungary and Chile respectively. They also provide a natural adaptation of the Deferred Acceptance algorithm to compute H-stable and L-stable based on ascending score limits, and provide an alternative proof of the manipulability of H-stable and L-stable mechanisms. In a recent paper, Kamiyama (2017) presents a polynomial algorithm to check whether a student can manipulate her preferences to obtain a better allocation.

Our paper is also related to the literature on affirmative action. Most of the research in this strand has focused on proposing mechanisms to solve the college admissions problem with diversity constraints and deriving properties such as stability, strategy proofness and Pareto optimality. From a theoretical perspective, Echenique and Yenmez (2012) point out that the main tension between diversity concerns and stability is the existence of complementarities, although the theory requires substitutability for colleges' choices. Abdulkadiroğlu (2007) explores the Deferred Acceptance algorithm under type-specific quotas, finding that the student-proposing DA is strategy proof for students if colleges' preferences satisfy responsiveness. Kojima (2012) shows that majority quotas may actually hurt minority students, which will be to the detriment of the aims of affirmative action policies. Consequently, Hafalir et al. (2013) propose the use of minority reserves to overcome this problem, showing that the deferred acceptance algorithm with minority reserves Pareto dominates the one with majority quotas. Ehlers et al. (2014) extend the previous model to account for multiple disjoint types, and propose extensions of the Deferred-Acceptance algorithm to incorporate soft and hard bounds. Other types of constraints are considered by Kamada and Kojima

(2015), who study problems with distributional constraints motivated by the Japanese Medical Residency. The authors propose a mechanism that respects these constraints while satisfying other desirable properties such as stability, efficiency and incentives.

Some authors have recently analyzed the impact of the order in which reserves are processed. Dur et al. (2016a) analyze the Boston's school system and show that the precedence order in which seats are filled has important quantitative effects on distributional objectives. This paper formalizes our idea that processing reserve seats in a lower precedence order benefits BEA students. In a follow-up paper, Dur et al. (2016b) characterize optimal policies when there are multiple reserve groups, and analyze their impact using Chicago's system data.

4. Model

The following framework is assumed hereafter. Consider two finite sets of agents: programs $C = \{c_1, \ldots, c_m\}$ and applicants $A = \{a_1, \ldots, a_n\}$. Let $V \subset C \times A$ be the set of *feasible pairs*, with $(c, a) \in V$ meaning that student a has submitted an application to program c and meets the specific requirements to be admissible in that program. A *feasible assignment* is any subset $\mu \subseteq V$. We denote by $\mu(a) = \{c \in C : (c, a) \in \mu\}$ the set of programs assigned to a and $\mu(c) = \{a \in A : (c, a) \in \mu\}$ the set of students assigned to program c. Each program c has a quota $q_c \in \mathbb{N}$ that limits the number of students that can be admitted. Moreover, program $c \in C$ ranks the applicants according to a *total pre-order* $\leq_c$, namely a transitive relation in which all pairs of students are comparable. The indifference $a \sim_c a'$ denotes as usual the fact that we simultaneously have $a \leq_c a'$ and $a' \leq_c a$, and we write $a <_c a'$ when $a \leq_c a'$ but not $a \sim_c a'$. On the other side of the market, each applicant $a \in A$ ranks the programs according to a *strict total order* $<_a$, i.e. for any programs c, c' such that $c >_a \emptyset$ and $c' >_a \emptyset$ (i.e. c, c' are acceptable to student a), we have either $c <_a c'$ or $c' <_a c$.

The rest of this section is organized as follows. In Section 4.1 we revisit the college admissions problem with strict preferences in both sides. In Section 4.2 we extend this model to incorporate non-strict preferences and flexible quotas. In Section 4.3 we describe how to adapt the model to incorporate the affirmative action, and in Section 4.4 we characterize the properties of this mechanism.

4.1. Stable Matchings with Strict Preferences

We start with some basic definitions. A *matching* is an assignment $\mu \subseteq V$ such that for each applicant a the set of assigned programs $\mu(a)$ has at most one element, while for each program c the set of assigned students $\mu(c)$ has at most q_c elements. When preferences are strict no ties will occur, so we don't have to worry about flexible quotas.

A matching μ is *stable* if for all pairs $(c,a) \in V \setminus \mu$ we have that either the set $\mu(a)$ has an element preferred over c in the strict order $<_a$, or the set $\mu(c)$ contains q_c elements preferred over a in the strict order $<_c$. In the first case the applicant a likes the match proposed by μ better than c, while in the second case the program has all its positions filled with students strictly preferred over a. If both conditions fail simultaneously a and c would be better off by being matched together rather than accepting the assignment μ, in which case (c,a) forms a *blocking pair*. In other words, a matching is stable if it has no blocking pairs.

An *instance* of the college admissions problem can be fully described in terms of a pair $\Gamma = (G,q)$, where $G = (V,E)$ is an admission graph consisting of a set of feasible nodes V on a grid $C \times A$ and a set of directed arcs $E \subseteq V \times V$ that represent programs and applicants preferences; and q is a vector of quotas. Each row in the grid represents a program $c \in C$, and each column an applicant $a \in A$. The preferences of program c are encoded by horizontal arcs from (c,a) to (c,a') whenever $a \leq_c a'$, and those of student a by vertical arcs from (c,a) to (c',a) representing $c <_a c'$. Figure 1 shows a minor variation of the example considered in Baïou and Balinski (2004). For simplicity, the arcs that can be inferred by transitivity are omitted.

By exploiting this graph representation, Baïou and Balinski (2004) develop an algorithm to find two extreme (and the most interesting) stable matchings. The algorithm works by recursively eliminating pairs $(c,a) \in V$ that are strictly dominated. More precisely, (c,a) is *a-dominated* if there are q_c or more applicants that have c as their top choice and dominate a in the strict preference $<_c$. In this case, program c is guaranteed to fill its quota with applicants above a so that a has no chance to be assigned to c, and the pair (c,a) can be eliminated from further consideration. Similarly,

Figure 1 Admission graph with strict preferences.

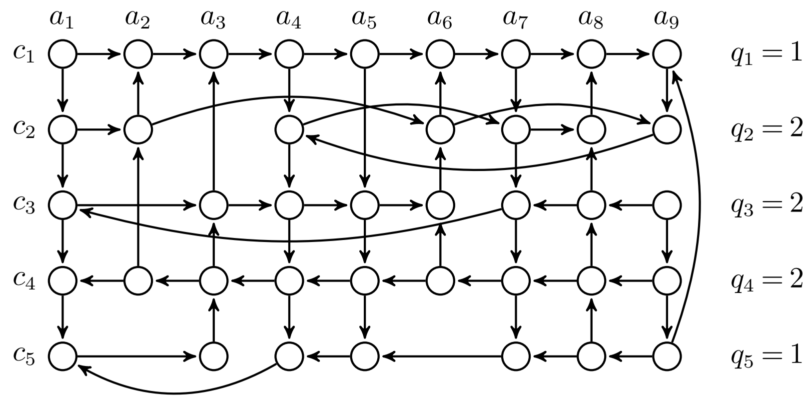
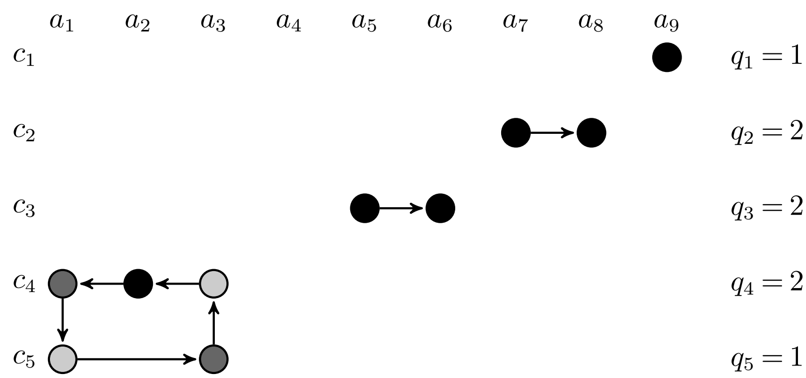


Figure 2 Domination-free subgraph, student-optimal and university-optimal matchings



(c, a) is *c-dominated* if there is a program $\tilde{c} >_a c$ which places a among the top $q_{\tilde{c}}$ applicants (*i.e.* less than $q_{\tilde{c}}$ applicants are ranked above a).

Since removing a node may change the top choices for students and programs, new dominations may arise and the elimination must continue recursively until no further dominated nodes are found. Upon completion, the admission graph is reduced to a *domination-free subgraph* $G^* = (V^*, E^*)$, with node set $V^* \subseteq V$ in which all dominated nodes have been removed. Figure 2 presents the domination free-subgraph corresponding to the instance described in Figure 1.

The domination free equivalent subgraph G^* contains all possible stable allocations, including the two most interesting (and extreme) cases: the *student-optimal* matching μ_A^*, that assigns each applicant $a \in A$ to its best remaining choice in G^* (if any); and the *university-optimal* matching μ_C^* that assigns to each program $c \in C$ its q_c top choices in G^*. In Figure 2 the light gray circles are nodes belonging exclusively to μ_A^*, the gray circles are nodes only in μ_C^*, and the black circles represent the nodes that are simultaneously in μ^*_A and μ^*_C. Notice that these two extreme allocations are not the same.

4.2. Stability with Ties and Flexible Quotas: FQ-matchings

Consider now the case where preferences of programs may not be strict, and that programs are required to adjust their quotas to include all potential candidates tied in the last vacancy. More precisely, a program c may exceed its quota q_c only if the last group of students admitted are in a tie and upon rejecting all these students there are unassigned vacancies. We also impose a non-discrimination condition: an applicant a' who is tied with a student a admitted to a program c must himself be granted admission to c or better. The following definitions state these conditions formally.

Definition 1. We say that μ satisfies *quotas-up-to-ties* if for each program c and $a \in \mu(c)$ the set of strictly preferred students assigned to c satisfies $|\{a' \in \mu(c) : a' >_c a\}| < q_c$.

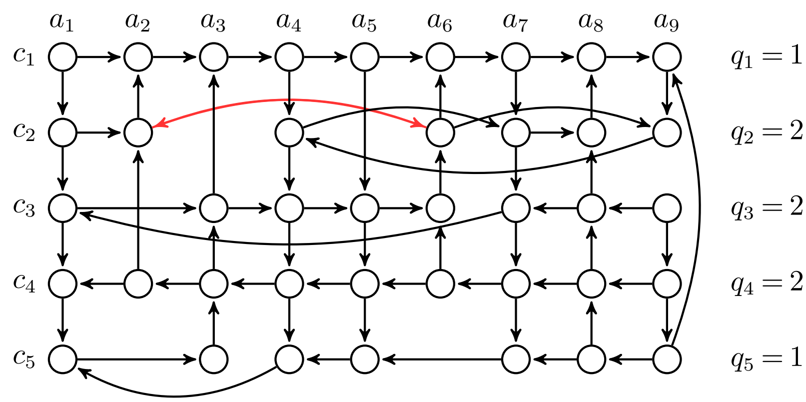
Definition 2. We say that μ satisfies *non-discrimination* if whenever $a \in \mu(c)$ and $a' \sim_c a$ with $(c, a') \in V$ then $a' \in \mu(c')$ for some program $c' \geq_{a'} c$.

With these preliminary definitions we introduce our notion of matching, which requires in addition that each applicant is assigned to at most one program.

Definition 3. A *matching with flexible quotas* (FQ-matching) is an assignment $\mu \subseteq V$ that satisfies quotas-up-to-ties and non-discrimination, and for which every applicant $a \in A$ is assigned to at most one program so that $\mu(a)$ has at most one element.

By construction, FQ-matchings are stable. In Appendix A we provide a formal proof of stability, and in Appendix B we describe how our definition of FQ-matching relates to other notions of stability, such as weak, strong and super-stability. In addition, notice that if there are no ties in the preferences of programs then non-discrimination holds trivially, while quotas-up-to-ties reduces to $|\mu(c)| \leq q_c$; hence, FQ-matching coincides with the standard notion of stable matching.

Figure 3 Admission graph with non-strict preferences



Now we analyze how to compute FQ-matchings. Notice first that, given any instance $\Gamma = (G, q)$ with $G = (V, E)$, we can build two extreme assignments (which are not necessarily FQ-matchings): one that greedily assigns each student to its best option and the other that greedily fills the quota of each program with its top q_c applicants. We say that an applicant a is top q_c for program c if $(c, a) \in V$ and there are less than q_c candidates strictly above him, namely, $|\{(c, a') \in V : a' >_c a\}| < q_c$.

Definition 4. The *student-optimal greedy assignment* μ_A matches each applicant a to his/her top preference in G (if any), while the *university-optimal greedy assignment* μ_C is obtained similarly by assigning to each program c its top q_c applicants.

By construction both μ_A and μ_C are stable, and they also satisfy non-discrimination. However, they may fail to be FQ-matchings as they ignore the quota restrictions. Indeed, while μ_A assigns each student to at most one program, there is no reason that quotas-up-to-ties should be satisfied, and in fact this may fail for the most demanded programs. This is illustrated[11] in Figure 3, where $\mu_A(c_1) = \{a_2, a_3, a_6, a_8\}$ and therefore μ_A is not an FQ-matching.

Similarly, although μ_C always satisfies quotas-up-to-ties, a good student may be top ranked in several programs and could be multiply assigned. Note however that in the previous example μ_C assigns each student to exactly one program (except for a_4 who remains unassigned) so in this case μ_C provides a stable FQ-matching.

The next theorem shows that quota violations in the greedy assignments do not occur when G has no strictly dominated nodes. Here we keep the same definition of a-dominated and c-dominated nodes stated previously.

THEOREM 1.

a) *If the instance* Γ *has no a-dominated nodes then* μ_A *is a stable FQ-matching.*

b) *If the instance* Γ *has no c-dominated nodes then* μ_C *is a stable FQ-matching.*

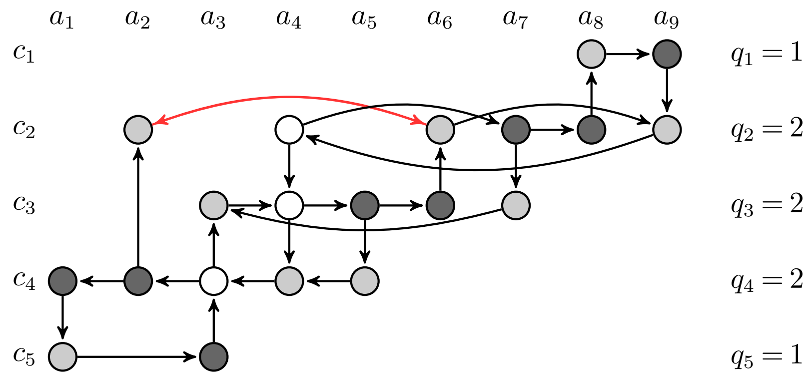
This result suggests that μ_A and μ_C should be computed only after removing the corresponding strictly dominated nodes. In order to justify this procedure we must show first that such nodes have no chance of being in any stable FQ-matching and that they can be safely dropped without destroying nor creating any new stable FQ-matching. In the following we say that a sub-instance $\Gamma' = (G', q)$ with $G' = (V', E')$, $V' \subseteq V$ and $E' \subseteq V' \times V'$, is *FQ-equivalent* to Γ if it has exactly the same stable FQ-matchings as Γ.

THEOREM 2. *Given an instance* $\Gamma = (G, q)$ *with* $G = (V, E)$*, if* $(c, a) \in V$ *is either a-dominated or c-dominated then* $(c, a) \notin \mu$ *for any stable FQ-matching* μ*, and the instance* $\Gamma' = (G \setminus (c, a), q)$ *is equivalent to* Γ.

The previous results justify the following procedure to compute stable matchings. Start by removing every dominated node. After removing some nodes, the top choices for students and programs may change and new dominations may arise, so the process must continue recursively until no further dominated nodes are found. Upon completion we obtain a domination-free subgraph $G^* \subseteq G$. By Theorem 2, all along this elimination process we preserve exactly the same stable FQ-matchings as in the original instance Γ. Then, in the last step we use the greedy algorithm in G^* to obtain an allocation. Theorem 1 ensures that the greedy assignments μ_A and μ_C computed for $\Gamma^* = (G^*, q)$ are stable FQ-matchings[12].

Applying this procedure to the example in Figure 3 we obtain the reduced graph G^* presented in Figure 4 and the greedy assignments μ_A (light gray) and μ_C (gray). In this case $\mu_A \cap \mu_C = \emptyset$ so there are no black nodes. This example shows that the inclusion of a single tie may considerably change the outcome. Further details on this example are provided in the Appendix E.

Figure 4 Domination-free equivalent and optimal matchings with flexible quotas.



4.3. Unifying Admission Tracks

Using the model described in the previous section we can directly include the affirmative action and solve both admission tracks (Regular and BEA) simultaneously.

To accomplish this, we consider a unified admission instance $\Gamma^U = (G^U, q^U)$ where each program $c \in C$ is split into two virtual programs c^R and c^B that represent the Regular and the BEA processes, with quotas q_c^R and q_c^B respectively. The preferences of students that are not shortlisted for the scholarship remain unchanged. In contrast, each program in the preference list of a BEA student is also divided into the two virtual programs, giving a higher position in the preference list to the Regular process, i.e. for any two programs with $c_1 >_a c_2$, the new preference order is $c_1^R >_a c_1^B >_a c_2^R >_a c_2^B$.

We decided to use this order because DEMRE wanted to prioritize BEA students. Then, by applying to the regular seats first, BEA students with good scores can be admitted in regular seats, reducing the competition for reserve seats and therefore weakly increasing the total number of BEA students admitted in the system. This idea was formalized in Dur et al. (2013), and recently extended to more reserve groups in Dur et al. (2016b).

Our next theorem shows that every student is weakly better off compared to the sequential solution. Let μ_A^S be the student-optimal FQ-matching obtained by applying the sequential process, and let $\mu_A(\Gamma)$ be the student-optimal FQ-matching for an instance $\Gamma = (G, q)$. In addition, let

Γ^R and Γ^B be the Regular and BEA instances respectively. Then, the *sequential student-optimal FQ-matching* μ_A^S is defined as

$$\mu_A^s(a) = \begin{cases} \mu_A(\Gamma^B)(a) \text{ if } \mu_A(\Gamma^B)(a) \neq \emptyset \\ \mu_A(\Gamma^R)(a) \text{ otherwise.} \end{cases} \tag{1}$$

THEOREM 3. *The unified student-optimal FQ-matching $\mu_A(\Gamma^U)$ dominates the sequential student-optimal assignment μ_A^S, that is to say, $\mu_A(\Gamma^U)(a) \geq_a \mu_A^S(a)$ for all $a \in A$.*

4.4. Basic Properties of FQ-matchings

We already know that FQ-matchings are stable and satisfy *quotas-up-to-ties* and *non-discrimintation.* In what follows we discuss four other relevant properties.

4.4.1. Optimality. Ensuring that a given matching is optimal among the set of stable matchings guarantees that no agent (student or program) can be better off without harming another agent. In the Appendix A we show that the allocations μ_A and μ_C obtained from the procedure described in the previous section are the optimal matchings for students and programs within the set of stable FQ-matchings.

4.4.2. Lack of Monotonicity. Monotonicity is a relevant feature since it ensures that any improvement in the valuation of an agent cannot harm his assignment. The examples we provide in the Appendix E show that neither the student-optimal nor the university-optimal FQ-matchings are monotone for students. This lack of monotonicity implies that a student could improve his outcome by strategically under-performing in the tests. However, to accomplish this he would have to know beforehand the preferences and scores of all other students, which is not possible since the results are announced to all students at the same time. Thus, in practice the lack of monotonicity of μ_A and μ_C is not a serious concern.

4.4.3. Lack of Strategy-Proofness. A strategy-proof (SP) mechanism ensures that no student can be assigned to a more preferred program by misreporting their true preferences. In our context we only focus on strategy-proofness for students since applicants have to state their preferences after universities and when they already know their scores. Moreover, we will assume that the reported weights of each program reflect the true "preference orders" over students, although the underlying preferences of programs could be different than just a rank over students. For instance, programs or the universities they belong to could also have preferences over sets of students, and the quota policy could be seen as a strategy to reach some distributional concern. The examples in the Appendix E show that neither the student-optimal FQ-matching nor the university-optimal FQ-matching are strategy-proof.

4.4.4. Strategy-Proofness in the Large. The lack of strategy-proofness can be troublesome because it may induce agents to misreport their preferences strategically, giving an unfair advantage to more sophisticated students. However, we claim that this is not a problem in our setting due to the large size of the market.

According to Azevedo and Budish (2017), a mechanism is strategy-proof in the large (SP-L) if, for any full-support i.i.d. distribution of students' reports, being truthful is approximately optimal in large markets. SP-L is less restrictive than the classic notion of strategy-proofness, since it does not require truthful reporting to be optimal for any market size, and only asks it to be approximately optimal in large markets.

To use the framework in Azevedo and Budish (2017) we need some definitions. A mechanism is *semi-anonymous* if agents can be divided in a finite set of types, and their outcome depends only in their actions and their type. In addition, a mechanism is *envy-free* if no agent prefers the assignment of another agent with the same type. Then, Theorem 1 in Azevedo and Budish (2017) shows that a sufficient condition for a *semi-anonymous* mechanism to be SP-L is envy-freeness. Therefore, to show that our mechanism is SP-L it is enough to show that it satisfies envy-freeness.

The proof of this is almost direct if we assume that students' types are given by their vector of scores and whether they belong to the Regular or BEA groups. Conditional on their types, we have already shown that both university-optimal and student-optimal FQ-matchings are stable and satisfy *non-discrimination*. This directly implies that both FQ-matchings satisfy envy-freeness, and therefore we conclude that our mechanism is approximately strategy-proof and interim incentive compatible as long as the size of the market is large, which is the case in the Chilean college admissions problem.

4.5. Flexible quotas as a sensible policy

As we have already discussed, the implementation of an FQ-matching involves a trade-off between sacrificing SP and potentially exceeding capacities, and obtaining a better allocation for students in the Pareto sense. If universities don't want to arbitrarily discriminate students and their marginal cost of increasing their capacity is low enough, allowing for ties and flexible quotas can be a sensible policy because it translates to a Pareto improvement for students[13] and eliminates any fairness concerns that can arise due to tie breaking rules[14]. On the other hand, the implementation of an FQ-matching implies the loss of strategy-proofness, raising the question on how important this issue is in the current setting. However, the fact that our mechanism is SP-L and that the Chilean college admissions problem is large ensures that strategic behavior is not a concern. In fact, as Azevedo and Budish (2017) argue, the relevant distinction for practice in a large market is whether a mechanism is "SP-L vs not SP-L" and not "SP vs not SP", since students in a large market do not know what are the realized reports of every other student, so imposing optimality of truthful reporting against every report realization (as in SP) is too strong.

5. Implementation

In this section we report the results on the implementation of this project. We start providing a general description of the Chilean college admissions problem. Then, we describe the results of our first goal, which was to find the algorithm that has been used in Chile to perform the allocation. Finally, we close this section with the results of unifying the admission tracks and other additional side effects of this project.

5.1. General Description

In Table 1 we present general descriptives on the programs that are part of the centralized admission system. We observe that the number of universities has not changed, while the number of programs has slightly increased. However, we see that the number of seats available has decreased over the years.

Table 1 General description — Programs

	2014	2015	2016
Universities	33	33	33
Programs	1,419	1,423	1,436
Regular seats	110,380	105,516	105,513
Reserve seats	4,394	4,422	4,295

To describe the other side of the market, in Table 2 we present the number of Regular and BEA students at each stage of the admission process. A *participant* is a student that registered to participate in the standardized national exam and has at least one valid score. Once the results of the national exam are published, participants have five days to submit their applications to the centralized clearinghouse. We refer to the students that apply to at least one program that is part of the centralized system as *applicants*. Finally, we refer to students that were admitted by a program that is part of the centralized system as *assigned*.

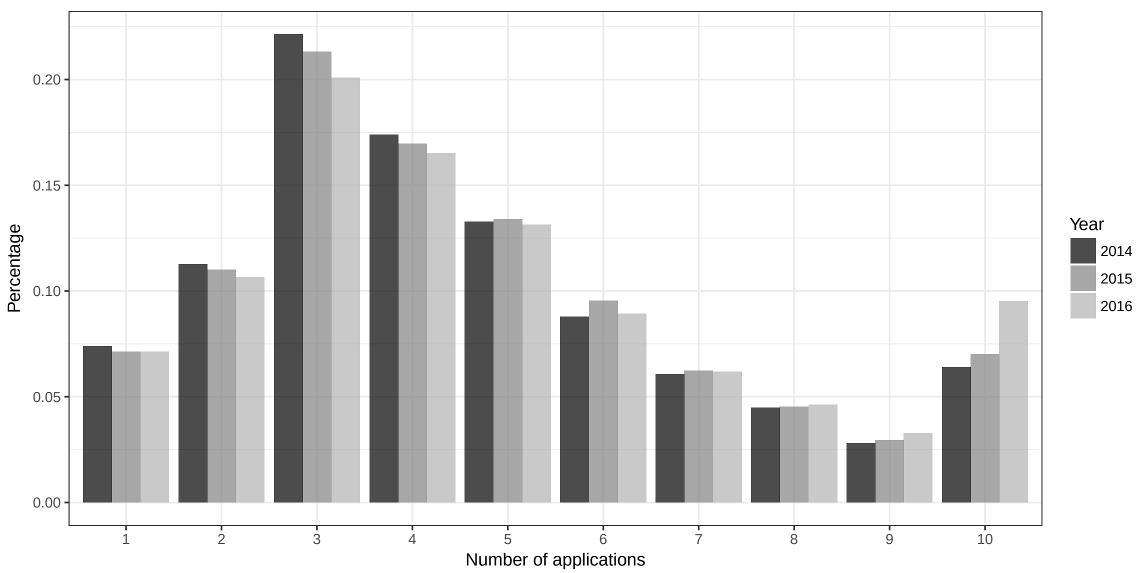
First, we observe that the total number of participants has increased over the years, reaching a total of 267,311 participants in 2016. Second, comparing the number of participants and the number of applicants we observe that less than a half of the students that registered for the national exam applied to programs that are part of the system. The main reason for this is that CRUCH sets a minimum threshold of 450 points for students to be eligible by any program that is part of the system, and since tests are standardized to have mean 500 roughly half of the students will not satisfy this criteria.

Table 2 General description — students

		Regular			BEA		
		2014	2015	2016	2014	2015	2016
Participants	Total	228,318	241,873	250,320	15,990	16,710	16,911
Applicants	Total	108,144	113,900	129,896	11,017	11,688	12,010
Assigned	Regular seats	86,048	87,466	90,741	9,520	10,154	8,886
	Reserve seats	-	-	-	1,325	1,404	1,345

Also related to the application process, in Figure 5 we show the distribution of applications per student for each year.

Figure 5 Distribution of applications per student



The median number of applications is 4 and the share of each number of applications stays roughly constant across years. As students are restricted to submit a list with no more than 10 programs, we observe that between 5% and 10% of applicants submit a full list of 10 applications.

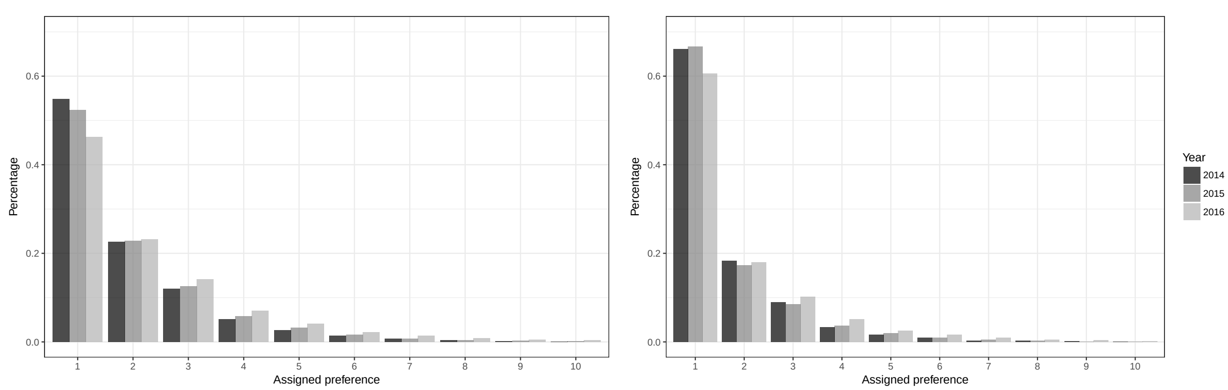
Notice that some universities further restrict the number of programs to which a student can apply, and also the position that an application can take in the applicant's list[15]. Theoretically, any restriction on the length of the application list will break strategy-proofness. Nevertheless, whether these constraints are binding or not in practice, and what the strategic implications are for students, are questions for future research.

Even though the number of participants and applicants have increased over the years, Table 2 shows that the total number of admitted students has decreased, in line with the reduction of seats that we see in Table 1. In fact, comparing applicants and assigned students we find that close to 80% get assigned to some program in 2014, 78% in 2015 and 70% in 2016.

In Figures 6a and 6b we show the distribution of the preference of assignment for Regular and BEA students respectively. We see that close to 50% of students get assigned to their first reported preference, and close to 90% get assigned to one of their first three preferences. Although both Regular and BEA students exhibit the same pattern of assignment, it is worth noticing that the latter get assigned consistently more to their first preference than Regular students.

Figure 6 Preference of assignment

(a) Regular (b) BEA



Finally, in Table 3 we present demographic characteristics of the students that are assigned.

Table 3 General description — Assigned

		Regular			BEA		
		2014	2015	2016	2014	2015	2016
Assigned	Total	86,048	87,466	90,741	9,745	10,378	10,231
Gender	Female	49.5%	49.5%	50.2%	57.6%	58.4%	59.5%
Average Scores	Math/Verbal[1]	588	589.3	588.7	591.1	595.9	593.4
	NEM[2]	586	588.4	592	696.8	696.7	700.5
	Rank[3]	608.7	614.4	615.5	770.6	776.9	774.5
Income[4]	[0, $288]	28.8%	26.4%	23.5%	46%	42.6%	40.9%
	($288, $576]	26.7%	27.5%	29%	34.6%	37.1%	38.6%
	($576, $1,584]	26.8%	27.5%	28.9%	18.7%	19.5%	19.1%
	> $1,584	17.8%	18.6%	18.7%	0.7%	0.8%	1.4%
High-School	Private	23.4%	23.4%	22.5%	0%	0%	0%
	Voucher[5]	52.4%	52.9%	53.3%	60.7%	61%	61.4%
	Public	24.2%	23.7%	24.1%	39.3%	39%	38.6%

[1] Score constructed with the average Math score and Verbal score. For students using scores from previous year, we considered the maximum of both averages.
[2] Score constructed with the average grade along high-school.
[3] Score constructed with the relative position of the student among his/her classmates.
[4] Gross Family monthly income in thousands Chilean pesos (nominal).
[5] Partially Subsidized schools.

The fact that around 23% of admitted students graduated from a private school is surprising, considering that they represent only 12% of the total number of participants in the admission process. Similarly, students from the highest income group only represent 9% of the total number of participants, but they account for 18% of admitted students. These numbers shed some light on the huge inequalities in opportunities that characterize the Chilean college admission process.

The point of having reserve seats is to alleviate these inequalities and favor underrepresented groups. The comparison of the left and right columns in Table 3 illustrates this. Compared to Regular students, in the BEA group the fraction of female students is higher, the average scores in all the exams are also higher, the fraction of students with higher income levels is smaller, and most of them come from public/voucher schools.

5.2. Determining the Current Mechanism

Our first goal was to identify which mechanism has been used to solve the Chilean college admissions problem. After implementing the algorithms described above and including all the constraints that are part of the system, we solved the admission instances from 2012 to 2014, comparing the FQ-student-optimal and FQ-university-optimal allocations with the official results obtained using DEMRE's black box. Based on this comparison, the rules of the system and evidence provided by DEMRE we concluded that the algorithm used is equivalent to the FQ-university-optimal matching, as the results are exactly the same for all the years considered.

Given that our algorithm returns all stable allocations for each instance, by comparing the two extreme assignments (FQ-student and FQ-univesity optimal) we find that the number of differences between these allocations has been at most 10 since 2012. This suggests that the size of the core of stable assignments in the Chilean case is rather small, supporting the theoretical results in Roth and Peranson (2002) and Ashlagi et al. (2017).

Even though the number of differences is small, we proposed DEMRE to adopt a FQ-student optimal matching because it benefits some students but, more importantly, because it is a message for students that the mechanism aims to give them the best possible allocation. DEMRE agreed with this view, and after a pilot version in 2014 they adopted our FQ-student optimal algorithm to perform the allocation since 2015.

5.3. Integrating Admission Tracks

Having determined the algorithm that is used to perform the allocation, our second goal was to integrate the admission tracks in order to alleviate the aforementioned inefficiencies. To accomplish this, we implemented the framework described in Section 4.3 and we ran it in parallel during the admission processes of 2014 and 2015. Based on the results, we convinced DEMRE to adopt our unified FQ-student optimal allocation in 2016, and this allocation has been the official mechanism used since then. In this section we report the results from our simulations (2014 and 2015), and the actual impact of our implementation in 2016.

In Table 4 we present a summary of the results. The first row presents the number of students that would have been double-assigned under the old system. The second row presents the number of students that improved their assignment compared to the old system. Finally, the third row shows the number of students who are assigned to some program under the new system and weren't assigned in the old system.

Table 4 Impact of Unified Assignment 2014-2016

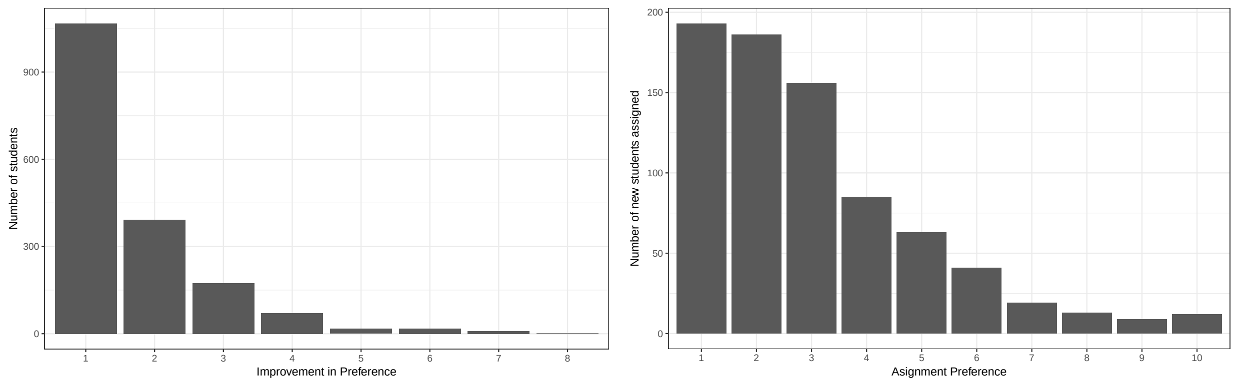
	2014	2015	2016
Double Assignments	1,100	1,180	1,127
Improvements	1,737	1,915	1,749
New assignments	568	672	777

We observe that the number of students that benefit from the better use of vacancies is larger than the number of vacancies lost due to double-assignments. The reason is that a student that directly benefits from a new vacancy releases a vacancy that can be used by another student, who in turn allows another student to take his old seat, and so on. This chain of improvements eventually ends either because there are no students wait-listed in that program, or because they reach a student who was unassigned and therefore does not release another seat. Overall, we observe that around 3% of students who decide to apply to a program in the system benefited from our implementation, and this number is relatively constant across years.

Improving the assignment of students is relevant because the probability of enrollment is increasing in the preference of assignment[16], and most programs in the system have positive and high

Figure 7 Benefits of unifying the admission tracks - 2016

(a) Improvement in assigned preference (b) Preference assigned of new assigned students



expected returns (see Lara et al. (2017)). Moreover, there is evidence that students that were assigned in low listed preferences have a higher probability of future dropout from their programs, even after enrollment (see Canales and De los Ríos (2007)).

In Figure 7a we present the distribution of improvements in allocated preferences due to our new approach. We observe that most students improve in their allocation by getting assigned to the program listed immediately above the program they were assigned previously (improvement equal to 1). In Figure 7b we show the preference of assignment for those students who were not assigned under the old system and thanks to the new system are allocated. Most of the students that were unassigned under the sequential allocation benefit from the unified allocation by getting their top choice. A potential reason for this is that an important fraction of these students apply to less than three programs.

We provide more details on those students that benefit from unifying the admission tracks in Table 5.

Table 5 General description — impact of unifying admission tracks

		Improvements						New Assignments					
		Regular			BEA			Regular			BEA		
		2014	2015	2016	2014	2015	2016	2014	2015	2016	2014	2015	2016
Assigned	Total	1,592	1,791	1,640	145	124	109	548	647	765	20	25	12
Gender	Female	47.2%	44.4%	45.7%	66.9%	60.5%	62.4%	52.2%	47.9%	51.6%	65%	68%	58.3%
Average Scores	Math/Verbal[1]	584.2	582	579.7	579.8	573.7	584.1	560.8	551.2	548.2	564.8	561	558
	NEM[2]	577.8	576.9	575	686	675.7	691.1	540	538.9	538.4	677.1	658	689.2
	Rank[3]	600.2	600.6	596.1	755.8	752.9	770	554.8	556.4	553.5	753.2	734	764.1
Income[4]	[$0, $288]	28.1%	27.4%	25.2%	48.3%	41.9%	48.6%	30.1%	32.3%	28.8%	60%	36%	75%
	($288, $576]	30.2%	29%	31.3%	29.7%	39.5%	37.6%	34.5%	32.9%	34%	25%	32%	8.3%
	($576, $1,584]	27.4%	29.3%	28.5%	20%	18.5%	13.8%	26.1%	24.3%	28.8%	15%	32%	16.7%
	> $1,584	14.3%	14.3%	14.9%	2.1%	0%	0%	9.3%	10.5%	8.5%	0%	0%	0%
High-School	Private	18.4%	18.6%	16.2%	0%	0%	0%	13.8%	14.5%	13.4%	0%	0%	0%
	Voucher[5]	56.7%	57.3%	56.7%	63.4%	61.3%	65.1%	59.5%	61.4%	61.5%	60%	72%	58.3%
	Public	26.7%	24.9%	24.2%	27.1%	36.6%	38.7%	34.9%	24.1%	25%	40%	28%	41.7%

[1] Score constructed with the average Math score and Verbal score. For students using scores from previous year, we considered the maximum of both averages.
[2] Score constructed with the average grade along high-school.
[3] Score constructed with the relative position of the student among his/her classmates.
[4] Gross Family monthly income in thousands Chilean pesos (nominal).
[5] Partially Subsidized schools.

We first observe that most of the students that benefit from unifying the admission tracks are Regular students. The reason is that vacancies that were dropped by a BEA student with double assignment are now used by other students, and this generates improvement chains that reach other (mostly Regular) students. In addition, comparing the characteristics of those who improve (*Improvements*) with those who get assigned and would not under the old system (*New Assignments*) we find that the latter group has lower scores, and as a larger fraction of students who come from lower income families and public schools. The reason for this is that students who improved were also assigned under the old system, while those from the *New Assignments* group were not. Therefore, students from the *Improvements* group have on average higher scores, and these are positively correlated with family income.

The differences in terms of scores and demographics are also present if we compare these groups with the overall group of assigned students described in Table 3. Indeed, previously assigned students have on average higher scores and higher family income than students that were benefited by the unified assignment. For instance, the share of assigned students coming from private high-schools was about 24% for Regular students, while it was close to 18% and 14% for students in the groups of *Improvements* and *New Assignments* respectively.

Another interesting result is that most of BEA students are assigned to regular seats, and more than half of the reserve seats remain unfilled. Indeed, we proposed to DEMRE that they could use the unfilled reserve seats with students from the Regular process, but they decided not to because some universities “were not open to this option”. Nevertheless, we are working on convincing CRUCH to incorporate this in future versions of the admission process.

5.4. Additional Side Effects

In terms of running times, our implementation considerably outperforms the algorithm used previously by DEMRE to solve the admissions problem. In fact, their black-box software takes up to 5 hours to return the final assignment, while our implementation solves the problem in less than 2 minutes on a standard laptop. This time reduction has had a significant impact since it allows to evaluate different policy changes in the system, such as the inclusion of new admission criteria, the impact of new instruments, and the redesign of affirmative action policies performed by the mechanism. In particular, the new algorithm has been used to evaluate the effect of including High School Grade Rankings in the Chilean admission process by performing several simulations changing the conditions in which this new instrument is included by universities (Larroucau et al. (2015)). This kind of exercise was not possible in the past due to the computational time involved in the black-box implementation.

The efficiency gains also opened the possibility to simulate and evaluate the effects of policy changes that could stress the system in the future. For instance, the impact of free-of-charge access, the inclusion of professional and vocational institutions to the admission process, and the implementation of admission quotas for underrepresented groups. All of these vital questions could be handled by building structural models for which computational efficiency is crucial.

6. Conclusions

We investigate how the Chilean college admissions system works. There are two main features that make the Chilean system different from the classic college admissions problem: (i) preferences of colleges are not strict, and all students tied for the last seat of a program must be assigned; and (ii) the system also considers an affirmative action that is solved sequentially after the Regular process. Then, students who benefit from the affirmative action policies can be double-assigned, introducing a series of inefficiencies in the assignment and enrollment processes. Even though the authorities were aware of this problem, they couldn't solve it because they relied in a black-box software that couldn't be updated to incorporate the affirmative action.

To determine which mechanism was used, we develop an algorithm that finds all stable allocations satisfying the rules of the system, i.e. flexible quotas and non-discrimination of tied students. We also introduce the notion of FQ-matching to account for these features, and we characterize its main properties. We show that this mechanism leads to the optimal stable allocations satisfying flexible quotas and non-discrimination, but it lacks monotonicity and strategy-proofness. Nevertheless, we also show that our mechanism is SP-L, and since the Chilean college admissions problem is large, the lack of SP is not a major concern.

By comparing the results of our algorithm with historical data we find that the algorithm that has been used is a variation of the university-optimal stable assignment that satisfies flexible quotas and non-discrimination. Even though the number of differences is small, we convinced DEMRE to switch to the FQ-student optimal mechanism, which was finally adopted in 2015 after a pilot version in 2014.

Having identified the algorithm, we propose a new method to incorporate the affirmative action that is based on treating regular and reserve seats as different programs. The unified approach to solve the problem was adopted and implemented by DEMRE in 2016, after two years of analyzing its potential impact. The results of the implementation in 2016, as well as the pilot results in 2014 and 2015, show that around 3% of the total number of students that are admitted each year benefit from the unified assignment. From the students who actually benefited in 2016, 30.8% would not have been assigned to any program under the old system, and 69.2% are students who improved compared to what they would get under the old system. The benefited students have, on average, lower scores and lower family incomes compared to the students that would have been assigned under the old system.

In addition to its direct impact on students, the efficiency of our algorithm reduced the running time by about two orders of magnitude relative to the old system, enabling the realization of simulations to evaluate different policies oriented to make the admission process more fair and inclusive. Finally, our method helped to improve the transparency of the system, and allowed other policies to be implemented on top of it.

Certainly there is a lot of room for further improvement, since there are still many BEA vacancies that are not used by any students. Nevertheless, we hope that the current results encourage the Chilean authorities to keep improving the system, increasing the overall efficiency of the process and giving more and better chances of admission to under-represented students.

Appendix

Appendix A: Proofs

To ease exposition we introduce some more notation. For any feasible pair $(c,a) \in V$ we write $c \rhd_a \mu(a)$ if a is either unassigned or strictly prefers c to his assigned program in μ, while $a \rhd_c \mu(c)$ means that c has not completed its vacancies or strictly prefers a to its worst assigned student. Formally, we define

- $c \rhd_a \mu(a) \Leftrightarrow$ the set $\mu(a)$ is either empty or it contains an element $c' <_a c$,
- $a \rhd_c \mu(c) \Leftrightarrow$ the set $\mu(c)$ has fewer than q_c elements or it contains an element $a' <_c a$

as well as the analog notions with non-strict preferences

- $c \trianglerighteq_a \mu(a) \Leftrightarrow$ the set $\mu(a)$ is either empty or it contains an element $c' \leq_a c$,
- $a \trianglerighteq_c \mu(c) \Leftrightarrow$ the set $\mu(c)$ has fewer than q_c elements or it contains an element $a' \leq_c a$.

Proof (Theorem 1): a) Since μ_A assigns each student a to its most preferred program it is obvious that it satisfies non-discrimination as well as stability, and $\mu_A(a)$ contains at most one element. It remains to show that it satisfies quotas-up-to-ties, which follows directly from the definition of μ_A and the fact that there are no a-dominated nodes. Indeed, for each program c and $a \in \mu_A(c)$ the node $(c,a) \in V$ is not a-dominated so that the set $\{a' \in \mu_A(c) : a' >_c a\}$ cannot contain q_c or more elements.

b) We already observed that in general μ_C is stable and satisfies quotas-up-to-ties. Also no student a can be simultaneously among the top q_c applicants for two different programs: otherwise the one which is less preferred by a would be c-dominated. Hence, $\mu_C(a)$ contains at most one element. It remains to show that μ_C satisfies non-discrimination. Consider a student $a \in \mu_C(c)$ and a node $(c,a') \in V$ with $a' \sim_c a$. By definition of μ_C we have that a is among the top q_c applicants for c, and then the same holds for a' so that $a' \in \mu_C(c)$.

Proof (Theorem 2): Let μ be a FQ-matching in Γ. Clearly the conditions for FQ-matching are preserved when we remove a node not in μ. Hence, if we prove the first assertion $(c,a) \notin \mu$ it will also follow that μ is a FQ-matching for Γ'. Now, if (c,a) is c-dominated there is a program $\tilde{c} >_a c$ for which a is top $q_{\tilde{c}}$, and therefore (c,a) cannot belong to μ since otherwise $(\tilde{c},a)$ would be a blocking pair. Similarly, if (c,a) is a-dominated, then there are q_c or more applicants $a' >_c a$ with c as their top choice. If $(c,a) \in \mu$, then all the pairs (c,a') must also be in μ since otherwise we would have a blocking pair, but this contradicts quotas-up-to-ties so that $(c,a) \in \mu$ is impossible in this case too.

Let us prove conversely that any assignment μ that is a FQ-matching for Γ' is also a FQ-matching for Γ. The properties of quotas-up-to-ties and $|\mu(a)| \leq 1$ involve only the nodes in μ and are not affected by the addition of the node (c,a). Hence, it suffices to establish non-discrimination and stability.

Non-discrimination. Since this property already holds for all the nodes in $V \setminus \{(c,a)\}$ we must only prove that it also holds for (c,a). Indeed, suppose that a is tied with some $a' \in \mu(c)$. Recall that (c,a) is strictly dominated, however it cannot be a-dominated since otherwise the same would occur for a' and it could not have been assigned to c. Hence, it must be the case that (c,a) is c-dominated which means that a is among the top $q_{\tilde{c}}$ applicants for some program $\tilde{c} >_a c$. But then a must be assigned to an even better choice $c' \geq_a \tilde{c}$, since otherwise $(\tilde{c},a)$ would provide a blocking pair in Γ'. Then $a \in \mu(c')$ for some $c' >_a c$ and non-discrimination holds for (c,a) as claimed.

Stability. We already know that there are no blocking pairs in Γ'. Let us prove that (c,a) is not a blocking pair in Γ. Suppose first that (c,a) is c-dominated so that a is top $q_{\tilde{c}}$ on some program $\tilde{c} >_a c$. In this case μ must assign a to some $c' \geq_a \tilde{c}$ since otherwise $(\tilde{c},a)$ would be a blocking pair in Γ', and therefore we do not have $c \rhd_a \mu(a)$. Similarly, if (c,a) is a-dominated there are q_c or more applicants a' ranked strictly above a that have c as their top choice. All the nodes (c,a') are in Γ' so that $\mu(c)$ must fill the quota q_c with applicants at least as good as the lowest ranked of these a'. Since this is still above a we do not have $a \rhd_c \mu(c)$. Combining both cases, we cannot have $c \rhd_a \mu(a)$ and $a \rhd_c \mu(c)$, proving that (c,a) is not a blocking pair.

Proof (Theorem 3): We first observe that the Regular matching $\mu_A(\Gamma^R)$ is the same as the one obtained from the unified graph G^U by setting the BEA quotas to 0, that is $q_c^B = 0$ for all $c \in C$. Since increasing quotas only benefits students in the student-optimal allocation we have that $\mu_A(\Gamma^R)(a) \leq_a \mu_A(\Gamma^U)(a)$ for all applicants $a \in A$. This already shows that all non-BEA students are not worse off in $\mu_A(\Gamma^U)$ as compared to μ_A^s.

Let us consider next the second stage in which BEA students compete for the programs in which they were not admitted in the Regular process. Consider the residual graph $\hat{G}$ obtained after running the Regular process on the unified instance Γ^U. We note that all the nodes (c,a) where a is shortlisted for the BEA scholarship and $c <_a \mu_A(\Gamma^R)$ are c-dominated and can be removed from $\hat{G}$. This further reduces the graph to $\bar{G}$ in which every BEA student keeps only the programs in which he/she was not admitted in the Regular process. Then, the second stage process can be seen as equivalent to running the matching over the residual graph $\bar{G}$, this time with the regular quotas set to 0, $q_c^R = 0$ for all $c \in C$. Following a similar argument as in the previous case, we know that $\mu_A(\Gamma^B)(a) \leq_a \mu_A(\Gamma^U)(a)$ for all applicants $a \in A$. This shows that all BEA students are not worse off in $\mu_A(\Gamma^U)$ compared to μ_A^s.

A.1. Optimality

THEOREM 4. *The assignments μ_A and μ_C obtained from our procedure are FQ-matchings and are optimal for students and programs respectively among the set of FQ-matchings. Moreover,*

- *the allocation obtained from greedily assigning each students his top choice in G^*_A is equivalent to μ_A,*
- *the allocation obtained from greedily assigning each program its favorite students upon completing capacity in G^*_C is equivalent to μ_C.*

Proof (optimality): Let $\mathcal{M}$ be the set of nodes that belong to any FQ-matching. From Theorem 2 we know that the domination-free subgraph G^* constructed by deleting both a-dominated and c-dominated nodes contains all FQ-matchings, and therefore $\mathcal{M} \subseteq G^*$. Since we also know that $\mu_A = \mu_A(G^*, q)$ is a FQ-matching we get $\mu_A \subseteq \mathcal{M} \subseteq G^*$. Finally, since $\mu_A(G^*, q)$ assigns each student his top choice in G^*, we conclude that the top program for each applicant a is the same in $\mathcal{M}$ and G^*. Similarly, $\mu_C(G^*, q) = \mu^*_C \subseteq \mathcal{M} \subseteq G^*$, and since $\mu_C(G^*, q)$ assigns each program c its top q_c students, we have that the top q_c choices for any program c are the same in $\mathcal{M}$ and G^*. This yields in particular that $\hat{\mu}_C$ is an FQ-matching, concluding our proof.

The arguments for the other algorithms are essentially the same. Each of these algorithms computes a reduced subgraph G^* with $\mathcal{M} \subseteq G^*$ and the resulting student-optimal and/or university-optimal assignments are stable FQ-matchings so that they are contained in $\mathcal{M}$. Hence the top choices for applicants and/or programs are the same in $\mathcal{M}$ and G^*.

Appendix B: Notions of stability

As observed in Irving (1994) and Irving et al. (2000), when agents have non-strict preferences the notions of blocking pair and stability admit three natural extensions: weak, strong and super stability. For the case with no ties, it was shown that weakly stable matchings always exist but not necessarily the others.

Formally, a matching μ is

(a) *weakly stable* if there is no pair $(c, a) \in V \setminus \mu$ such that $c \rhd_a \mu(a)$ and $a \rhd_c \mu(c)$.

(b) *strongly stable* if there is no pair $(c, a) \in V \setminus \mu$ such that $c \trianglerighteq_a \mu(a)$ and $a \trianglerighteq_c \mu(c)$ with one of these preferences in the strict sense.

(c) *super stable* if there is no pair $(c, a) \in V \setminus \mu$ such that $c \trianglerighteq_a \mu(a)$ and $a \trianglerighteq_c \mu(c)$.

Clearly super stability implies strong stability which in turn implies weak stability, and the three concepts collapse to stability when preferences are strict. We show next that for FQ-matchings these three notions coincide.

THEOREM 5. *If an FQ-matching μ is weakly stable then it is super stable.*

Proof: Let μ be a weakly FQ-matching and suppose by contradiction that it is not super stable: there exists $(c, a) \in V \setminus \mu$ with $c \trianglerighteq_a \mu(a)$ and $a \trianglerighteq_c \mu(c)$. Since by assumption $\leq_a$ is a total order and $c \notin \mu(a)$, any $c' \in \mu(a)$ with $c' \leq_a c$ satisfies also $c' <_a c$ so that $c \rhd_a \mu(a)$. From weak stability it follows that $a \triangleright_c \mu(c)$ cannot hold, so that $|\mu(c)| \geq q_c$ and there exists $a' \in \mu(c)$ such that $a' \sim_c a$. By non-discrimination it follows that a should have been assigned to a program at least as good as c, which contradicts $c \triangleright_a \mu(a)$.

Appendix C: Equivalence between FQ-matchings and L-stable score limits

FQ-matchings turn out to be the same as the allocations obtained from L-stable score limits introduced in Biró and Kiselgof (2013). In their setting an applicant $a \in A$ is characterized by a set of integer scores s_c^a that determine the preferences of the programs. Note that in a college admission instance (G, q) we may define the score s_c^a as the rank[17] of student a in the preference list of the program c so that both settings are basically equivalent.

Given a set of score limits $l = (l_c)_{c \in C}$, Biró and Kiselgof (2013) define a corresponding assignment μ^l by letting $(c, a) \in \mu^l$ if c is the most preferred program of student a for which he attains the score limit, that is to say, $s_c^a \geq l_c$ and $s_{c'}^a < l_{c'}$ for all $c' >_a c$. Let $x_c^l = |\mu^l(c)|$ be the number of students assigned to program c. A score limit l is called *L-feasible* if for each program c with $|\mu^l(c)| \geq q_c$ we have $|\{(c, a) \in \mu^l : s_c^a > l_c\}| < q_c$, and is called *L-stable* if moreover any reduction of a score limit $l_c > 0$ leads to infeasibility.

L-feasibility is the analog of quotas-up-to-ties: a program may exceed its quota q_c but only to the extent that the last group of admitted students are tied with score equal to l_c. Since μ^l always satisfies non-discrimination and each student is assigned at most once, it follows that l is L-feasible if and only if μ^l is an FQ-matching. The following result establishes the connection between FQ-matchings and L-stable score limits.

THEOREM 6. *If l is an L-stable score limit then μ^l is a FQ-matching. Conversely, any FQ-matching can be expressed as $\mu = \mu^l$ for an L-stable score limit l.*

Proof: Let l be L-stable and suppose that μ^l has a blocking pair $(c, a) \notin \mu^l$. This means that c prefers a over some of its currently matched students $b \in \mu^l(c)$ so that $s_c^a > s_c^b \geq l_c$. Hence, a attains the score limit l_c and must have been assigned to c or better in μ^l, so that (c, a) cannot be a blocking pair. Therefore, μ^l is a FQ-matching.

Now let μ be a FQ-matching and let l_c be the rank s_c^a of the q_c-th student $a \in \mu(c)$, setting $l_c = 0$ if $|\mu(c)| < q_c$. We claim that $\mu = \mu^l$. Indeed, for each $(c, a) \in \mu$ we have that $s_c^a \geq l_c$ by definition of l_c, whereas stability implies $s_{c'}^a < l_{c'}$ for all $c' >_a c$, so that $(c, a) \in \mu^l$ and therefore $\mu \subseteq \mu^l$. Conversely, let $(c, a) \in \mu^l$. Then $s_c^a \geq l_c$ by non-discrimination a must be admitted to c or better. Since, moreover, $s_{c'}^a < l_{c'}$ for all $c' >_a c$ it must be the case that a is precisely matched with c and $(c, a) \in \mu$. This establishes the equality $\mu = \mu^l$. In particular μ^l is an FQ-matching which, as noted before, is equivalent to the fact that l is L-feasible. In order to show that l is L-stable let us consider a program with $l_c > 0$ and denote $\tilde{\mu}^l$ the assignment obtained after reducing l_c by one unit. The property $l_c > 0$ implies that the program had all its positions filled, namely $|\mu(c)| \geq q_c$ and $|\{(c,a) \in \mu^l : s_c^a > l_c\}| < q_c$. After reducing the score limit to $l_c - 1$ the set of students assigned to program c increases and L-feasibility fails since $|\{(c,a) \in \tilde{\mu}^l : s_c^a > l_c - 1\}| = |\mu(c)| \geq q_c$. This proves that l is L-stable completing the proof.

Appendix D: Implementation and complexity of the algorithms

The admission graph can be built in $O(|V|)$ operations, while removing a node can be done in constant time as it suffices to reassign the pointers and flags for its 4 adjacent nodes. We also need the following procedures for detecting strictly dominated nodes:

a-DOMINATION: For each program c scan the corresponding row in the graph from the top ranked student downwards, counting the students that place c at the top. When this counter reaches q_c continue scanning the row removing all nodes (c, a') which are strictly below in the order $<_c$.

c-DOMINATION: For each applicant a scan the corresponding column in the graph from the top program downward and stop as soon as a program c is found for which a is among the top q_c candidates. Then continue scanning the column removing all subsequent nodes (c', a) with $c' <_a c$.

Concerning the algorithm's complexity we already noted that initializing the admission graph takes $O(|V|)$. Since removing a node changes the top preferences for students and programs, the cycle must be repeated as long as strictly dominated nodes are found. However, each successful cycle removes at least one node so there are at most $|V|$ cycles, and in every iteration the procedures for detecting strict dominations run in $O(|V|)$ so that the overall complexity for the repeat is $O(|V|^2)$. When these procedures do not find any strictly dominated node we have the domination-free subgraph G^* and we proceed to compute μ_A and μ_C by greedily assigning the top remaining preferences. This final step takes $O(|V|)$ operations. We summarize this discussion in the following theorem.

THEOREM 7. *Our procedure computes FQ-matchings μ_A and μ_C in time $O(|V|^2)$.*

Note that the overall time spent in removing nodes is bounded by $O(|V|)$, and therefore the quadratic complexity comes from the search of these dominated nodes. In fact there are instances where this search takes indeed $O(|V|^2)$ operations. In AppendixF we present an alternative algorithm that improves this complexity.

In Appendix A (Theorem 4) we show that there is no need to drop all dominated nodes to obtain μ_A and μ_C. In fact, μ_A can also be obtained by greedily assigning each student to his top choice in the admission graph G_A^*, which is obtained by recursively dropping all a-dominated nodes. Similarly, μ_C can be obtained from G_C^*, which is obtained by erasing all c-dominated nodes. This can reduce the computation time if only one of these assignments is needed.

Appendix E: Additional examples

E.1. Example 1: Effect of allowing indifference in preferences

Table 6 compares the program assigned to each applicant for the case of strict preferences (first two columns) and when there is a single tie and flexible quotas (last two columns). In parenthesis we show the rank of the assigned program in the applicant's preference list.

Table 6 Matchings with strict vs unstrict preferences.

	Strict		Unstrict	
Applicant	μ_A	μ_C	μ_A	μ_C
a_1	$c_5(1)$	$c_4(2)$	$c_5(1)$	$c_4(2)$
a_2	$c_4(3)$	$c_4(3)$	$c_2(2)$	$c_4(3)$
a_3	$c_4(3)$	$c_5(4)$	$c_3(2)$	$c_5(4)$
a_4	-	-	$c_4(2)$	-
a_5	$c_3(2)$	$c_3(2)$	$c_4(1)$	$c_3(2)$
a_6	$c_3(3)$	$c_3(3)$	$c_2(2)$	$c_3(3)$
a_7	$c_2(4)$	$c_2(4)$	$c_3(3)$	$c_2(4)$
a_8	$c_2(2)$	$c_2(2)$	$c_1(1)$	$c_2(2)$
a_9	$c_1(2)$	$c_1(2)$	$c_2(1)$	$c_1(2)$

The assignment μ_A allowing for ties and flexible quotas benefits considerably the students: 7 applicants improve the order of the preference on which they are assigned, one moves from being unassigned to be selected in his second most desired program, and only one remains in the same program. This result is in line with Balinski and Sönmez (1999), who observed that increasing the quotas (in the fixed quota model) can never hurt a student under a student-optimal matching. In contrast, in this example the assignment μ_C does not change when we move from strict to non-strict preferences. This is not always the case and changes may occur if the university-optimal matching happens to have a tie in the last vacancy of some program, so that a student may shift to a different alternative inducing further shifts in a domino effect.

E.2. Example 2: Student-optimal FQ-matching is not monotone

Consider the admission graph of Figure 8 with program c_2 indifferent between a_4 and a_5. Note that (c_1, a_5) is strictly c-dominated and can be dropped. This gives G^* from which we obtain the matching μ_A (black nodes). As both a_4 and a_5 are tied in the last vacant, the student-optimal

Figure 8 Admission graph with student-optimal FQ-matching.

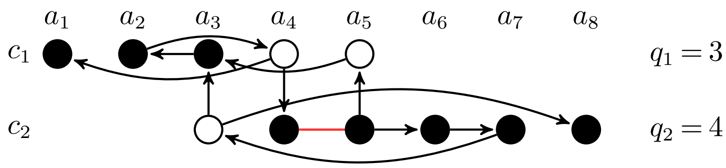


Figure 9 Non-monotone student-optimal FQ-matching.

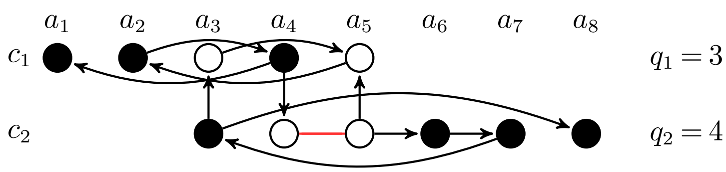
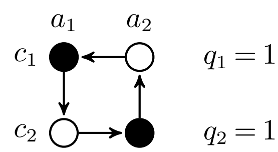


Figure 10 Admission graph with university-optimal FQ-matching.



matching μ_A with flexible quotas assigns both of them to program c_2 exceeding the quota by one unit.

Suppose now that a_5 improves its ranking for program c_1 so that $a_5 >_{c_1} a_3$, while everything else remains the same as shown in Figure 9. In this case node (c_1, a_3) is applicant dominated as there are 3 better ranked applicants whose first preference is c_1, and it is then removed. Nodes (c_2, a_4) and (c_2, a_5) are also applicant dominated as now the only remaining choice for a_3 is c_2. After cleaning all dominated nodes and assigning each student to his top remaining preference we obtain μ_A depicted by the black nodes in Figure 9. Thus, a_5 improved its ranking in c_1 but moved from being assigned in c_2 to be unassigned. This shows that the student-optimal mechanism μ_A is not monotone.

E.3. Example 3: University-optimal FQ-matching is not monotone

Consider the admission graph of Figure 10, where the black nodes represent the university-optimal assignment.

Suppose that a_2 decreases its ranking in c_2 so that $a_2 <_{c_2} a_1$. The new admission graph and the resulting university-optimal assignment are shown in Figure 11. Comparing both results we observe that a_2 moves from being assigned in c_2 (his second preference) to be matched in c_1 (his top preference). Thus, being worst ranked by c_2 helped him to improve his assignment, and therefore the university-optimal assignment is not applicant-monotone.

Figure 11 Non-monotone university-optimal FQ-matching.

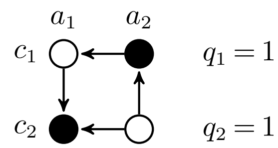


Figure 12 Student-optimal FQ-matching is not strategy-proof.

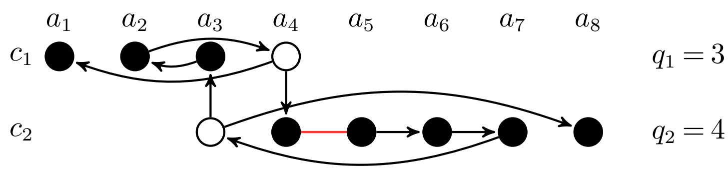
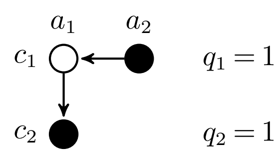


Figure 13 University-optimal FQ-matching is not strategy-proof.



E.4. Example 4: Student-optimal FQ-matching is not strategy-proof

In the admission graph of Figure 9 applicant a_5 was unassigned under μ_A. Suppose that this applicant lies when he states his preferences, applying only to program c_2. The resulting admission graph and μ_A are presented in Figure 12. We observe that in this case each student is assigned to his top preference under μ_A. In particular, by lying about his true preferences, a_5 moves from being unassigned to be matched in c_2. This shows that the student-optimal matching with flexible quotas is not strategy-proof.

E.5. Example 5: University-optimal FQ-matching is not strategy-proof

Consider the university-optimal matching μ_C in Figure 10 where both students are assigned to their second preference. If a_2 lies and only applies to c_1, the results returned by μ_A and μ_C are the same since node (c_1, a_1) is program dominated, and both students are assigned to their top choice. Thus c_2 can improve his assignment in μ_C by not revealing his true preferences.

E.6. Example 6: Unified admission process

Consider the admission graph in Figure 14 and assume that the students a_1 and a_4 are shortlisted for the scholarship. In this context, the quotas of program c are given by a pair (q_c^R, q_c^B), where q_c^R is the number of seats offered in the Regular process and q_c^B the vacancies offered in the BEA process.

In the sequential procedure the Regular process is run first with admission graph G^R in Figure 14 and quotas $q_c = q_c^R$, which gives the allocation[18] $\mu(\Gamma^R) = \{(c_1, a_4), (c_1, a_5), (c_2, a_2)\}$. Then, the

Figure 14 Original admission graph G^R.

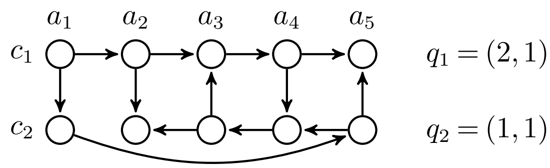


Figure 15 BEA admission graph G^B.

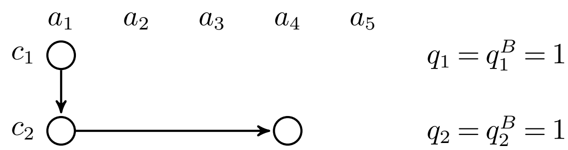
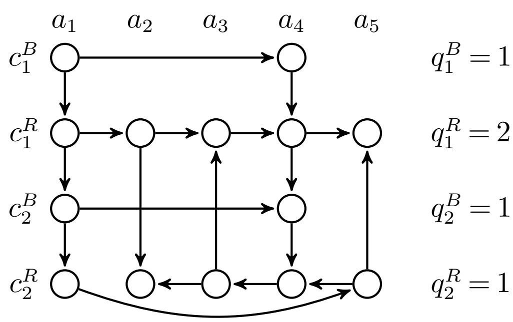


Figure 16 The unified admission graph G^U.



BEA admission process $\Gamma^B = (G^B, q^B)$ is built considering only the shortlisted students $A^B \subseteq A$ and the preferences where they were wait-listed in the Regular process. Continuing with the example, Figure 15 shows the corresponding graph G^B.

The resulting allocation for the BEA process is $\mu(\Gamma^B) = \{(c_1, a_1), (c_2, a_4)\}$, and therefore student a_4 is assigned to c_1 in the Regular process and to c_2 in the BEA process, while a_3 remains unassigned. Independently of which option is taken by a_4, a vacancy will be lost that could have been otherwise assigned to a_3.

The unified graph of this problem is shown in Figure 16. The allocation obtained from applying any of the two algorithms to the graph Γ^U is

$$\mu_A(\Gamma^U) = \mu_C(\Gamma^U) = \left\{(c_1^R, a_3), (c_1^R, a_5), (c_1^B, a_1), (c_2^R, a_2), (c_2^B, a_4)\right\}.$$

In this matching all applicants are assigned and no vacancies are lost. More importantly, every student is either equal or better off than in the sequential assignment.

Appendix F: Faster algorithm for FQ-matchings

An observation that might be exploited to improve the previous algorithms is that not all dominated nodes need to be removed. For instance, the greedy assignment μ_C computed from the original graph in Figure 1 assigns at most one program to each student and therefore it gives already a FQ-matching, without removing any of the c-dominated nodes in G. Also, for the student-optimal assignment μ_A not all a-dominated nodes will induce violations of the quotas-up-to-ties. A natural approach would then consist in dropping only those nodes that are causing these violations. This is similar to the strategy used in the deferred-acceptance algorithm of Gale and Shapley (1962).

We describe the idea for the student-optimal assignment μ_A. For each program c we set up an ordered list L_c in which we will sequentially add and remove applicants, with a counter $c.size$ to record the size of the list. Initially these lists are empty with $c.size = 0$. For each $a \in A$ we set a pointer $a.top$ to its most preferred program and, if this pointer is not null, we push a into a stack S that contains the applicants with no program assigned yet.

We iterate as follows. We pop a student a from the stack S and insert it in the ordered list L_c of the most preferred program c, increasing $c.size$ by one unit. If this counter exceeds q_c we check quotas-up-to-ties and eventually remove the last group of students in L_c to ensure that this property holds, by using the following procedure.

CHECK-QUOTAS-UP-TO-TIES: Find the set T_c of applicants tied in the last position in L_c. If $c.size \leq q_c + |T_c|$ we keep the list as it is, otherwise each node (c, b) for $b \in T_c$ is a-dominated so we remove b from L_c and update $b.top$ to its next most preferred program $c' <_b c$. If such c' exists we push b back into the stack S and otherwise we leave b unassigned. If the tie T_c is removed we update $c.size \leftarrow c.size - |T_c|$.

Algorithm 1 Fast student-optimal FQ-matching

read instance and build admission graph
initialize admission lists L_c and stack S
while (S is non-empty) **do**
    $a \leftarrow \text{pop}(S)$
    $c \leftarrow a.top$
    insert a into L_c and increase $c.size$ by one
    if ($c.size > q_c$) **then**
        CHECK-QUOTAS-UP-TO-TIES
    end if
end while
return assignment μ_A represented by the final lists L_c

To present our next result we let r_c be the largest size of a tie in the preorder $\leq_c$ so that $|\mu(c)| \leq q_c + r_c$ in any assignment satisfying quotas-up-to-ties. We denote $\bar{q}$ the maximum of the quantities $q_c + r_c$ and $\bar{r}$ the maximum of r_c.

THEOREM 8. *Algorithm 1 computes a FQ-matching μ_A in time $O((\bar{r} + \log \bar{q})|V|)$.*

Proof: To prove that the algorithm is finite we observe that each **while** cycle inserts a student into a new program, in decreasing order, so that the number of times a student may be reassigned is bounded by the number of nodes in its corresponding column in Γ. It follows that the algorithm terminates after at most $|V|$ cycles. Upon termination we have each student assigned to its most preferred program in the instance $\tilde{\Gamma}$ that contains all the nodes not removed during the execution, so that μ_A is the corresponding student-optimal assignment. Now, by construction the lists L_c satisfy quotas-up-to-ties throughout the algorithm, so that the computed μ_A is a FQ-matching for $\tilde{\Gamma}$. Since the nodes removed by the algorithm were all a-dominated, Theorem 2 guarantees that μ_A is also a FQ-matching for Γ.

In order to estimate the worst case complexity let us compute the number of basic operations per cycle. The insertion operation can be executed in time $O(\log |L_c|)$ which is bounded by $O(\log \bar{q})$ since L_c satisfy quotas-up-to-ties. The rest of the cycle deals with T_c which can have up to r_c elements, and therefore the number of operations involved is $O(\bar{r})$. Since there are at most $|V|$ cycles this yields the announced worst case complexity.

Appendix G: Enrollment

As we have discussed before, there exists the possibility that the inefficiencies generated by the double assignment could be resolved in the subsequent enrollment process. The intuition behind this is that after the assignment is announced, students must decide whether or not to enroll in the program they were assigned, and BEA students with double-assignment must choose only one program to enroll. Students who decide to enroll have three days to complete the process, which is called "First Period of Enrollment" (FPE). If a student decides not to enroll (either because the student has double assignment and/or opted for an outside option), his seat can be used by a wait-listed student. This can happen during the "Second Period of Enrollment" (SPE), which takes place right after the FPE. However, the SPE only lasts for two days, and therefore programs don't have much time to call wait-listed students and offer them admission. In addition, to speed up the enrollment process, universities announce more vacancies (normal vacancies plus overcrowd vacancies) than the number of students they want to enroll (normal vacancies). Every student has the right to enroll into their assigned program, but programs won't call wait-listed students unless the number of enrolled students is less than the normal vacancies. Hence, students who are not initially assigned are not guaranteed to be admitted into a program even if they are in the first place of the wait list and assigned students decide not to enroll during the FPE.

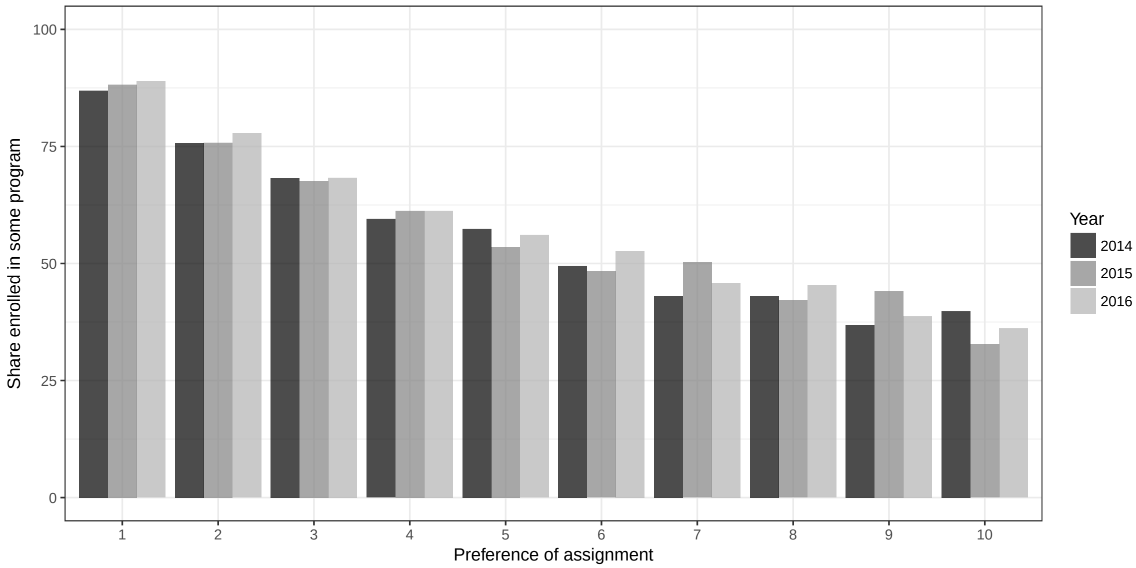
In Table 7 we show assignment and enrollment statistics for the years considered in our study (2014-2016). We observe that around 75% of those students who are admitted end up enrolling in a program that is part of the centralized system. Among those who enroll, we find that most of them do it in the program where they were assigned by the clearinghouse. However, around 3.5% of the students enroll in a program they prefer compared to the one assigned by the clearinghouse, mostly thanks to the SPE. Finally, we find that less than 1% enroll in a less preferred program. We don't have a clear explanation for this behavior.

Table 7 Enrollment - General

		2014	2015	2016
Admitted	Total	95,793	97,844	100,972
Enrolled	As assigned	72,006	74,030	75,909
	Better	2,664	2,706	2,676
	Worst	1,031	708	435
	Total	75,701	77,444	79,020

In Figure 17 we show how the probability of enrollment in any program (fraction of Enrolled Total and Admitted Total) decreases with the preference where the student was admitted. This illustrates that being assigned to a higher preference increases the probability of enrolling in a program that is part of the system, which in turn justifies the relevance of admitting students in a preference that is as good as possible for them.

Figure 17 Share of students enrolled by assignment preference



In terms of the impact of the unified assignment, in Table 8 we compare the results of the unified allocation (simulated in 2014-2015 and actual in 2016) with the actual enrollment decisions of students. The idea is to assess whether the enrollment process leads to the same results as the ones obtained with the unified allocation. We find that, among students who could have improved in their assignment if our policy were in place in 2014 and 2015, only a third of them get a better allocation in the enrollment process. In contrast, almost all of them get enrolled in a better assignment in 2016 (when our policy was in place). This suggests that the inefficiencies generated by the double assignment are not completely solved in the enrollment process[19].

Table 8 Enrollment

		2014	2015	2016
Improved	Admitted - Total	1,737	1,915	1,749
	Enrolled - As assigned	431	469	1,392
	Enrolled - Better	44	38	34
	Enrolled - Worst	802	909	13
	Enrolled - Total	1,277	1,416	1,439
New admitted	Admitted - Total	568	672	777
	Enrolled - As assigned	146	172	446
	Enrolled - Better	20	38	17
	Enrolled - Worst	10	15	0
	Enrolled - Total	191	249	467

Endnotes

1. In Chile, students apply directly to a major in a given university, such as Medicine in the University of Chile. We refer to program as a pair major-university.

2. In addition to what we describe in this paper, each university has special admission programs such as for athletes, racial minorities, among others. In addition, there are other centralized admission tracks that were added to the system in 2017 that we don't address in this paper for simplicity.

3. BEA students are indifferent between regular and reserve seats because they obtain the scholarship regardless of how they were admitted, and there are no differences between these types of vacancies.

4. The *Consejo de Rectores de las Universidades Chilenas* (CRUCH) is the institution that gathers these universities and is responsible to drive the admission process, while DEMRE is the organism in charge of applying the selection tests and carrying out the assignment of students to programs.

5. Some programs such as music, arts and acting, may require additional aptitude tests.

6. Respecting some basic criteria defined by CRUCH.

7. To be more precise, the system differentiates between normal vacancies and overcrowd vacancies. During the enrollment process, if an admitted student does not enroll then wait-listed students are offered admission only up to the normal vacancies. Hence, only those students who were admitted before the enrollment process can use overcrowd vacancies.

8. This was directly translated from the document "Normas, Inscripcion y Aspectos Importantes del Proceso de Admision, 2013" CRUCH (2013), page 8.

9. For instance, when we start collaborating in 2013 there were 1,162 students with double assignment, which is equivalent to around 10% of the total number of BEA students that are admitted.

10. In Appendix G we compare our results using enrollment data and we show that the inefficiencies introduced by the double assignment are not solved in the subsequent enrollment process.

11. This example is a small variation of the example in Figure 1, where program c_2 is indifferent between a_2 and a_6.

12. In Appendix D we show that the complexity of this procedure is $O(|V|^2)$.

13. As long as we consider applications as fixed, allowing for ties and flexible quotas will weakly increase the number of vacancies per program, resulting in a Pareto improvement for students.

14. In a merit based admission system tie breaking rules that rely on randomness can be controversial and perceived as unfair from the perspective of students.

15. For example, the University of Chile requires applicants to apply to at most 4 programs, and these applications must be listed within the top 4 positions in the applicant's list.

16. In Figure 17 (see Appendix G) we show that the share of students that enroll after being assigned in one of their 10 listed preferences is decreasing in the number of assigned preference.

17. The students in the least preferred group have rank 1, the next group is ranked 2, and so on.

18. In this example the student-optimal and the university-optimal algorithms return the same allocation.

19. To be able to fully measure the impact of our policy change in the enrollment process, we would have to structurally model the application and enrollment behavior of students. However, the previous statistics are strong evidence that the enrollment process does not fully solve the inefficiencies generated by the double assignment.

Acknowledgments

This research was supported by ICM/FIC P10-024F *Núcleo Milenio Información y Coordinación en Redes.* We gratefully acknowledge the support of the *Departamento de Evaluación, Medición y Registro Educacional* (DEMRE) for providing essential information and data without which the present study could not have been completed. We also express our thanks to José Correa and Nicolás Figueroa for their useful advice and support. Finally, we would like to thank Eduardo Azevedo, Hanming Fang, Greg Macnamara, Javier Martínez de Albéniz, Rakesh Vohra and Xavier Warnes for their constructive feed-back.

References

Abdulkadiroğlu A (2007) Controlled School Choice. *New York* 2006(April):1–44.

Abdulkadiroglu A, Pathak PA, Roth AE, Slon T (2005) The Boston Public School Match. *American Economic Review* 95(2):368–371.

Ashlagi I, Kanoria Y, Leshno JD (2017) Unbalanced Random Matching Markets: The Stark Effect of Competition. *Journal of Political Economy* 125(1):69–98, ISSN 0022-3808.

Azevedo EM, Budish E (2017) Strategy-proofness in the Large (23771).

Baïou M, Balinski M (2004) Student admissions and faculty recruitment. *Theoretical Computer Science* 322:245–265.

Balinski M, Sönmez T (1999) A tale of two mechanisms: student placement. *Journal of Economic Theory* 84(1):73–94.

Biró P, Kiselgof S (2013) College admissions with stable score-limits. *Central European Journal of Operations Research* 1 – 15.

Canales A, De los Ríos D (2007) Factores explicativos de la deserción universitaria. *Revista Calidad en la Educación* 26:171–201.

CRUCH (2013) Proceso de admision 2013: Normas, inscripcion y aspectos importantes del proceso de admision. Diario El Mercurio.

Dur U, Duke S, Parag K, Pathak PA, Sönmez T (2016a) Reserve Design: Unintended Consequences and The Demise of Boston's Walk Zones. *Journal of Political Economy* .

Dur U, Kominers SD, Pathak PA, Sönmez T (2013) Priorities vs. Precedence in School Choice: Theory and Evidence from Boston .

Dur U, Pathak PA, Sönmez T (2016b) Explicit Vs. Statistical Preferential Treatment in Affirmative Action: Theory and Evidence From Chicago'S Exam Schools. *NBER Working Paper Series* .

Echenique F, Yenmez MB (2012) How to Control Controlled School Choice. *SSRN Electronic Journal* 105(8):1–45, ISSN 1556-5068.

Ehlers L, Hafalir IE, Yenmez MB, Yildirim MA (2014) School choice with controlled choice constraints: Hard bounds versus soft bounds. *Journal of Economic Theory* 1–42.

Ergin H, Somnez T (2006) Games of School Choice under the Boston Mechanism. *Journal of Public Economics* 90:215–237.

Gale D, Shapley LS (1962) College admissions and the stability of marriage. *American Mathematical Monthly* 69(1):9–15.

Hafalir IE, Yenmez MB, Yildirim MA (2013) Effective Affirmative Action in School Choice. *Theoretical Economics* 8:325–363.

Irving RW (1994) Stable marriage and indifference. *Discrete Applied Mathematics* 48:261–272.

Irving RW, Manlove DF, Scott S (2000) The hospitals/residents problem with ties. *Algorithm Theory - SWAT 2000*, volume 1851 of *Lecture Notes in Computer Science*, 259–271 (Springer Berlin Heidelberg), ISBN 978-3-540-67690-4.

Kamada Y, Kojima F (2015) Efficient matching under distributional constraints: Theory and applications. *American Economic Review* 105(1):67–99, ISSN 00028282.

Kamiyama N (2017) Strategic issues in college admissions with score-limits. *Operations Research Letters* 45(2):105 – 108, ISSN 0167-6377.

Kojima F (2012) School choice: Impossibilities for affirmative action. *Games and Economic Behavior* 75(2):685–693, ISSN 08998256.

Lara B, Meller P, Valdés G (2017) Life-Cycle Valuation of Different University Majors. Case Sudy of Chile 42(6).

Larroucau T, Mizala A, Ríos I (2015) The effect of including high school grade rankings in the admission process for chilean universities. *Pensamiento Educativo. Revista de Investigación Educacional Latinoamericana.* 52(1):95–118.

Roth AE, Peranson E (2002) The redesign of the matching market for american physicians: Some engineering aspects of economic design. *American Economic Review* 89:748–780.

Shapley L, Scarf H (1974) On Cores and Extensibility. *Journal of Mathematical Economics* 1:23–37.