Robust Admission Via Two-Stage Stable Matching Under Ranking Uncertainty

Gustavo Angulo[*] Matías Giddings[†] Pablo Marshall[‡]

Abstract

We study a two-stage admission and assignment problem under uncertainty arising in university admission systems. In the first stage, applicants are admitted to an initial two-year program. In the second stage, admitted applicants are assigned to degree programs through an articulation mechanism subject to capacity constraints. Uncertainty stems from the academic performance of admitted applicants within the initial program, as this exit ranking determines the sequence in which students are processed by the articulation mechanism. Admission decisions must therefore be robust to all possible exit orders, giving rise to a robust stable matching problem.

We introduce a notion of robust compatibility for sets of admitted applicants and establish some of its properties. Building on a standard integer programming formulation of stable matching, we derive an adversarial formulation that determines whether a given admission set remains compatible under worst-case realizations of the exit order. This adversarial model underpins a sequential admission algorithm that incrementally constructs a robustly compatible set of applicants. Computational experiments on real admission data from three consecutive cohorts show that computational effort is highly unevenly distributed across iterations. In particular, certifying acceptance decisions is substantially more expensive than certifying rejections, with overall running time dominated by a small number of late iterations in highly congested regimes. We also address how the algorithm has been applied in practice and highlight the benefits in first-year retention rate driven by the proposed mechanism.

Keywords: OR in Education, Robust Stable Matching, College Admission, Uncertain Preferences

[*]Corresponding author: gangulo@uc.cl, School of Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile 7820436.

[†]matias.giddings@uc.cl, School of Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile 7820436.

[‡]pmarshall@uc.cl, School of Management and Bachillerato Inicia, Pontificia Universidad Católica de Chile, Santiago, Chile 7820436.

1 Introduction

The Pontificia Universidad Católica de Chile (PUC) has more than 30,000 undergraduate students and, according to various international rankings, is considered one of the leading universities in Latin America and among the top 100 universities worldwide. The undergraduate admissions process at PUC is carried out through a centralized system that includes nearly 50 Chilean universities (Bordon and Fu (2015); Ríos et al. (2021); DEMRE (2026)).

In line with its institutional mission, PUC has developed several programs over the past 15 years aimed at increasing the inclusion of students with educational gaps while maintaining high academic standards. Since 2010, multiple alternative admission pathways have been implemented to facilitate access to PUC's academic programs for students from low socioeconomic backgrounds, graduates of public secondary schools, Indigenous peoples, migrants, and individuals with disabilities. In addition, PUC has adhered to inclusion programs developed by the Chilean government, such as those coordinated by the Ministry of Education and DEMRE, thereby contributing to the strengthening of equity-oriented policies in access to higher education in Chile (Ríos et al. (2021); MINEDUC (2026a)). These initiatives have led to a sustained increase in the enrollment of students admitted through equity-based pathways in recent years, consolidating the university's institutional commitment to a more inclusive and diverse higher education institution.

In 2024, PUC developed a new academic program, *Bachillerato Inicia*, aimed exclusively at students from technical-professional secondary education. In Chile, as in many Latin American and European countries, upper secondary students can choose, during their final two years of schooling, between an academically oriented track designed to prepare them for higher education and a technical-professional track that prepares students for the labor market in specific technical occupations (Salas-Velasco (2024)). In Chile, the proportion of students graduating from technical-professional secondary education in 2024 represented 27% of total upper secondary graduates (MINEDUC (2026b)), and a growing share of these students aspire to continue into higher education. After two years of academic preparation, students enrolled in *Bachillerato Inicia* articulate their studies into the regular undergraduate degree programs offered by PUC.

The *Bachillerato Inicia* program has a special admissions process that considers students' secondary school grade point average as well as specialized assessments of intra- and interpersonal skills. This selective process is designed to identify students with high academic potential from technical-professional backgrounds and to facilitate their transition into university-level studies, thereby strengthening equity in access to higher education. The *Bachillerato Inicia* program allocates a fixed number of articulation places for entry into PUC's regular undergraduate degree programs. Although the total number of available articulation places is approximately 2.4 times the number of students selected into the program, some articulation programs are in particularly high demand among applicants, substantially exceeding the number of places available for those specific programs.

This phenomenon of excess demand for certain programs is common in Chile in two-year academic preparation programs that subsequently allow students to articulate into regular undergraduate degrees at the host university. These programs typically operate with limited places or vacancies and experience significant excess demand in specific fields. As a result, many students become dissatisfied with the exit programs to which they are assigned and choose to restart their educational trajectories the following year, thereby losing time and resources. For example, in the 2024 admissions cycle, the Chilean university system offered 39 two-year academic programs with characteristics similar to *Bachillerato Inicia*. First-year student retention rates in these programs were the lowest in the system, at approximately 40%, compared to 84% for the Chilean university system as a whole (MINEDUC (2026c)). The issue of excess demand for certain programs or majors could also extend to the U.S. higher education system, in which students typically select a college and subsequently a major, and where competitive access to high-demand majors similarly generates bottlenecks and student dissatisfaction (Bordon and Fu (2015)).

At the time of application to *Bachillerato Inicia*, students submit a preliminary declaration of three academic programs, ranked according to their preferences for subsequent articulation. The purpose of this preference declaration is to support a selection process that balances students' preferences with the availability of articulation places. For the 2026 cohort, 248 applicants advanced to the final stage of the *Bachillerato Inicia* selection process, while the program had 125 available places. This implies that, at the final selection stage, nearly one in two applicants is selected. At the end of the first year, students in this cohort will be able to choose among 51 different academic programs, with a total of 305 articulation places. If the 125 students were selected strictly according to the admission ranking, constructed based on secondary school performance and specialized assessments of intra- and interpersonal skills, the first preferences of selected students would be concentrated in 29 of the 51 available academic programs.

The purpose of this study is to develop a selection model for admission to the *Bachillerato*

Inicia program that balances students' preferences with the availability of places in the university's regular undergraduate degree programs. The objective is to ensure that all selected students are assigned, at the end of the first year, to one of the three academic programs declared in their initial preference list. Critically, academic performance ranking at the end of the first year determines which of the three declared preferences each student can access. This requirement adds complexity to admission planning, as considering all possible exit-ranking scenarios is computationally and practically infeasible. Furthermore, the absence of historical data or robust predictive models exacerbates this complexity, making it difficult to reliably estimate future academic performance and outcomes for admitted students.

By leveraging tools from stable matching (Gale and Shapley (1962)) and integer programming (IP), we frame the admission problem faced by *Bachillerato Inicia* as a robust two-stage stable matching problem under preference uncertainty, leading to the development of an algorithm that has been successfully applied in the 2024, 2025, and 2026 admission cycles. More precisely, our contributions are:

1. We introduce and formalize a class of robust two-stage college admission problems, where programs share a common yet uncertain preference profile over students.

2. We introduce a notion of robust compatibility among students and study some of its properties.

3. We develop an acceptance/rejection algorithm based on an adversarial IP formulation of the stable matching problem.

4. We conduct numerical experiments on the 2024-2026 datasets and assess the actual implementation of the algorithm by *Bachillerato Inicia*.

The remainder of the paper is organized as follows. In Section 2, we review the relevant literature on stable matching under uncertainty. In Section 3, we formalize the robust admission problem and the concept of robust stability, with its properties studied in Section 4. Then, in Section 5, we develop an IP-based algorithm to test robust compatibility of a given set of students. Numerical results are presented in Section 6, while the implementation is discussed in Section 7. Finally, conclusions are drawn in Section 8.

2 Literature review

The design of centralized admission mechanisms is grounded in the theory of stable matching, initiated by the seminal work of Gale and Shapley (1962). They introduced the concept of stability, defined as the absence of blocking pairs, and the Deferred Acceptance (DA) algorithm, which guarantees finding a stable matching. This framework was established as the canonical theoretical foundation for two-sided matching markets by Roth and Sotomayor (1990) and later adapted to school choice by Abdulkadiroğlu and Sönmez (2003), who demonstrated that student-optimal stable mechanisms eliminate justified envy and provide strategy-proofness for students. While these classical models assume that agent preferences and institutional priorities are fixed and known, recent literature has increasingly focused on environments characterized by uncertainty and dynamic changes.

Early departures from the deterministic model focused on incomplete information regarding agent preferences. Roth (1989) showed that stability and strategy-proofness are generally incompatible when preferences are private. This spurred a line of research investigating stability concepts under various information structures, ranging from Bayesian settings to informationally constrained environments (Chakraborty et al., 2010; Liu et al., 2014; Ehlers and Massó, 2015; Bikhchandani, 2017). In these environments, the primary challenge is often incentivizing truthfulness or defining stability when agents cannot perfectly identify blocking pairs. Our setting differs as the uncertainty is not about hidden preferences, but about future priorities (rankings) that simply do not exist at the time of the initial decision.

A critical distinction in modern matching literature is between robustness against strategic manipulation and robustness against structural perturbations. Kojima (2011) formalized robust stability as immunity to combined manipulation strategies (misreporting followed by blocking) and showed that such robustness is achievable if and only if priority structures are acyclic. Afacan (2012) extended this notion to group deviations and established an impossibility result: no mechanism can satisfy group robust stability, even under acyclic priorities, motivating weaker robustness notions. In contrast, Mai and Vazirani (2018) and more recently Gangam et al. (2026) study robust stable matchings from a structural perspective, seeking matchings that remain stable across multiple realized instances of preferences. Gangam et al. (2026) demonstrate that while the set of robust matchings retains a lattice structure under simple local changes, this structure collapses under general, multi-agent perturbations. These results indicate that tractability can deteriorate under broad uncertainty, motivating optimization-based approaches when seeking robust solutions. Other structural approaches include the work of Genc et al. (2017) on supermatches that are resilient to breakups, and Chen et al. (2021) on d-robustness against swap perturbations. Furthermore, Aziz et al. (2020) showed that verifying stability guarantees can be NP-complete under uncertain preferences.

Our work is most closely related to the emerging field of two-stage stable matching, where decisions are split temporally. Gajulapalli et al. (2020) and Bredereck et al. (2020) study how to adapt stable matchings to evolving preferences or new arrivals without disrupting existing assignments. Most notably, Faenza et al. (2024, 2025) have recently introduced a framework for two-stage stochastic stable matching. Their model optimizes a first-stage matching to minimize the expected cost of recourse (reassignment) after uncertainty is revealed, reducing the problem to minimum cuts in rotation digraphs.

However, our problem differs fundamentally from these stochastic approaches. While Faenza et al. (2024) focus on minimizing expected disutility under uncertainty, we address a problem of robust feasibility. In the *Bachillerato Inicia* context, we cannot rely on recourse or expectation; we must select a set of students in the first stage such that a stable assignment is guaranteed in the second stage for every possible realization of the exit ranking (an adversarial or worst-case approach). We do not seek a single matching that is stable across scenarios, but rather a compatible admission set that allows for some stable matching to exist regardless of how the priority order permutes. This constraint of zero-recourse feasibility under global priority uncertainty defines a new class of robust two-stage stable matching problems that, to our knowledge, has not been explicitly studied in the literature.

3 Problem description

Consider a set of applicants S seeking admission to an initial preparatory program, which after a first stage of study allows students to articulate into a set of degree programs P. Each applicant $s \in S$ has declared a strictly ordered subset of programs $P_s \subseteq P$, representing their preferences. Preferences are denoted by $\rhd_s$, where $q \rhd_s p$ means that applicant s prefers program q over p. Conversely, each program $p \in P$ has a set of interested applicants $S_p = \{s \in S : \; p \in P_s\}$ and a predetermined number of available slots c_p. We assume that students prefer any program in their list to be left without a seat, denoted by $\emptyset$, and thus for all $s \in S$ we extend $\rhd_s$ to $P_s \cup \{\emptyset\}$ so that $p \rhd_s \emptyset$ for all $p \in P_s$. Similarly, we assume that any program $p \in P$ prefers to fill its slots with any student in S_p rather than leaving empty seats.

First-stage problem (ex-ante admission). The objective is to determine a set $S^* \subseteq S$ of applicants to be admitted into the initial program, while guaranteeing that all admitted applicants can articulate for any realization of the exit ranking in the ex-post problem. In this particular case, the preparatory program is interested in admitting as many applicants as possible, giving priority to those with higher entrance ranking.

Second-stage problem (ex-post articulation). Consider the set of admitted applicants $S^* \subseteq S$ who have completed the first year of the initial program and must be assigned (articulated) to one of their declared undergraduate programs. An articulation mechanism proceeds as follows: first, students in S^* are ranked according to their exit ranking, represented by a strict priority order $\blacktriangleright$. Students are reviewed sequentially, starting from the highest-ranked student. For each student s, programs in their preference list P_s are checked in descending order of priority $\rhd_s$. The student is assigned to the highest-priority program that still has available slots. Once a program reaches its capacity, it no longer accepts additional students.

Definition 1. For any $S^* \subseteq S$, the set of all strict total orders $\blacktriangleright$ over S^* is denoted by $\mathcal{R}(S^*)$.

We interpret any $\blacktriangleright \in \mathcal{R}(S^*)$ as an exit ranking or common preferences over S^* used by all programs in the articulation mechanism.

Definition 2. We say that $S^* \subseteq S$ is compatible with exit ranking $\blacktriangleright \in \mathcal{R}(S^*)$ if all students are matched to a preferred program by the articulation mechanism.

Definition 3. We say that $S^* \subseteq S$ is robustly compatible if, for all exit ranking $\blacktriangleright \in \mathcal{R}(S^*)$, S^* is compatible with $\blacktriangleright$.

Example 1 (Robust vs. non-robust compatibility)**.** *Consider three programs $P = \{A, B, C\}$ with capacities $(c_A, c_B, c_C) = (2, 2, 1)$, and an admitted cohort $S^* = \{1, 2, 3, 4, 5\}$. Each student $s \in S^*$ has submitted a strict preference ranking $\rhd_s$ over $P_s = P$, shown in Table 1.*

We examine how different realizations of the exit ranking affect the compatibility of the same admitted set.

Exit ranking α : $2 \blacktriangleright 4 \blacktriangleright 5 \blacktriangleright 1 \blacktriangleright 3$. *Students 2 and 4 are processed first and occupy the two available seats in program A. Student 5 is then assigned to program B. When student 1 is considered, program A is already full, so she is assigned to her second choice, B, completing its capacity. Finally, student 3 is assigned to the only remaining seat in program C. Thus, under ranking α, all students in S^* are successfully articulated, and S^* is compatible with α.*

s	$\rhd_s$
1	$A \rhd B \rhd C$
2	$A \rhd B \rhd C$
3	$C \rhd B \rhd A$
4	$A \rhd C \rhd B$
5	$B \rhd A \rhd C$

Table 1: Applicants' preference rankings.

Exit ranking β : $2 \blacktriangleright 4 \blacktriangleright 5 \blacktriangleright 3 \blacktriangleright 1$*. The first three assignments coincide with those under* α*: program* A *becomes full, and one seat in* B *is taken. However, student 3 now appears earlier and claims the only seat in program* C*. When student 1 is processed last, all programs in her preference list are already at capacity. She therefore remains unmatched, and the same cohort* S^* *is incompatible with ranking* β*.*

This example illustrates that compatibility of an admitted set may depend on the realized exit ranking. Indeed, while S^* *is compatible with some rankings, it fails to be compatible with others. By contrast, if the admitted set is reduced to* $S^* = \{1, 2, 3, 4\}$*, then all four students can be matched for every one of the* $4!$ *possible exit rankings. Hence,* $\{1, 2, 3, 4\}$ *is robustly compatible, whereas admitting student 5 violates robustness.*

It can be observed that the articulation mechanism corresponds to a simple serial dictatorship, which yields a unique assignment given $S^* \subseteq S$ and exit ranking $\blacktriangleright \in \mathcal{R}(S^*)$. Moreover, such an assignment is a stable matching. This connection allows us to draw tools from the rich literature on the subject. In the following section, we formalize the model.

4 Definitions and basic properties

Definition 4. A pair $(s, p) \in S \times P$ is admissible if $p \in P_s$, or equivalently in our case, if $s \in S_p$. The set of admissible pairs is denoted by A, and for $S^* \subseteq S$, its restriction to $S^* \times P$ is denoted by $A(S^*)$.

Definition 5. For $\mu \subseteq A$, we write $\mu(s) = \{p \in P : (s, p) \in \mu\}$ for $s \in S$ and $\mu(p) = \{s \in S : (s, p) \in \mu\}$ for $p \in P$. A matching is a set $\mu \subseteq A$ such that $|\mu(s)| \leq 1$ for all $s \in S$ and $|\mu(p)| \leq c_p$ for all $p \in P$. If $\mu(s) = \{p\}$, we simply write $\mu(s) = p$. For $S^* \subseteq S$, an S^*-matching is a matching $\mu \subseteq A(S^*)$.

Definition 6. Given an S^*-matching $\mu \subseteq A(S^*)$ and $\blacktriangleright \in \mathcal{R}(S^*)$, a pair $(s,p) \in A(S^*) \setminus \mu$ blocks μ under $\blacktriangleright$ if (i) $\mu(s) = \emptyset$ or $p \rhd_s \mu(s)$ and (ii) $|\mu(p)| < c_p$ or $|\{t \in \mu(p) : \ s \blacktriangleright t\}| \geq 1$. If μ has no blocking pair under $\blacktriangleright$, μ is a $\blacktriangleright$-stable matching.

In other words, a matching μ is not $\blacktriangleright$-stable if there exists an admissible pair (s,p), unmatched to each other by μ, such that s prefers to be matched to p (either because she is unmatched or matched to a program she prefers less than p) and p is willing to accept s (either because it has empty slots or one of them is occupied by another student it prefers less than s under $\blacktriangleright$).

In general, there can be multiple stable matchings when programs have different priorities over students. However, the situation simplifies substantially when programs share a common preference profile as in our case. Proposition 2 will be instrumental in our developments.

Proposition 2 (Irving et al. (2008)). *For any $S^* \subseteq S$ and $\blacktriangleright \in \mathcal{R}(S^*)$, the assignment given by the articulation mechanism is the only possible $\blacktriangleright$-stable matching.*

Proposition 3 below shows that the collection of robustly compatible sets is down-monotone or closed under inclusion. This property ensures that if an admitted student declines to enroll, a situation that can occur in actual admission processes, the remaining admitted students still form a robustly compatible set.

Proposition 3. *Let $S', S^* \subseteq S$ be such that $S' \subseteq S^*$ and S^* is robustly compatible. Then, S' is robustly compatible.*

Proof. Fix an arbitrary exit ranking $\blacktriangleright' \in \mathcal{R}(S')$ and consider any extension $\blacktriangleright^* \in \mathcal{R}(S^*)$ such that $s \blacktriangleright^* t$ for all $s \in S'$ and $t \in S^* \setminus S'$. Note that the execution of the articulation mechanism over S' under $\blacktriangleright'$ is equivalent to halting its execution over S^* under $\blacktriangleright^*$ once all students in S' are processed. Since S^* is robustly compatible, the latter matches all students in S^*, in particular, those in S'. Since $\blacktriangleright' \in \mathcal{R}(S')$ was arbitrary, S' is robustly compatible. $\square$

Propositions 5 and 4 below show that as students gain more flexibility in choosing and articulating to programs, robust compatibility is not lost.

Proposition 4. *Let $S^* \subseteq S$ be robustly compatible. If students extend their list of preferred programs, S^* remains robustly compatible.*

Proof. Let $\blacktriangleright \in \mathcal{R}(S^*)$. If all students in S^* are matched by the articulation mechanism under $\blacktriangleright$, none of them will be matched to a new preferred program added at the end of their lists.

Therefore, the articulation mechanism yields the same matching and, in particular, S^* remains compatible with $\blacktriangleright$. Since this holds for any $\blacktriangleright \in \mathcal{R}(S^*)$, S^* remains robustly compatible. $\square$

Proposition 5. *Let $S^* \subseteq S$ be robustly compatible. If the number of slots increases, S^* remains robustly compatible.*

Proof. Let $\blacktriangleright \in \mathcal{R}(S^*)$. It is known that the student-optimal stable matching cannot harm any student when slots increase (Roth and Sotomayor (1990)). By Proposition 2, the uniqueness of the stable matching implies it is student-optimal. Therefore, if all students in S^* are matched by the articulation mechanism under $\blacktriangleright$, they remain matched when slots increase, that is, S^* remains compatible with $\blacktriangleright$. Since this holds for any $\blacktriangleright \in \mathcal{R}(S^*)$, S^* remains robustly compatible. $\square$

We close this section with Proposition 6, which will allow us to strengthen the IP formulation to be presented in Section 5.

Proposition 6. *Let $S^* \subsetneq S$ be robustly compatible and let $s \in S \setminus S^*$. Then, for every $\blacktriangleright \in \mathcal{R}(S^* \cup \{s\})$, the articulation mechanism over $S^* \cup \{s\}$ leaves at most one student unmatched.*

Proof. Fix an arbitrary exit ranking $\blacktriangleright \in \mathcal{R}(S^* \cup \{s\})$ and let $\hat{\mu} \subseteq A(S^* \cup \{s\})$ be the unique $S^* \cup \{s\}$-matching produced by the articulation mechanism under $\blacktriangleright$. Let $\mu \subseteq A(S^*)$ be the S^*-matching produced by the articulation mechanism under the restriction of $\blacktriangleright$ to S^*. By robust compatibility of S^*, μ matches all students in S^*. Let u be the number of unmatched students produced by $\hat{\mu}$, i.e., $u = |\{t \in S^* \cup \{s\} : \hat{\mu}(t) = \emptyset\}|$.

We compare the two executions of the mechanism (on S^* and on $S^* \cup \{s\}$) along the common order $\blacktriangleright$. Consider the first student $t_1 \in S^*$ in the order $\blacktriangleright$ having different assignments between the two executions. If no such student exists, then all students in S^* receive the same program under μ and $\hat{\mu}$. Hence, $\hat{\mu}$ matches all students in S^* and $u \leq 1$ trivially. If t_1 exists, note that necessarily $s \blacktriangleright t_1$. Since t_1 is matched under μ, she has some assigned program $p_0 = \mu(t_1) \in P_{t_1}$ when only students in S^* are considered. Because t_1 is the earliest student whose assignment differs, t_1 must find p_0 full in the execution on $S^* \cup \{s\}$ at the moment she is processed, and moreover, her seat in p_0 must have been taken by s. Now, in the execution on $S^* \cup \{s\}$, student t_1 either becomes unmatched, or she is assigned to some program $p_1 \in P_{t_1}$ such that $p_0 \rhd_{t_1} p_1$. If t_1 becomes unmatched, after processing t_1, all programs have the same number of remaining slots in both executions (because the only change has been p_0 accepting s instead of t_1). Therefore, from that point on, both executions coincide and thus $u = 1$. Otherwise, t_1 is assigned to $p_1 \neq p_0$, consuming one seat of p_1 that, in the execution on S^*, was either left empty or allocated later to some student of S^*. If it was left empty in the execution on S^*, then this reassignment does not change the assignment of remaining students, so $u = 0$ in this case. If it was allocated later to some student, then there exists a set of students whose assignments are modified as a consequence, because the displaced student must move to another acceptable program, and so on.

Formally, define a sequence of distinct students $t_1, t_2, \dots$ in S^* as follows: for $i \geq 1$, having defined t_i and its new assignment $p_i := \hat{\mu}(t_i)$, let t_{i+1} be the earliest student (in the order $\blacktriangleright$) whose assignment under μ equals p_i but who cannot be assigned to p_i in the execution on $S^* \cup \{s\}$ due to the seat taken by t_i. If no such student exists, the sequence terminates. By construction, the mechanism differs from μ only along this single chain: each t_i takes (in the enlarged instance) a seat that was previously used (in the restricted instance) by t_{i+1}, and all other students in S^* keep their original assignment. Since the chain is triggered by the insertion of a single student s, the chain can terminate in one of two ways: (i) some t_i is reassigned to a program that had an empty seat in the restricted instance and no student becomes unmatched, or (ii) the last student in the chain has no acceptable program with remaining capacity and becomes unmatched. In either case, at most one student can end up unmatched and thus $u \leq 1$. Since $\blacktriangleright \in \mathcal{R}(S^* \cup \{s\})$ was arbitrary, the result holds for every exit ranking. □

5 An optimization problem for robust compatibility

Consider the second-stage problem over students $S^* \subseteq S$ and exit ranking $\blacktriangleright \in \mathcal{R}(S^*)$. For each admissible pair $(s, p) \in A(S^*)$, let x_{sp} be a binary decision variable indicating whether student s is assigned to program p. Let $X(S^*)$ be the set of vectors x that represent an S^*-matching, that is, that satisfy

$$\sum_{p \in P_s} x_{sp} \leq 1 \quad \forall s \in S^* \tag{1}$$

$$\sum_{s \in S_p^*} x_{sp} \leq c_p \quad \forall p \in P \tag{2}$$

$$x_{sp} \in \{0, 1\} \quad \forall (s, p) \in A(S^*), \tag{3}$$

where $S_p^* = S_p \cap S^*$. Constraints (1) and (2) ensure each student is assigned to at most one program and program capacities are respected, respectively, while (3) impose binary constraints. Note that solutions in $X(S^*)$ may not necessarily respect the students' preference order. To enforce this, it is enough to include the stability constraints from an IP formulation of the many-to-one stable matching problem. Following Baïou and Balinski (2000), we consider

$$c_p x_{sp} + c_p \sum_{q \in P_s : q \rhd_s p} x_{sq} + \sum_{t \in S_p^* : t \blacktriangleright s} x_{tp} \geq c_p \quad \forall (s,p) \in A(S^*). \tag{4}$$

Constraints (4) ensure, for each admissible pair (s, p), that either s is assigned to p or to a more preferred program according to $\rhd_s$, or p is full with higher-ranked students according to $\blacktriangleright$. Since the ranking $\blacktriangleright$ is common to all programs in the current setting, by Proposition 2, there is a unique $x^{\blacktriangleright} \in X(S^*)$ satisfying (4).

Remark. When $c_p = 1$ for all $p \in P$, we recover the classical one-to-one or marriage problem. In that case, by relaxing constraints (3) to nonnegativity only, we obtain a polytope whose vertices are binary vectors (Vate (1989); Rothblum (1992)). However, this is no longer the case for general c_p since the polytope might have fractional vertices (Baïou and Balinski (2000)). Of course, as Rothblum (1992) suggests, we can split each program p into c_p copies with a unique available slot each, thus reducing the many-to-one to the one-to-one setting and recovering integrality of extreme points. We are, however, increasing the number of variables and constraints involved in the formulation, which can become rather expensive to solve if the numbers are large.

We introduce a robust formulation by parameterizing the stability constraints. Define a binary vector $z^{\blacktriangleright}$, representing the exit ranking $\blacktriangleright$, with $z^{\blacktriangleright}_{ts} = 1$ if and only if $t \blacktriangleright s$ for $s, t \in S^*,\ s \neq t$. Then, constraints (4) become

$$c_p x_{sp} + c_p \sum_{q \in P_s : q \rhd_s p} x_{sq} + \sum_{t \in S_p^* :\ t \neq s} z^{\blacktriangleright}_{ts} x_{tp} \geq c_p \quad \forall (s,p) \in A(S^*). \tag{5}$$

We now treat $z^{\blacktriangleright}$ as an adversarial decision vector z rather than a fixed parameter. The feasible set $Z(S^*)$ for exit rankings z corresponds to the set of linear orderings on S^*, which is described by

$$z_{ts} + z_{st} = 1 \quad \forall s, t \in S^* :\ s \neq t \tag{6}$$

$$z_{st} + z_{tu} \leq z_{su} + 1 \quad \forall s, t, u \in S^* :\ s \neq t,\ t \neq u,\ u \neq s \tag{7}$$

$$z_{st} \in \{0, 1\} \quad \forall s, t \in S^* :\ s \neq t. \tag{8}$$

Constraints (6) and (7) enforce antisymmetry and transitivity, respectively, while constraints (8) define the variable domain.

We seek a set $S^* \subseteq S$ that guarantees compatibility across all possible exit rankings $z \in Z(S^*)$. We verify (in)compatibility by solving the optimization problem

$$\begin{aligned} m(S^*) = \min & \sum_{(s,p) \in A(S^*)} x_{sp} \\ \text{s.t.} \quad & c_p x_{sp} + c_p \sum_{q \in P_s : q \rhd_s p} x_{sq} + \sum_{t \in S^*_p :\ t \neq s} z_{ts} x_{tp} \geq c_p \quad \forall (s,p) \in A(S^*) \\ & x \in X(S^*) \\ & z \in Z(S^*). \end{aligned} \tag{9}$$

The set S^* is not robustly compatible if and only if $m(S^*) < |S^*|$, indicating that there exists an exit ranking $\blacktriangleright \in \mathcal{R}(S^*)$, represented by some $z^* \in Z(S^*)$, such that the unique matching $x^* = x^*(z^*) \in X(S^*)$ satisfying (9) leaves at least one student not assigned to a program.

To obtain a linear IP, we linearize the bilinear term $z_{ts} x_{tp}$ in (9) using auxiliary variables w_{spt} along with the constraints

$$w_{spt} \leq x_{tp} \quad \forall (s,p) \in A(S^*),\ t \in S^*_p :\ s \neq t \tag{10}$$

$$w_{spt} \leq z_{ts} \quad \forall (s,p) \in A(S^*),\ t \in S^*_p :\ s \neq t \tag{11}$$

$$w_{spt} \geq x_{tp} + z_{ts} - 1 \quad \forall (s,p) \in A(S^*),\ t \in S^*_p :\ s \neq t \tag{12}$$

$$w_{spt} \in \{0,1\} \quad \forall (s,p) \in A(S^*),\ t \in S^*_p :\ s \neq t, \tag{13}$$

transforming (9) into

$$c_p x_{sp} + c_p \sum_{q \in P_s : q \rhd_s p} x_{sq} + \sum_{t \in S^*_p :\ t \neq s} w_{spt} \geq c_p \quad \forall (s,p) \in A(S^*). \tag{14}$$

Defining $W(S^*)$ as the solution set of (10)–(14), we have that $m(S^*)$ is given by the optimal value of (RobIP) defined by

$$\begin{aligned} \text{(RobIP)} \qquad \min & \sum_{s \in S^*} \sum_{p \in P_s} x_{sp} \\ \text{s.t.} \quad & x \in X(S^*) \\ & z \in Z(S^*) \\ & (x,z,w) \in W(S^*). \end{aligned}$$

We slightly modify the formulation of (RobIP) to improve computational performance. First, note that constraints (13) can be relaxed to nonnegativity only and that constraints (12) can be dropped altogether. Moreover, by writing (7) as

$$z_{st} + z_{tu} + z_{us} \leq 2 \quad \forall s,t,u \in S^* :\ s \neq t,\ t \neq u,\ u \neq s \tag{15}$$

and relaxing (6) to

$$z_{ts} + z_{st} \leq 1 \quad \forall s, t \in S^* : \ s \neq t, \tag{16}$$

we obtain an equivalent formulation in the sense of having the same optimal value $m(S^*)$, but having constraints of packing type which showed better computational performance.

Also, note that by Proposition 6, we have either $m(S^*) = |S^*|$ or $m(S^*) = |S^*| - 1$. Therefore, we include the constraint

$$\sum_{s \in S^*} \sum_{p \in P_s} x_{sp} \geq |S^*| - 1 \tag{17}$$

to improve the lower bound given by the linear relaxation of (RobIP) and to stop solving in case an integer feasible solution with objective value $|S^*| - 1$ is found. We tried other strategies, such as casting the optimization problem as a feasibility decision problem, but ultimately adding constraint (17) reported the best computational performance.

Remark. The linear ordering polytope (Grötschel et al. (1985a)) is an intractable set which is a face of the acyclic subgraph polytope (Grötschel et al. (1985b)), for which facets and cutting planes methods have been studied (Leung and Lee (1994); Grötschel et al. (1984)). Therefore, the sole inclusion of this set in the above formulation is likely to render the problem intractable. This issue is worsened by the fact that to linearize the terms $z_{ts}x_{tp}$, we incur in a cubic number of additional variables and constraints, thus exacerbating the difficulty of solving the problem.

The acceptance/rejection algorithm (Algorithm 1) identifies a robust subset of students from an initial set $S = 1, \ldots, n$, ordered by their entrance ranking. The algorithm starts with an empty set S^* and sequentially considers each applicant s, computing $m(S^* \cup \{s\})$. If adding applicant s maintains robust compatibility, indicated by $m(S^* \cup \{s\}) = |S^*| + 1$, the applicant is accepted into S^*. Otherwise, she is rejected. This sequential process ensures robustness against any possible exit ranking.

Remark. After each iteration where an applicant is admitted, the set S^* increases and so does the overall formulation size and solving time. Since the ground set changes over iterations, the model is solved from scratch without retaining information from previous iterations.

Algorithm 1 Acceptance/Rejection Algorithm for Robust Admission

```
Require: Applicants $S = \{1, \dots, n\}$ ordered by entrance ranking; function $m(\cdot)$ returning
    the number of students matched in the second-stage problem
Ensure: A robustly compatible subset $S^* \subseteq S$
 1: Initialize $S^* \leftarrow \emptyset$
 2: for each applicant $s \in S$ in entrance-ranking order do
 3:     Compute $m(S^* \cup \{s\})$
 4:     if $m(S^* \cup \{s\}) = |S^*| + 1$ then
 5:         $S^* \leftarrow S^* \cup \{s\}$
 6:     else
 7:         Reject applicant $s$
 8:     end if
 9: end for
10: return $S^*$
```

6 Numerical experiments

6.1 Data

We consider data from the 2024, 2025, and 2026 admission cycles of *Bachillerato Inicia*. Table 2 reports the size of each instance in terms of the number of applicants, the number of programs, and the total number of available articulation slots. We have that across the three years, the instances exhibit a clear growth in scale. The number of applicants increases substantially from 146 in 2024 to 248 in 2026, while the number of programs grows more moderately, from 40 to 51. In contrast, the total number of slots remains relatively stable across years, increasing only slightly from 273 to 305. This divergence between applicant growth and relatively flat capacity levels anticipates increased congestion and competition in later cohorts.

Year	Applicants	Programs	Slots
2024	146	40	273
2025	206	50	303
2026	248	51	305

Table 2: Instances size.

Figure 1 displays, for the 2026 instance, the distribution of applicants' declared preferences across programs, disaggregated by first, second, and third choices, together with the available number of slots per program. Programs are sorted in decreasing order of total demand, revealing a highly skewed demand profile: a small subset of programs attracts a disproportionately large number of applicants, while the majority of programs receive relatively low demand. Moreover, the most demanded programs accumulate substantial mass not only from first choices but also from second and third choices, suggesting that these programs play a central role as both top preferences and fallback options. Conversely, programs in the lower-demand tail tend to receive very few first-choice applications and are mostly supported by lower-ranked preferences, if at all. The red line representing slots shows that capacities are relatively homogeneous compared to demand, and do not scale proportionally with popularity. As a result, several highly demanded programs face demand that substantially exceeds their capacity, while many low-demand programs operate with slack capacity. Years 2024 and 2025 share almost identical patterns (skewed demand profile, mismatch between demand concentration, and relatively flat capacities), and we thus omit their figures.

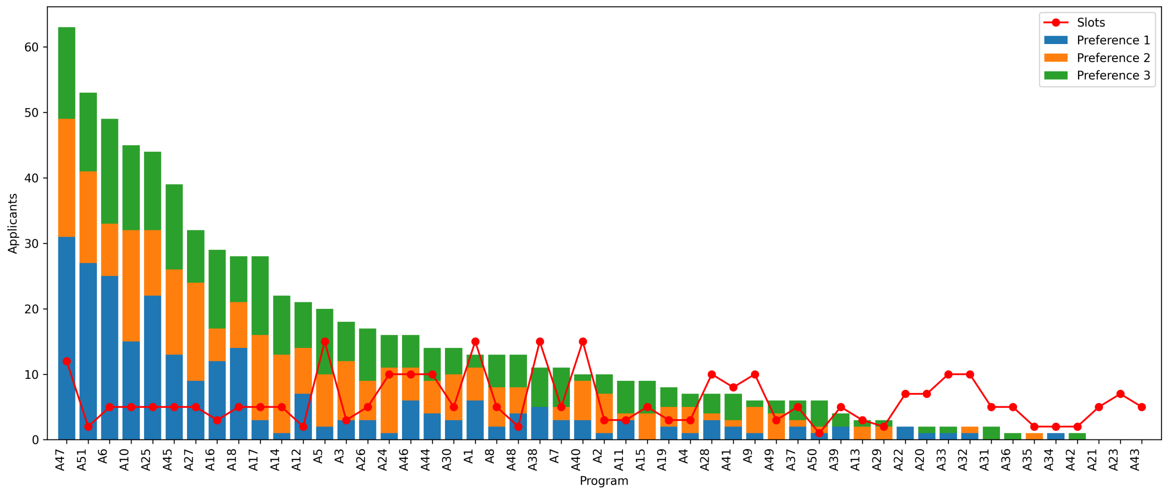

Figure 1: Student preferences and available program slots.

Figure 2 depicts, for the 2026 year, the program co-occurrence graph induced by applicants' preference lists. Nodes represent programs, and an undirected edge connects two programs whenever there exists at least one applicant who listed both programs among their preferences. Moreover, edge width is proportional to the frequency with which the connected programs jointly appear in applicants' lists. We observe that the graph exhibits a highly connected structure. Apart from a small number of isolated nodes corresponding to programs that do not appear in applicants' lists, the graph contains a single giant connected component encompassing almost all programs. In particular, there are no nontrivial connected components that would allow the instance to be decomposed into smaller, independent subproblems. This connectivity reflects substantial overlap in applicants' preference sets, indicating that programs compete for applicants not only directly but also indirectly through chains of shared preferences. As a consequence, allocation decisions at one program can propagate through the entire system, affecting the stability of assignments elsewhere. From a computational perspective, this structural property implies that the overall admission problem cannot be simplified via graph-based decomposition techniques in an obvious way and must be solved as a series of increasingly large instances. As with demand, for 2024 and 2025, the overall topology of the co-occurrence graphs remains stable: they also exhibit dense cores with a few peripheral or isolated programs. We thus omit the corresponding figures.

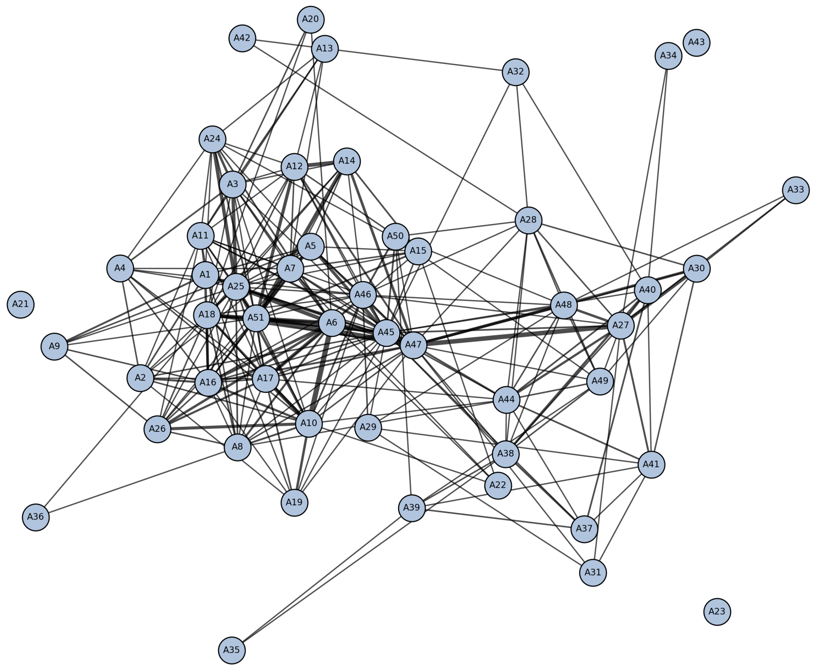

Figure 2: Co-occurrence graph induced by applicants' preference.

6.2 Results

Algorithm 1 was implemented in Python 3.12, using COPT 8.0.3 as IP solver (Ge et al. (2022)). The computational experiments were run under Linux on a desktop computer with an AMD Ryzen 9 9950X processor and 64 GB of RAM. Table 3 summarizes the main outcomes and computational characteristics of the algorithm for each instance.

Year	Applicants	Total accepted	First rejected	90-th accepted	Total time (minutes)
2024	146	97	69	129	6
2025	206	127	47	120	802
2026	248	135	61	115	217

Table 3: Summary of results per year.

The column *Total accepted* reports the final size of the robustly compatible set S^*. Despite the growth in the applicant pool, the number of accepted applicants increases at a slower rate, from 97 in 2024 to 135 in 2026, reflecting tighter compatibility constraints as demand intensifies.

The column *First rejected* indicates the iteration at which the first applicant is rejected. In all instances, this occurs relatively early, well before the end of the process, signaling that incompatibilities arise even when the admitted set is still small. The column 90-th accepted reports the iteration at which the 90th applicant is accepted, providing a proxy for the transition from the initial low-congestion regime to a more saturated phase. Notably, this threshold is reached earlier in the larger instances, indicating that the system enters the high-congestion regime sooner as the market grows.

Finally, *Total time* reports the overall running time of the algorithm in minutes. Computational effort increases dramatically with instance size and congestion. While the 2024 instance is solved in a few minutes, the 2025 and 2026 instances require several orders of magnitude more time. The non-monotonic relationship between instance size and total time further highlights the sensitivity of the algorithm to structural properties of the data, rather than scale alone.

6.3 Detailed results for 2026

We focus our analysis on the 2026 instance since most trends also appear in the 2024 and 2025 instances.

Figure 3 shows the cumulative number of accepted applicants as a function of the iteration. The dashed line represents the identity function, which would correspond to the hypothetical case in which every tested applicant is accepted. We observe that the acceptance curve initially follows the identity line, indicating that during the early iterations every applicant tested can be feasibly admitted. This regime corresponds to a low-congestion phase in which capacity and robustness constraints are largely nonbinding. As iterations progress, the acceptance curve gradually bends away from the identity line, reflecting an increasing frequency of rejections. This slowdown marks the onset of congestion: as the set of accepted applicants grows, additional applicants are more likely to violate robustness conditions, leading to rejections and a sublinear growth of the number of accepted applicants.

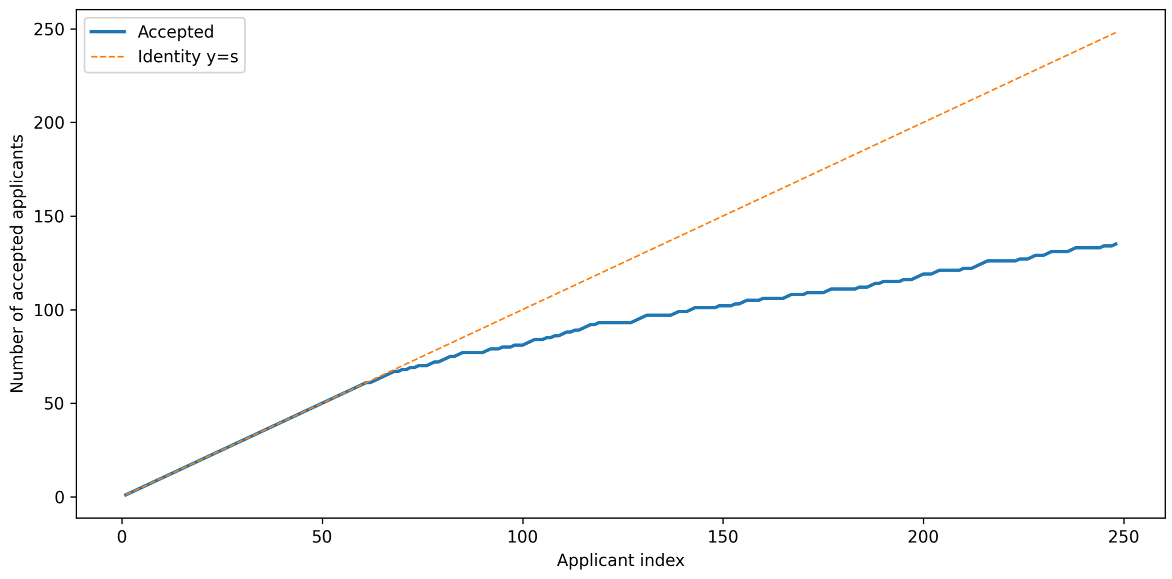

Figure 3: Cumulative number of accepted applicants.

Regarding computational performance, Figure 4 reports the running time (in seconds) of the first 230 iterations, out of 248, of Algorithm 1. Each iteration corresponds to testing the inclusion of a single applicant, and bars are color-coded according to whether the applicant is ultimately accepted (blue) or rejected (orange). Somewhat expected, running times exhibit a clear increasing trend as the algorithm progresses. Early iterations are computationally inexpensive, while later iterations become significantly more costly. This pattern reflects the cumulative nature of the procedure: as more applicants are admitted, the size of (RobIP) grows and verifying that an additional applicant can be included requires solving increasingly complex problems. A salient feature of the results is that accepted applicants systematically require more computational effort than rejected ones, especially in the later stages of the algorithm. Intuitively, acceptance requires certifying that the applicant is compatible with all currently admitted applicants, which involves verifying that all feasible solutions to the IP match all students. In contrast, rejection can often be concluded earlier, once a feasible solution that leaves a student unmatched is identified.

Table 4 reports detailed results for iterations 231-248, corresponding to the final applicants processed. For each applicant, the table shows the acceptance decision, the total running

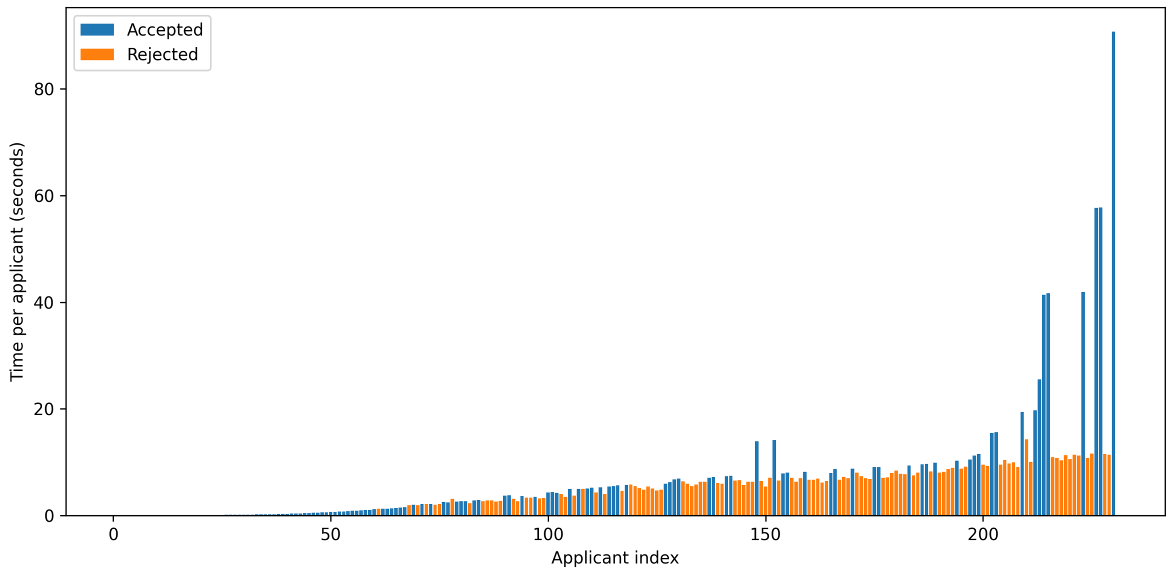

Figure 4: Time spent testing the inclusion of each applicant.

time of the iteration (in seconds), and the number of branch-and-bound nodes explored by the solver. Here, the contrast between accepted and rejected applicants is stark. Rejected applicants are resolved almost immediately, with running times on the order of a few seconds and negligible search effort, often requiring the exploration of only a handful of nodes. In contrast, accepted applicants trigger extremely large search trees and dominate the computational cost of the algorithm, requiring between several hundred and several thousand seconds, and involving the exploration of hundreds of thousands to several million branch-and-bound nodes. These iterations correspond precisely to cases in which the algorithm must certify that all feasible solutions of the underlying IP match all currently admitted applicants, a task that is inherently more demanding than exhibiting a single violating solution.

Finally, Figure 5 compares the optimal value of the LP relaxation of (RobIP) without the lower-bounding constraint (17) with the optimal IP value at each iteration. The blue bars represent the LP optimum, while the orange bars show the remaining integrality gap. We observe that, without additional constraints, the LP relaxation becomes increasingly loose as the algorithm progresses, leading to a growing gap in later iterations. However, by Proposition 6, when the lower bound (17) is enforced, the LP objective differs from the IP optimum by at most one unit at every iteration. In the absence of this lower bound, the IP solver would need to generate additional cutting planes and explore substantially larger branch-and-bound trees to close the gap, significantly increasing computational effort in iterations that lead to rejection.

Applicant	Status	Time (seconds)	Nodes
231	Accepted	738	523196
232	Rejected	12	383
233	Rejected	13	381
234	Rejected	12	191
235	Rejected	13	31
236	Accepted	731	523196
237	Accepted	317	201714
238	Rejected	13	1
239	Rejected	13	1
240	Rejected	13	1
241	Rejected	13	1
242	Rejected	13	57
243	Rejected	13	1
244	Accepted	2861	1559627
245	Rejected	12	1
246	Rejected	14	289
247	Accepted	1630	899569
248	Accepted	5118	2869837

Table 4: Detailed results of iterations 231-248.

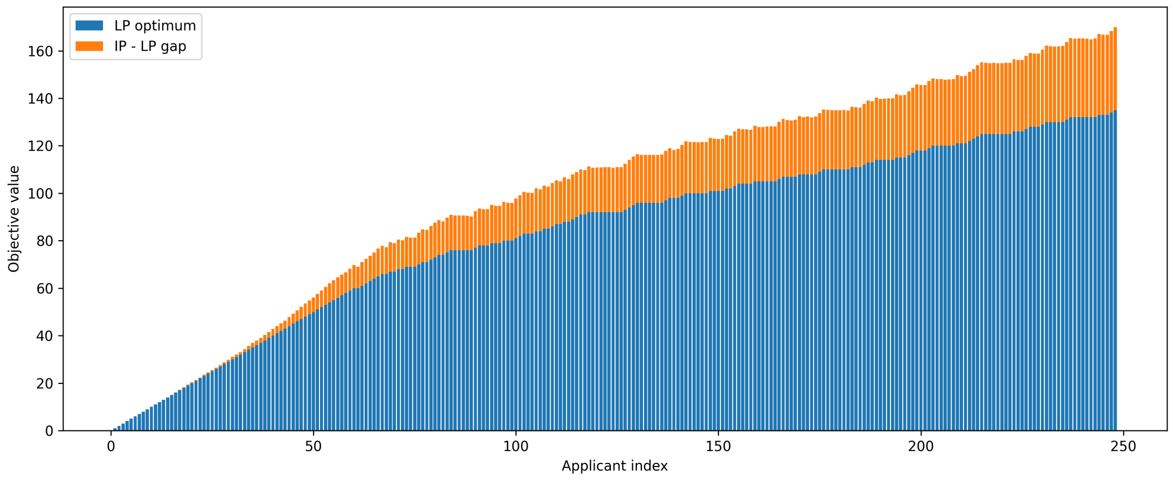

Figure 5: Comparison of LP relaxation and IP optimal value.

6.4 Augmenting preference lists and capacity

We consider two auxiliary procedures used to generate instances that reflect sensible actions that *Bachillerato Inicia* might take to increase the number and input ranking of admitted ap-

Algorithm 2 Correlation-based preference augmentation

```
Require: Preference lists $(P_s)_{s \in S}$, weighted graph $G = (P, E, w)$
Ensure: Augmented preference lists $(\tilde{P}_s)_{s \in S}$
1: for each applicant $s \in S$ do
2:     for each $p \in P \setminus P_s$ do
3:         compute $\text{score}_s(p) = \sum_{q \in P_s : \{p,q\} \in E} w_{pq}$
4:     end for
5:     select $p_4 \in \arg\max_p \text{score}_s(p)$
6:     set $\tilde{P}_s = (P_s, p_4)$
7: end for
```

plicants. The first procedure (Algorithm 2) augments applicants' preference lists by adding a fourth option based on empirical correlation patterns across programs. The second procedure (Algorithm 3) constructs alternative capacity vectors through a saturation-based allocation rule that concentrates additional slots on the most constrained programs.

Correlation-based fourth preference In order to construct augmented preference profiles, we assign a fourth program to each applicant using aggregate information on the similarity between programs. This information is summarized by a weighted, undirected graph on the set of programs, denoted by $G = (P, E, w)$, which is constructed from application data and illustrated in Figure 2. The graph captures empirical patterns of joint demand across programs and is interpreted as a measure of proximity or relatedness between academic options. Using this graph as input, the fourth program assigned to an applicant is chosen so as to be most closely related, in aggregate, to her originally declared choices, while excluding the programs already listed in her preference set.

Saturation-based capacity increments To construct alternative capacity scenarios, we consider a rule that allocates additional slots to programs as a function of their overall demand pressure. For each program $p \in P$, let d_p denote its total demand, defined as the number of applicants who list p anywhere in their preference profile. Let c_p be the original capacity of program p, and let $k \in \mathbb{Z}_+$ denote a fixed budget of additional slots to be distributed. Starting from the original capacity vector, additional slots are assigned sequentially to programs that exhibit the highest level of saturation, measured as a ratio between demand and current capacity. At each step, one unit of capacity is allocated to the program with the largest demand-to-capacity ratio, and the capacity vector is updated accordingly before proceeding

Algorithm 3 Saturation-based capacity expansion

```
Require: Capacities $(c_p)_{p \in P}$, demand $(d_p)_{p \in P}$, budget $k$
Ensure: Capacity increments $(\Delta_p)_{p \in P}$
1: $c^{(0)} \leftarrow c$
2: for $t = 1$ to $k$ do
3:     choose $p^\star \in \arg\max_{p \in P} \frac{d_p}{c_p^{(t-1)}}$
4:     $c_{p^\star}^{(t)} \leftarrow c_{p^\star}^{(t-1)} + 1$
5: end for
6: $\Delta_p \leftarrow c_p^{(k)} - c_p \ \ \forall p \in P$
```

to the next allocation.

We built the additional instances based on the 2026 dataset, restricting executions of Algorithm 1 to the first 230 applicants for computational tractability. Table 5 reports summary statistics analogous to those in Table 3, considering the baseline instance, a variant in which applicants are allowed to declare a fourth program, and scenarios in which articulation capacities are increased by $k = 10, 20, 30$ slots in saturated programs. Figure 6 visualizes the acceptance outcomes across these scenarios. Each row corresponds to one execution, and each applicant is marked as accepted or rejected, allowing a direct comparison of acceptance patterns across instances.

Instance	Applicants	Total accepted	First rejected	90-th accepted	Total time (minutes)
Baseline	230	129	61	115	24
Fourth program	230	129	61	115	1270
$k = 10$	230	128	72	113	22
$k = 20$	230	136	82	103	325
$k = 30$	230	142	86	94	641

Table 5: Summary of results per instance.

The results show that allowing a fourth declared program has no effect on admission outcomes. However, this modification substantially increases total running time, indicating that, under correlated preferences, additional declared options increase computational complexity without improving admission performance. In contrast, increasing articulation capacity has a clear and systematic effect. As k grows, the first rejection is delayed, and the algorithm reaches the 90th accepted applicant earlier, reflecting reduced congestion in the articulation stage. At the same time, the total number of accepted applicants increases from 129 in the

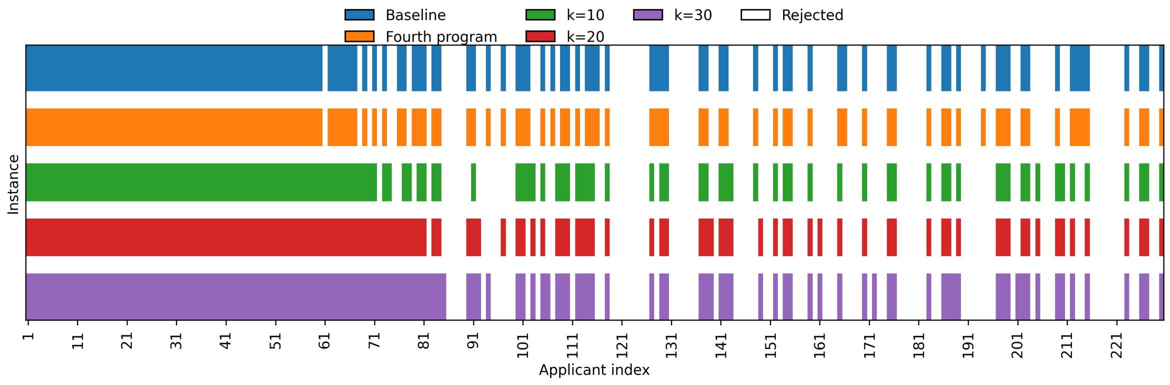

Figure 6: Accepted applicants per instance.

baseline to 142 for $k = 30$. This improvement comes at a significant computational cost, as total running time grows rapidly with k, exceeding several hundred minutes for larger capacity increases.

A closer inspection of Figure 6 also reveals that acceptance sets are not nested across scenarios. In particular, focusing on applicants in the range 90-100, we observe that the sets of accepted applicants differ in a non-monotone way: some applicants accepted in one execution may be rejected in another, and vice versa. This behavior reflects the path-dependent nature of Algorithm 1. Once two executions diverge at a given applicant, subsequent compatibility tests are performed against different admitted sets, which may trigger further divergences downstream. Importantly, this observation is not in conflict with Propositions 4 and 5. Both results are comparative statics statements that apply to a fixed robustly compatible set S^*: extending preference lists or increasing capacity cannot invalidate robustness of an already robust set. In contrast, the algorithm constructs S^* sequentially, and different parameter choices may lead the algorithm to select different robustly compatible sets, none of which need to be nested. Thus, non-containment across executions is an algorithmic phenomenon rather than a violation of robustness monotonicity.

Taken together, Table 5 and Figure 6 suggest that while expanding preference lists offers no practical benefit, moderate increases in articulation capacity at saturated programs can meaningfully improve admission outcomes. From a practical perspective, these results indicate that targeted capacity expansions may allow *Bachillerato Inicia* to admit more and higher-ranked students, albeit at the expense of increased computational effort.

7 Implementation experience

The enrollment process for inclusion and special access programs at Chilean universities is conducted within a highly constrained time window, typically lasting between two and three days after the official publication of the data reflecting students' performance in secondary education and national examinations. *Bachillerato Inicia* extends enrollment offers concurrently with other inclusion programs at PUC and with academic programs at other universities. Consequently, students may be preselected and receive enrollment offers from multiple programs. Most academic programs in Chile do not employ algorithmic optimization in the final stage of selection. Instead, enrollment offers are issued strictly in accordance with the input ranking.

A nontrivial fraction of students accepted by the algorithm into *Bachillerato Inicia* decline the enrollment offer. These rejections create vacancies, allowing admission of students who were not selected in the initial round. Prior to extending offers to students lower in the input ranking, it is preferable to update the candidate list by removing students who have declined enrollment. This adjustment allows higher-ranked students to be reconsidered and potentially selected in subsequent rounds. In practice, the algorithm is therefore executed multiple times. The short enrollment window, combined with the need for repeated re-optimization, imposes significant constraints on the computational performance of the algorithm and emphasizes minimizing execution time.

In the 2026 admission cycle, the algorithm was executed in three rounds. In the first round, the algorithm considered the top 150 students in the input ranking, of whom 101 were selected. In the second round, students who declined enrollment in the first round were removed, and candidates ranked 151-248 were included. In this round, the algorithm accepted 37 additional students, including two students who had not been selected in the first round. In the third round, after additional removals due to enrollment rejections, the algorithm selected nine students who had not been previously selected. The enrollment rejection rate remained approximately constant across rounds, at approximately 23%. The final enrolled cohort for the program in the 2026 admission consisted of 113 students. Note that by Proposition 3, admitted students who accept the offer remain robustly compatible, and they provide the initialization of S^* in Algorithm 1 for subsequent rounds.

The model developed in this study can be implemented in two different ways. In the first case, a strict implementation guarantees that all selected students are assigned to one of the three preferences declared at the time of application, based on their performance ranking at the end of the first year. Under this strict implementation, students who wish to access a program different from their initially declared preferences lose priority in the ranking and may only access their preferred program if places remain available after all other assignments have been completed. Again, by Proposition 3, the students who do not change their lists remain robustly compatible.

A second implementation of the model allows students to engage in vocational exploration during the first year and to change their program preferences without losing their position in the exit ranking. In this case, the role of the model is to guarantee an initial balance between preferences and available places. If preference changes are relatively limited and there are unused places, the number of students who fail to access their preferred programs at exit may be relatively small.

In *Bachillerato Inicia*, the model was applied in an almost strict manner for the 2024 cohort and with preference changes allowed for the 2025 cohort. In the 2024 cohort, 3% of students changed their first preference, and 92% were assigned to their first preference at the end of the first year. In the 2025 cohort, 19% of students changed their first preference, and 95% were assigned to their first preference at the end of the first year. For the *Bachillerato Inicia* program, the implementation of the model that allows changes in preferences introduces greater flexibility in students' options, without a significant loss in first-preference assignment outcomes.

We also highlight that the first-year retention rate of *Bachillerato Inicia* was 93% in years 2024 and 2025. This sharply contrasts with the situation of similar programs in Chile as outlined in the Introduction. For the 2026 cohort, if the first 125 applicants were admitted, by using (RobIP), we find that up to 28 students could be left unmatched in the worst case. By avoiding such situations, *Bachillerato Inicia* is likely to keep high retention rates.

8 Conclusions

This paper studies a two-stage admission and assignment problem under uncertainty, where admission decisions must remain feasible for all possible exit orders of admitted applicants. We formalize this requirement through a notion of robust compatibility and derive an adversarial formulation, based on an integer programming model of stable matching, to test robustness under worst-case exit rankings. This formulation leads to a sequential admission algorithm that incrementally constructs a robustly compatible set of applicants. Computational results on real admission data show that robustness could come at a significant computational cost, with total running time driven by a small number of late acceptance decisions in congested regimes. The implementation of the mechanism has led to high first-year retention in *Bachillerato Inicia*.

There are several avenues for future research. From the stable matching theory perspective, it would be interesting to extend the problem to the general many-to-one case by allowing programs to have different preference profiles over students, or to specialize the results to the classical stable marriage problem. In either case, properties of robustly compatible sets should be revisited. From the algorithmic point of view, the large computing times exhibited by some iterations call for improving the IP formulation, for instance, by carrying out a polyhedral study to obtain facets of the convex hull of feasible solutions of (RobIP). It would also be interesting to allow some degree of risk and limit the set of exit rankings against which the model protects, similar to d-robustness of Chen et al. (2021).

References

Abdulkadiroğlu, A. and Sönmez, T. (2003). School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747.

Afacan, M. O. (2012). Group robust stability in matching markets. *Games and Economic Behavior*, 74(1):394–398.

Aziz, H., Biró, P., Gaspers, S., de Haan, R., Mattei, N., and Rastegari, B. (2020). Stable matching with uncertain linear preferences. *Algorithmica*, 82:1410–1433.

Baïou, M. and Balinski, M. (2000). The stable admissions polytope. *Mathematical Programming*, 87:427–439.

Bikhchandani, S. (2017). Stability with one-sided incomplete information. *Journal of Economic Theory*, 168:372–399.

Bordon, P. and Fu, C. (2015). College-major choice to college-then-major choice. *The Review of Economic Studies*, 82(4):1247–1288.

Bredereck, R., Chen, J., Knop, D., Luo, J., and Niedermeier, R. (2020). Adapting stable matchings to evolving preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1830–1837.

Chakraborty, A., Citanna, A., and Ostrovsky, M. (2010). Two-sided matching with interdependent values. *Journal of Economic Theory*, 145(1):85–105.

Chen, J., Skowron, P., and Sorge, M. (2021). Matchings under preferences: Strength of stability and tradeoffs. *ACM Transactions on Economics and Computation*, 9(4):20:1–20:55.

DEMRE (2026). Departamento de Evaluación, Medición y Registro Educacional. `https://www.demre.cl`. Accessed: January 2026.

Ehlers, L. and Massó, J. (2015). Matching markets under (in) complete information. *Journal of Economic Theory*, 157:295–314.

Faenza, Y., Foussoul, A., and He, C. (2024). Two-stage stochastic stable matching. In *Integer Programming and Combinatorial Optimization (IPCO 2024)*, pages 154–167. Springer.

Faenza, Y., Foussoul, A., and He, C. (2025). Minimum cut representability of stable matching problems. *arXiv preprint arXiv:2504.04577*.

Gajulapalli, K., Liu, J. A., Mai, T., and Vazirani, V. V. (2020). Stability-preserving, time-efficient mechanisms for school choice in two rounds. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

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

Gangam, R., Mai, T., Raju, N., and Vazirani, V. V. (2026). Robust stable matchings: Dealing with changes in preferences. *arXiv preprint arXiv:2601.07959*.

Ge, D., Huangfu, Q., Wang, Z., Wu, J., and Ye, Y. (2022). Cardinal Optimizer (COPT) user guide. https://guide.coap.online/copt/en-doc.

Genc, B., Siala, M., O'Sullivan, B., and Simonin, G. (2017). Robust stable marriage. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Grötschel, M., Jünger, M., and Reinelt, G. (1984). A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220.

Grötschel, M., Jünger, M., and Reinelt, G. (1985a). Facets of the linear ordering polytope. *Mathematical Programming*, 33:43–60.

Grötschel, M., Jünger, M., and Reinelt, G. (1985b). On the acyclic subgraph polytope. *Mathematical Programming*, 33:28–42.

Irving, R. W., Manlove, D. F., and Scott, S. (2008). The stable marriage problem with master preference lists. *Discrete Applied Mathematics*, 156(15):2959–2977.

Kojima, F. (2011). Robust stability in matching markets. *Theoretical Economics*, 6(2):257–267.

Leung, J. and Lee, J. (1994). More facets from fences for linear ordering and acyclic subgraph polytopes. *Discrete Applied Mathematics*, 50(2):185–200.

Liu, Q., Mailath, G. J., Postlewaite, A., and Samuelson, L. (2014). Stable matching with incomplete information. *Econometrica*, 82(2):541–587.

Mai, T. and Vazirani, V. V. (2018). Finding stable matchings that are robust to errors in the input. In *26th Annual European Symposium on Algorithms (ESA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

MINEDUC (2026a). Acceso Educación Superior. `https://www.mifuturo.cl/`. Accessed: January 2026.

MINEDUC (2026b). Datos Abiertos. `https://http://www.datosabiertos.mineduc.cl/`. Accessed: January 2026.

MINEDUC (2026c). Mi Futuro. `https://www.mifuturo.cl/`. Accessed: January 2026.

Ríos, I., Larroucau, T., Parra, G., and Cominetti, R. (2021). Improving the Chilean college admissions system. *Operations Research*, 69(4):1186–1205.

Roth, A. E. (1989). Two-sided matching with incomplete information about others' preferences. *Games and Economic Behavior*, 1(2):191–209.

Roth, A. E. and Sotomayor, M. A. O. (1990). *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monographs. Cambridge University Press, Cambridge.

Rothblum, U. G. (1992). Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, 54:57–67.

Salas-Velasco, M. (2024). Vocational education and training systems in europe: A cluster analysis. *European Educational Research Journal*, 23(3):434–449.

Vate, J. H. V. (1989). Linear programming brings marital bliss. *Operations Research Letters*, 8(3):147–153.