

Capacity Planning in Stable Matching: An Application to School Choice*

Federico Bobbio [†] Margarida Carvalho [‡] Andrea Lodi [§] Ignacio Rios [¶]
Alfredo Torrico ^{||}

Abstract

In this work, we introduce the problem of jointly allocating a school capacity expansion (given a fixed budget) and finding the best allocation for the students in the expanded market. Given the computational complexity of the problem, we provide an integer quadratically-constrained programming formulation and study its linear reformulations. We also propose two heuristics: A greedy algorithm and an LP-based method. We empirically evaluate the performance of our approaches in a detailed computational study. We observe the practical superiority of the linearized model in comparison with its quadratic counterpart and we outline their computational limits. Finally, we use the Chilean school choice system data to empirically demonstrate the impact of capacity planning under stability conditions. Our results show that each additional school seat can benefit multiple students. In addition, depending on the decision-maker, our methodology can prioritize the assignment of previously unassigned students or improve the assignment of several students through improvement chains.

1 Introduction

Centralized mechanisms are becoming the standard approach to solve several assignment problems. Examples include the allocation of students to schools, high-school graduates to colleges, residents to hospitals and refugees to cities. In most of these markets, a desirable property of the assignment is *stability*, which guarantees that no pair of agents has incentive to circumvent the match. As discussed in Roth [2002], finding a stable matching is crucial for the clearinghouse’s success, long-term sustainability and also ensures some notion of fairness as it eliminates the so-called *justified-envy*.

A common assumption in these markets is that capacities are fixed and known in advance. However, capacities are only a proxy of how many agents can be accommodated, and there is some flexibility to modify them in multiple settings. For instance, in refugee resettlement, local authorities define how many refugees they are willing to receive, but they could increase their capacity given proper incentives. Similarly, in school choice, administrators report how many open seats they have in each grade, based on their current enrollment and the size of their classrooms. However, they could switch classrooms of different sizes to modify the seats offered on each level. Finally, in college

*This work was funded by FRQ-IVADO Research Chair in Data Science for Combinatorial Game Theory, and NSERC grant 2019-04557.

[†]CIRRELT and DIRO, Université de Montréal. CERC, Polytechnique Montréal. federico.bobbio@umontreal.ca

[‡]CIRRELT and DIRO, Université de Montréal. carvalho@iro.umontreal.ca

[§]Jacobs Technion-Cornell Institute, Cornell Tech. andrea.lodi@cornell.edu

[¶]School of Management, University of Texas at Dallas. ignacio.riosuribe@utdallas.edu

^{||}CERC Data Science, Polytechnique Montréal. alfredo.torrico-palacios@polymtl.ca

admissions, colleges may increase their capacities to admit all tied students competing for the last seat [Rios et al., 2021].

As the previous discussion illustrates, capacities may be flexible, and it may be natural to incorporate them as a decision to further improve the assignment process. Indeed, by jointly deciding capacities and the allocation, the clearinghouse can leverage its knowledge about agents’ preferences to achieve different goals. On the one hand, one possible goal is to maximize *access*, i.e., to choose the capacity that maximizes the total number of agents being assigned. This objective is especially relevant in some settings, such as school choice, where the clearinghouse wants to ensure that it gives every student a seat in some school. On the other hand, the clearinghouse may wish to prioritize *fairness*, i.e., improving the assignment of high-priority students. This objective is common in merit-based settings such as college admissions and the hospital-resident problem. Therefore, in the setting with capacity expansion, there might be multiple *optimal assignments* due to the trade-off between *fairness* and *access*. This is particularly relevant, because in the vanilla version of the problem without capacity expansion, there is a unique optimal assignment and the aforementioned trade-off does not occur.

In this paper, we study how to jointly find a stable assignment and make capacity decisions. To accomplish this, we introduce a capacity planning model in the context of stable matching. We show that we can formulate the problem as a quadratically-constrained integer program, and we provide two linearizations to facilitate its solution. Given the complexity of the problem [Bobbio et al., 2022], we propose two natural heuristics: (1) a greedy approach, which increases the capacity of the school leading to the highest improvement; and (2) a linear programming approach, which relaxes the stability constraints to make capacity decisions, and later considers the updated capacities to find a stable assignment using the Deferred Acceptance (DA) algorithm—the classic method for stable assignments. Finally, we use the Chilean school choice system data to show the potential benefits of embedding capacity decisions. Our results show that each additional seat can benefit multiple students. In addition, depending on how we define the objective, our methodology can prioritize the assignment of previously unassigned students or improve the assignment of several students through improvement chains. Our results show that the proposed heuristics are computationally efficient and achieve close-to-optimal solutions.

1.1 Contributions and Societal Impact

Our work contributes to the literature in many dimensions.

Problem formulation. To capture the problem described above, we introduce a stylized model of a many-to-one matching market, in which the clearinghouse can make capacity planning decisions while simultaneously finding a student-optimal stable matching. We show theoretically and empirically that the clearinghouse can prioritize different goals when making capacity decisions. On one end, the clearinghouse may prioritize *access*, i.e., maximize the number of students assigned. On the other, the clearinghouse may prioritize *fairness*, i.e., maximize the preference of assignment of highly ranked students. Our model is flexible enough to accommodate any of these goals and, more importantly, to understand the trade-offs between them. Finally, we provide some results about the strategy-proofness of the mechanism, and we show that we can extend our model in several interesting directions.

Mathematical programming approach. In the standard case with no capacity decisions, we first show that we can find the student-optimal stable matching by minimizing the sum of the students’ assignment preferences over the set of feasible stable matchings. In addition, when we

include capacity decisions, we show that we can formulate the problem as an integer quadratically-constrained program (IQCP). We provide two different linearizations to improve the computational efficiency and we show that one outperforms the other.

Heuristics. As shown in [Bobbio et al., 2022], our problem is NP-hard, motivating the design of heuristics that would provide good solutions in a reasonable computational time. In particular, we propose two new heuristics to solve the problem. Our first heuristic, called *Greedy*, assigns extra seats to the schools leading to the most significant change in the objective function. Our second heuristic, called *LPH*, proceeds in two steps by first solving the problem without stability constraints to find the allocation of extra seats and then finding the student-optimal stable matching conditional on the capacities defined in the first step. Our computational experiments show that both heuristics significantly reduce the time to find a close-to-optimal solution. Moreover, we find that LPH outperforms Greedy when the budget of extra seats increases.

Empirical application. To illustrate the benefits of embedding capacity decisions, we use data from the Chilean school choice system and we adapt our framework to solve the problem including all the specific features described in [Correa et al., 2022]. First, we show that each additional seat can benefit multiple students. Second, in line with our theoretical results, we find that *access* and *fairness* can be prioritized depending on how unassigned students are penalized in the objective. Finally, our results show that the students’ allocation improve even if we upper bound the total number of additional seats per school. Given these positive results, we are currently collaborating with the institution in charge of implementing the Chilean school choice system to test our framework on the field.

1.2 Related Work

Gale and Shapley [1962] introduce the well-known Deferred Acceptance algorithm, which finds a stable matching in polynomial time for any instance of the problem. Since then, the literature on stable matchings has extensively grown and has focused on multiple variants of the problem. For this reason, we focus on the most closely related literature, and we refer the interested reader to [Manlove, 2013] for a broader literature review.

Mathematical programming formulations. The first mathematical programming formulations of the stable matching problem were studied in [Gusfield and Irving, 1989, Vate, 1989, Rothblum, 1992, Roth et al., 1993]. Thereafter, Baïou and Balinski [2000] provide an exponential size linear programming formulation and prove that it coincides with the convex hull of the set of feasible stable matchings. Moreover, they give a polynomial-time separation algorithm. Kwanashie and Manlove [2014] give an integer formulation of the problem when there are ties in the preference lists. Kojima et al. [2018] provide a way to represent preferences and constraints to guarantee strategy-proofness. Ágoston et al. [2016] introduce an integer model that incorporates upper and lower quotas. Delorme et al. [2019] devise new mixed-integer programming formulations and pre-processing procedures. More recently, Ágoston et al. [2021] propose similar mathematical programs and use them to compare different policies to deal with ties. We contribute to this literature by adding capacity decisions in finding a stable matching.

Capacity expansion. To the best of our knowledge, our paper is the first to introduce the problem of capacity planning in the context of stable matching. Closely related to our paper is [Bobbio et al.,

2022], who studies the complexity of our problem and of other extensions. Specifically, the authors show that the decision version of the problem is NP-complete and inapproximable within a $O(\sqrt{n})$ factor, where n is the number of students. Abe et al. [2022] study this problem and propose an alternative method to solve the capacity expansion problem. Specifically, their approach relies on an anytime method where the upper confidence tree (UCT) searches the space of capacity expansions. Finally, Nguyen and Vohra [2018] study the impact of capacity expansion when there are couples in the market. In particular, they show that the existence of a stable matching is guaranteed if the capacity of the market is expanded by at most 9 spots.

School choice. Starting with [Abdulkadiroğlu and Sönmez, 2003], a large body of literature has studied different elements of the school choice problem, including the use of different mechanisms such as DA, Boston, and TTC [Abdulkadiroğlu et al., 2005, Pathak and Sönmez, 2008, Abdulkadiroğlu et al., 2011]; the use of different tie-breaking rules [Abdulkadiroğlu et al., 2009, Arnosti, 2015, Ashlagi et al., 2019]; the handling of multiple and potentially overlapping quotas [Kurata et al., 2017, Sonmez and Yenmez, 2019]; the addition of affirmative action policies [Ehlers, 2010, Hafalir et al., 2013]; and the implementation in many school districts and countries [Abdulkadiroğlu et al., 2005, Calsamiglia and Güell, 2018, Correa et al., 2022]. Within this literature, the closest papers to ours are those that combine the optimization of different objectives with finding a stable assignment. Caro et al. [2004] introduce an integer programming model to make school redistricting decisions. Shi [2016] proposes a convex optimization model to decide the assortment of schools to offer to each student to maximize the sum of utilities. Ashlagi and Shi [2016] present an optimization framework that allows them to find an assignment pursuing (the combination of) different objectives, such as average and min-max welfare. Finally, Bodoh-Creed [2020] presents an optimization model to find the best stable and incentive-compatible match that maximizes any combination of welfare, diversity, and prioritizing the allocation of students to their neighborhood school. Our paper contributes to this literature by introducing capacity planning in the problem of finding a stable matching, and showing its potential impact in the Chilean school choice system.

Entry comparative static and strategy-proofness. The design of a stable matching mechanism when the number of participants of one side is increased has already been investigated through the lens of game theory. In particular, for the stable marriage problem, this is known as the *entry comparative static*. In [Kelso Jr and Crawford, 1982, Gale and Sotomayor, 1985, Roth and Sotomayor, 1990], the authors prove, for example, that when a new woman is added to the instance, all men are matched to a *weakly better* partner. Recently, Kominers [2019] extended this result to the school admission problem. In [Balinski and Sönmez, 1999], the authors show that the DA algorithm is invariant with respect to improvements of the students’ position in the preference lists of the schools. The literature has also studied matching mechanisms that incentivize participants to reveal their true preferences; this is known as strategy-proofness. For instance, Sönmez [1997] proves that schools can manipulate the stable matching in their favour by falsely reporting a reduced capacity. Romm [2014] proves that manipulation is still possible even if the reported capacities are enforced during the admission process.

2 Model

We formalize the stable matching problem using school choice as an illustrating example.¹ Let $\mathcal{S} = \{i_1, \dots, i_{|\mathcal{S}|}\}$ be the set of students, and let $\mathcal{C} = \{j_1, \dots, j_{|\mathcal{C}|}\}$ be the set of schools.²

Each student $s \in \mathcal{S}$ reports a strict preference order $\succ_s$ over the elements in $\mathcal{C} \cup \{\emptyset\}$. Note that we allow for $\emptyset \succ_s c$ for some $c \in \mathcal{C}$, so students may not include all schools in their preference list. In a slight abuse of notation, we use $|\succ_s|$ to represent the number of schools to which student s applies and prefers compared to being unassigned, and we use $c' \succeq_s c$ to represent that either $c' \succ c$ or that $c' = c$. On the other side of the market, each school ranks each student according to a strict order $\succ_c$. Moreover, we assume that each school $c \in \mathcal{C}$ offers a number of seats $q_c \in \mathbb{Z}_+$ and we assume that $\emptyset$ has infinite capacity.

Let $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{C} \cup \{\emptyset\}$ be the set of feasible pairs, with $(s, c) \in \mathcal{E}$ meaning that s applied and satisfies all requirements to be admissible at school c . We assume that $(s, \emptyset) \in \mathcal{E}$ for all $s \in \mathcal{S}$. Then, a *matching* is an assignment $\mu \subseteq \mathcal{E}$ such that each student is assigned to at most one school, and each school receives at most q_c students. In a slight abuse of notation, we use $\mu(s) \in \mathcal{C} \cup \{\emptyset\}$ to represent the school of student s in the assignment μ , with $\mu(s) = \emptyset$ representing that s is unassigned in μ . Similarly, we use $\mu(c) \subseteq \mathcal{S}$ to represent the set of students assigned to c in μ . A matching μ is stable if it has no *blocking pairs*, i.e., if it does not exist a pair $(s, c) \in \mathcal{E}$ that would prefer to be assigned to each other compared to their current assignment in μ . Formally, we say that (s, c) is a *blocking pair* if the following two conditions are satisfied: (1) student s prefers school c over $\mu(s) \in \mathcal{C} \cup \{\emptyset\}$, and (2) $|\mu(c)| < q_c$ or there exists $s' \in \mu(c)$ such that $s \succ_c s'$, i.e., c prefers s over s' .

Gale and Shapley [1962] show that, for any instance $\Gamma = \langle \mathcal{S}, \mathcal{C}, \succ, \mathbf{q} \rangle$ of the problem, a stable matching can be found in polynomial time using the Deferred Acceptance algorithm. Moreover, by changing the proposing side, DA can be adapted to find (i) the *student-optimal* stable matching, i.e., the unique stable matching that is weakly preferred by all students, and (ii) the *school-optimal* stable matching, i.e., the unique stable matching that all schools weakly prefer. In Appendix 9, we formally describe the student-optimal DA algorithm.

Let $r_{s,c}$ be the position of school $c \in \mathcal{C}$ in s 's preference order, and let $r_{s,\emptyset}$ be a parameter that represents the penalty to have student s unassigned. In Lemma 2.1 we show that, for any instance of the problem, we can find the student-optimal stable assignment by solving an integer linear program whose objective is to minimize the sum of students' preference of assignment. The proof can be found in Appendix 6.

Lemma 2.1. *Given an instance $\Gamma = \langle \mathcal{S}, \mathcal{C}, \succ, \mathbf{q} \rangle$, finding the student-optimal stable matching is*

¹Notice that once we break ties, students are strictly ordered in each school (given their priority, random tie-breaker, and possibly the quotas the student is eligible). Hence, the school choice setting becomes similar to other two-sided stable matching problems such as college admissions, the hospital-resident problem, among others.

²To facilitate the exposition, we assume that all students belong to the same grade, e.g., pre-kindergarten.

equivalent to solving the following integer program:

$$\begin{aligned}
\min_x \quad & \sum_{(s,c) \in \mathcal{E}} r_{s,c} \cdot x_{s,c} \\
\text{st.} \quad & \sum_{c \in \mathcal{C} \cup \{\emptyset\}} x_{s,c} = 1, & \text{for all } s \in \mathcal{S}, \\
& \sum_{s \in \mathcal{S}} x_{s,c} \leq q_c, & \text{for all } c \in \mathcal{C}, \\
& q_c x_{s,c} + q_c \cdot \sum_{c' \succ_s c} x_{s,c'} + \sum_{s' \succ_c s} x_{s',c} \geq q_c, & \text{for all } (s,c) \in \mathcal{E}, \\
& x_{s,c} \in \{0, 1\}, & \text{for all } (s,c) \in \mathcal{E}.
\end{aligned} \tag{1}$$

Note that the Integer Program (1) can be solved in polynomial time using the reformulation and the separation algorithm proposed in Baïou and Balinski [2000].

2.1 Capacity Expansion

In this section, we adapt Formulation (1) to incorporate capacity expansion decisions. Let $\mathbf{t} = \{t_c\}_{c \in \mathcal{C}} \in \mathbb{Z}_+^{|\mathcal{C}|}$ be the vector of additional seats allocated to each school $c \in \mathcal{C}$, and let $\Gamma_{\mathbf{t}} = \langle \mathcal{S}, \mathcal{C}, \succ, \mathbf{q} + \mathbf{t} \rangle$ the instance of the problem in which the capacity of each school c is $q_c + t_c$.³ The *capacity expansion problem* consist in finding how to allocate a budget B , and the corresponding student-optimal stable matching, leading to the minimum sum of preferences of assignment. This can be formalized as follows:

Problem 1. *Given a budget $B \in \mathbb{Z}_+$ and an instance Γ , the capacity expansion problem, is the following:*

$$\min_{\mathbf{t}, \mu} \left\{ \sum_{(s,c) \in \mu} r_{s,c} : \mathbf{t} \in \mathbb{Z}_+^{\mathcal{C}}, \sum_{c \in \mathcal{C}} t_c \leq B, \mu \text{ is a stable matching in instance } \Gamma_{\mathbf{t}} \right\}.$$

In other words, an optimal allocation of extra seats in Problem 1 leads to a student-optimal stable matching whose objective value is the best among all feasible capacity expansions. Note that Problem 1 may have multiple optimal allocations that lead to different student-optimal stable matchings with the same objective value (see Example 7.1 in Appendix 7). Also, observe that an optimal allocation may not necessarily use the entire budget since the objective value may no longer improve (e.g., if we assign every student to their top preference). However, we can assume that the entire budget was used by arbitrarily assigning the remaining seats without affecting the objective value. Finally, in Appendix 14 we show that the mechanism may not be strategy-proof for students if they know that capacities may be expanded but do not know the exact allocation of the extra seats.

It is also important to notice that the solution to Problem 1 largely depends on the value of $r_{s,\emptyset}$, i.e., the *penalty* for having unassigned students. One could expect that for larger penalty values, the optimal solution will prioritize *access* by matching initially unassigned students. In contrast, for smaller penalty values, the focus will be on *fairness* by prioritizing *chains of improvement* that enable multiple students to improve their initial assignment.

In the following results, we formalize the trade-off between access and fairness. Given a budget $B > 0$ and penalty values $\{r_{s,\emptyset}\}_{s \in \mathcal{S}}$, let $\mathcal{S}_E(B) \subseteq \mathcal{S}$ be the set of students who *enter* the system,

³Notice that $\Gamma_{\mathbf{0}}$ corresponds to the original instance Γ with no capacity expansion.

i.e., who are not initially assigned (with $B = 0$), but are assigned to one of their preferences when capacities are expanded. Similarly, let $\mathcal{S}_I(B) \subseteq \mathcal{S}$ be the set of students who *improve*, i.e., students who are initially assigned to some preference but improve their preference of assignment when capacities are expanded.⁴

In Theorem 2.2 we show that, when the penalty is small enough, the optimal solution will prioritize fairness by improving the assignment of previously assigned students.

Theorem 2.2. *Suppose that $r_{s,\emptyset} = 0$ for all $s \in \mathcal{S}$. Then, for any integer budget $B > 0$, $|\mathcal{S}_I(B)| \geq |\mathcal{S}_E(B)|$.*

On the other hand, if the penalty is sufficiently large, we show in Theorem 2.3 that the optimal solution will prioritize access by obtaining a stable matching with maximum cardinality.

Theorem 2.3. *Suppose that $r_{s,\emptyset}$ is sufficiently large for all $s \in \mathcal{S}$. Then, an optimal solution $(\mathbf{x}^*, \mathbf{t}^*)$ of Problem 1 returns a maximum cardinality stable-matching.*

Note that Theorem 2.3 guarantees that the resulting stable matching will assign as many students as possible. However, it does not specify which students are prioritize. As we show in Proposition 2.4, we cannot guarantee that the number of students who enter will exceed those who improve.

Proposition 2.4. *There exists an instance Γ of Problem 1 with $B = 1$ for which there is no penalty $r_{s,\emptyset}$ that guarantees $|\mathcal{S}_E(B)| > |\mathcal{S}_I(B)|$.*

All proofs can be found in Appendix 7. As these results illustrate, the penalty values directly impact the outcome of Problem 1. In Section 4, we empirically analyze the trade-off between access and fairness using real data from the Chilean school choice system. For instance, we observe that if $r_{s,\emptyset}$ is large, the optimal solution prioritizes initially unassigned students. On the other hand, if we set $r_{s,\emptyset} = |\succ_s| + 1$, we observe that the optimal solution considerably prioritizes the improvement of initially assigned students. Hence, which value of $r_{s,\emptyset}$ to use is a policy-relevant question that impacts the trade-off between access and fairness.

Our goal in this work is also to study different methodologies to solve Problem 1. Bobbio et al. [2022] prove that when the preference lists of the students are complete, i.e., they rank all the schools, Problem 1 is NP-hard. In this context, a possible approach is to design an approximation algorithm for this problem. Note that for a fixed $\mathbf{t} \in \mathbb{Z}_+^{\mathcal{C}}$, the value

$$f(\mathbf{t}) := \min_{\mu} \left\{ \sum_{(s,c) \in \mu} r_{s,c} : \mu \text{ is a stable matching in instance } \Gamma_{\mathbf{t}} \right\} \quad (2)$$

can be computed in polynomial time by using the DA algorithm on instance $\Gamma_{\mathbf{t}}$. Therefore, one might be tempted to show that f is a lattice submodular function [Soma and Yoshida, 2015].⁵ However, this is not the case as we show in the following result.

Proposition 2.5. *The function $f(\mathbf{t})$ defined in Expression (2) is neither lattice submodular nor supermodular.*

⁴Note that the sets of students that *enter* and *improve* are match-specific. However, our results apply to any stable matching resulting from solving our problem.

⁵A function $f : \mathbb{Z}_+^{\mathcal{C}} \rightarrow \mathbb{R}_+$ is said to be lattice submodular if $f(\mathbf{t} \vee \mathbf{t}') + f(\mathbf{t} \wedge \mathbf{t}') \leq f(\mathbf{t}) + f(\mathbf{t}')$ for any $\mathbf{t}, \mathbf{t}' \in \mathbb{Z}_+^{\mathcal{C}}$, where $\mathbf{t} \vee \mathbf{t}' := \max\{\mathbf{t}, \mathbf{t}'\}$ and $\mathbf{t} \wedge \mathbf{t}' := \min\{\mathbf{t}, \mathbf{t}'\}$ component-wise. Function f is lattice supermodular if, and only if, $-f$ is lattice submodular.

We provide the corresponding counterexamples in Appendix 8. Proposition 2.5 is in line with the following result presented in Bobbio et al. [2022].

Theorem 2.6 (Bobbio et al. [2022]). *Assume that students have complete preference lists. Then, for any $\varepsilon > 0$, Problem 1 cannot be approximated within a factor of $O(n^{\frac{1-\varepsilon}{2}})$, where n is the number of students, unless $\text{NP}=\text{P}$.*

The aforementioned hardness results motivate us to find exact formulations of the problem and to propose heuristics to obtain near-optimal solutions efficiently. We describe these in the next section.

3 Methodologies

In this section, we present different solution approaches. Specifically, in Section 3.1, we introduce two mathematical programs that out-of-the-box optimization solvers can solve to obtain an exact solution. Due to the potentially high computational time needed to solve these formulations, in Section 3.2 we describe two heuristics that can efficiently find good solutions. In Appendix 15 we provide an empirical comparison of the methods described next.

3.1 Integer Programming Formulations

Recalling that $\mathbf{t}$ is the vector of extra seats allocated to the schools and $B \in \mathbb{Z}^+$ is the total budget of additional seats that can be allocated, we define

$$\mathcal{P} = \left\{ (\mathbf{x}, \mathbf{t}) \in [0, 1]^{\mathcal{E}} \times [0, B]^{\mathcal{C}} : \sum_{c: (s, c) \in \mathcal{E}} x_{s, c} = 1 \text{ for all } s \in \mathcal{S}, \sum_{s \in \mathcal{S}} x_{s, c} \leq q_c + t_c \text{ for all } c \in \mathcal{C}, \sum_{c \in \mathcal{C}} t_c \leq B \right\},$$

representing the set of fractional (potentially non-stable) matchings with capacity expansion. Note that the first condition states that each student must be assigned to one school in $\mathcal{C} \cup \{\emptyset\}$. The second condition sets a bound on the capacity of each school in $\mathcal{C}$, and the last condition establishes a budget for the extra seats to allocate. We denote by $\mathcal{P}_{\mathbb{Z}}$ the integral points of $\mathcal{P}$, i.e., $\mathcal{P}_{\mathbb{Z}} = \mathcal{P} \cap (\{0, 1\}^{\mathcal{E}} \times \{0, \dots, B\}^{\mathcal{C}})$.

Based on $\mathcal{P}_{\mathbb{Z}}$, we can formulate Problem 1 by adapting (2.1) to incorporate the decision vector $\mathbf{t}$. As a result, we obtain the following IQCP formulation:

$$\min_{\mathbf{x}, \mathbf{t}} \sum_{(s, c) \in \mathcal{E}} r_{s, c} \cdot x_{s, c} \tag{3a}$$

$$s.t. \quad (t_c + q_c) \cdot \left(1 - \sum_{c' \succeq_s c} x_{s, c'} \right) \leq \sum_{s' \succ_c s} x_{s', c}, \quad \text{for all } (s, c) \in \mathcal{E}, \tag{3b}$$

$$(\mathbf{x}, \mathbf{t}) \in \mathcal{P}_{\mathbb{Z}}. \tag{3c}$$

Constraint (3b) guarantees that the matching is stable. Note that its relaxed version, i.e., when $(\mathbf{x}, \mathbf{t})$ is allowed to be fractional, is a non-convex quadratic constraint, which makes the problem technically and computationally challenging. Finally, Constraint (3c) establishes the integrality of the decision variables and the matching feasibility.

In an attempt to obtain a mathematical programming model that can be solved efficiently in practice, we linearize the quadratic constraints (3b) through a McCormick envelope McCormick [1976]. In our formulation, the quadratic term $t_c \cdot \sum_{c' \succeq_s c} x_{s, c'}$ in (3b) can be linearized in at least two ways. Specifically,

- *Aggregated Linearization*: For every $(s, c) \in \mathcal{E}$, we define $w_{s,c} := t_c \cdot \sum_{c' \succeq_s c} x_{s,c'}$,
- *Non-Aggregated Linearization*: For every $(s, c) \in \mathcal{E}$ and $c' \succeq_s c$, we define $w_{s,c,c'} := t_c \cdot x_{s,c'}$.

The mixed-integer programming formulation of the McCormick envelope for the aggregated linearization reads as

$$\min_{\mathbf{x}, \mathbf{t}, \mathbf{w}} \sum_{(s,c) \in \mathcal{E}} r_{s,c} \cdot x_{sc} \quad (4a)$$

$$s.t. \quad t_c - w_{s,c} + q_c \cdot \left(1 - \sum_{c' \succeq_s c} x_{s,c'}\right) \leq \sum_{s' \succ_c s} x_{s',c}, \quad \text{for all } (s, c) \in \mathcal{E}, \quad (4b)$$

$$-w_{s,c} + t_c + B \cdot \sum_{c' \succeq_s c} x_{s,c'} \leq B, \quad \text{for all } (s, c) \in \mathcal{E}, \quad (4c)$$

$$w_{s,c} \leq t_c, \quad \text{for all } (s, c) \in \mathcal{E}, \quad (4d)$$

$$w_{s,c} \leq B \cdot \sum_{c' \succeq_s c} x_{s,c'}, \quad \text{for all } (s, c) \in \mathcal{E}, \quad (4e)$$

$$(\mathbf{x}, \mathbf{t}) \in \mathcal{P}_{\mathbb{Z}}, \quad \mathbf{w} \geq 0. \quad (4f)$$

Note that the original stability Constraints (3b) become linear Constraints (4b). Constraints (4c), (4d), (4e) and the non-negativity of the $w_{s,c}$ form the McCormick envelope (see Appendix 10 for a short background). The rest of the constraints remain the same as well as the objective function.

It is well known that whenever at least one of the variables involved in the linearization is binary, the McCormick envelope is exact. Namely, the set of feasible solutions for the original problem coincides with the set of feasible solutions obtained from the McCormick linearization. For the aggregated linearization, we have that $\sum_{c' \succeq_s c} x_{s,c'} \in \{0, 1\}$ due to matching constraints in $\mathcal{P}_{\mathbb{Z}}$.

Corollary 3.1. *The projection of the feasible region given by Constraints (4b)-(4f) in the variables $\mathbf{t}$ and $\mathbf{x}$ coincides with the region given by Constraints (3b) and (3c).*

We now discuss the mixed-integer programming formulation of the McCormick envelope for the non-aggregated linearization. Namely,

$$\min_{\mathbf{x}, \mathbf{t}, \mathbf{w}} \sum_{(s,c) \in \mathcal{E}} r_{s,c} \cdot x_{sc} \quad (5a)$$

$$s.t. \quad t_c - \sum_{c' \succeq_s c} w_{s,c,c'} + q_c \cdot \left(1 - \sum_{c' \succeq_s c} x_{s,c'}\right) \leq \sum_{s' \succ_c s} x_{s',c}, \quad \text{for all } (s, c) \in \mathcal{E}, \quad (5b)$$

$$-w_{s,c,c'} + t_c + B \cdot x_{s,c'} \leq B, \quad \text{for all } (s, c) \in \mathcal{E}, \quad c' \succeq_s c, \quad (5c)$$

$$w_{s,c,c'} \leq t_c, \quad \text{for all } (s, c) \in \mathcal{E}, \quad c' \succeq_s c, \quad (5d)$$

$$w_{s,c,c'} \leq B \cdot x_{s,c'}, \quad \text{for all } (s, c) \in \mathcal{E}, \quad c' \succeq_s c, \quad (5e)$$

$$(\mathbf{x}, \mathbf{t}) \in \mathcal{P}_{\mathbb{Z}}, \quad \mathbf{w} \geq 0. \quad (5f)$$

In this case, Constraints (5c), (5d), (5e) and the non-negativity of the $w_{s,c,c'}$ form the McCormick envelope. Similar to Corollary 3.1, since $x_{s,c'} \in \{0, 1\}$, we obtain the following corollary.

Corollary 3.2. *The projection of the feasible region given by Constraints (5b)-(5f) in the variables $\mathbf{t}$ and $\mathbf{x}$ coincides with the region given by Constraints (3b) and (3c).*

Therefore, both mixed-integer linear programming formulations yield the same set of feasible solutions. Interestingly, the feasible region of the relaxed aggregated linearization, i.e., when $(\mathbf{x}, \mathbf{t}) \in \mathcal{P}_{\mathbb{Z}}$ in (4f) is changed to $(\mathbf{x}, \mathbf{t}) \in \mathcal{P}$, is strictly contained in the feasible region of the relaxed non-aggregated linearization.

Theorem 3.3. *The feasible region of the relaxed aggregated linearization model is contained in the feasible region of the relaxed non-aggregated linearization model.*

Theorem 3.3 implies that the optimal value of the relaxed aggregated linearized model is greater than or equal to the optimal value of the relaxed non-aggregated linearized model. In Appendix 11, we provide a counterexample that shows that the inverse of Theorem 3.3 is not true, i.e., the inclusion in Theorem 3.3 is strict. Since solution approaches to mixed-integer programming formulations are based on the quality of their continuous relaxation, the aggregated linearization dominates the non-aggregated one, thus it is expected to perform better in practice.

3.2 Heuristics

In this section, we present two natural methods: (i) a greedy approach (Greedy), and (ii) an LP-based heuristic (LPH). These two heuristics rely on the computation of a student-optimal stable matching, which can be done in polynomial time using the DA algorithm. For completeness, we include a description of DA in Appendix 9.

Greedy Approach. In Greedy, we explore the fact that the objective function is decreasing in $\mathbf{t}$ and, iteratively, assign an extra seat to the college leading to the largest reduction in the objective. More precisely, Greedy performs B sequential iterations. In each iteration, we evaluate the objective function for each possible allocation of one extra seat using DA. Then, the school leading to the lowest objective receives that extra seat. At the end of this procedure, B extra seats are allocated.

LP-based Heuristic. If we relax the stability constraints, Problem 1 can be formulated as a minimum-cost flow problem whose polytope has integer vertices. Once we enrich this problem with the expansion of capacities, the integrality of the vertices is preserved. Hence, LPH starts by solving the linear program that minimizes Objective (3a), restricted to the set $\mathcal{P}$. Then, using the DA algorithm, we compute the student-optimal stable matching in the new instance that considers the capacity expansion obtained by the linear program.

4 Application to School Choice in Chile

To illustrate the potential benefits of capacity expansion, we adapt our framework to the Chilean school choice system. This system, introduced in 2016 in the southernmost region of the country (Magallanes), was fully implemented in 2020 and serves close to half a million students and more than eight thousand schools each year.

The Chilean school choice system is a good application for our methodology for multiple reasons. First, the system uses a variant of the student-proposing Deferred Acceptance algorithm, which incorporates priorities and overlapping quotas. Our framework can include all the features of the Chilean system, including the block application, the dynamic siblings' priority, etc. We refer to Correa et al. [2022] for a detailed description of the Chilean school choice system and the algorithm used to perform the allocation. Second, the Ministry of Education manages all schools that participate in the system and thus can ask them to modify their vacancies within a reasonable

range. Finally, the system is currently being redesigned, and we are collaborating with the authorities to include some of the ideas introduced in our work.

4.1 Data and Simulation Setting

We consider data from the admission process in 2018. Specifically, we focus on the southernmost region of the country as it is the region where all policy changes are first evaluated. Moreover, we restrict the analysis to Pre-K for two reasons: (1) it is the level with the highest number of applicants, as it is the first entry level in the system, and (2) to speed up computation. In Table 1 (see Appendix 12), we report summary statistics about the instance, and we compare it with the values nationwide for the same year.

We perform our simulations varying the budget $B \in \{0, 1, \dots, 50\}$ and the penalty for unassigned students $r_{s,\emptyset}$. For the latter, we consider two cases: (i) $r_{s,\emptyset} = |\mathcal{C}| + 1$ for all $s \in \mathcal{S}$, and (ii) $r_{s,\emptyset} = |\succ_s| + 1$ for all $s \in \mathcal{S}$. Notice that the two values for $r_{s,\emptyset}$ cover two extreme cases. When $r_{s,\emptyset} = |\mathcal{C}| + 1$ (or any large number), the model will use the extra vacancies to ensure that a student that was previously unassigned gets assigned. In contrast, when $r_{s,\emptyset} = |\succ_s| + 1$, the model will (most likely) assign the extra seat to the school that leads to the largest chain of improvements. Hence, from a practical standpoint, which penalty to use is a policy-relevant decision that must balance *fairness* and *access*. We provide our results in Section 4.2 and in Appendix 12.

4.2 Results

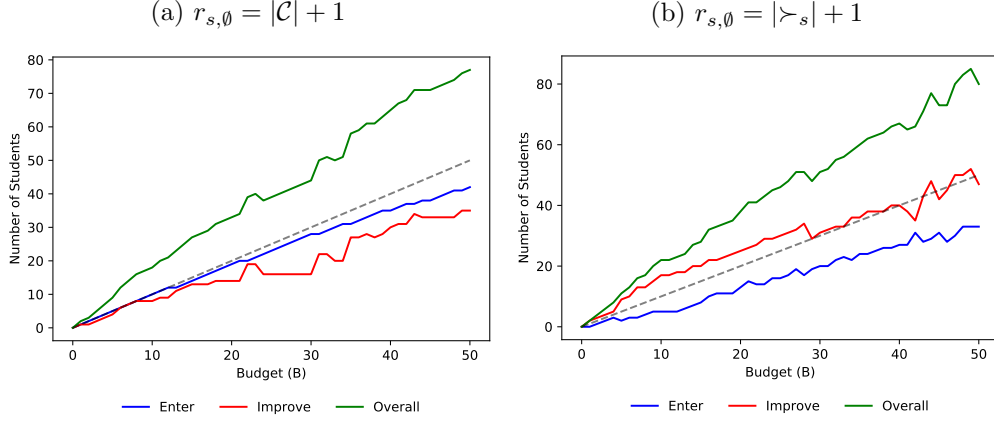
We report our main simulation results in Figure 1. For each budget, we plot the number of students who (1) *enter* the system, i.e., who are not initially assigned (with $B = 0$), but are assigned to one of their preferences when capacities are expanded; (2) *improve*, i.e., students who are initially assigned to some preference but improve their preference of assignment when capacities are expanded; and (3) *overall*, which is the total number of students who benefit relative to the baseline and is equal to the sum of the number of students who *enter* and *improve*.⁶

First, we confirm that all initially assigned students (with $B = 0$) get a school at least as preferred when we expand capacities. Second, increasing capacities with a high penalty primarily benefits initially unassigned students. In contrast, students who improve their assignments are the ones who most benefit when the penalty is low. Third, we observe that the total number of students who benefit (in green) is considerably larger than the number of additional seats (dashed). The reason is that an extra seat can lead to a chain of improvements that ends either on a student that *enters* the system or in a school that is under-demanded. Finally, we observe that the total number of students who gain from the additional seats is not strictly increasing in the budget. Indeed, the number may decrease if the extra seat allows a student to dramatically improve their assignment (e.g., moving from being unassigned to assigned to their top preference). This effect on the objective could be larger than that of a chain of minor improvements involving several students, and thus the number of students who benefit may decrease.

In Figure 2, we analyze the impact of expanding capacities on the number of schools with increased capacity and the maximum number of additional seats per school. We observe that the number of schools with extra seats remains relatively stable as we increase the budget. In addition, we observe that the maximum number of additional seats in a given school increases with the budget. This is because students' preferences are highly correlated (i.e., students have similar preferences),

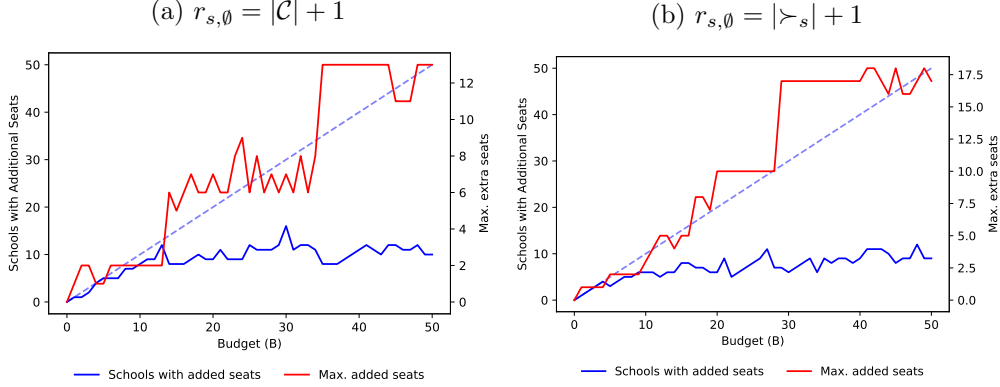
⁶For the case with $r_{s,\emptyset} = |\succ_s| + 1$, we consider a MipGap tolerance of 0.1% starting from $B = 35$. The reason is that the computational time increases considerably for larger budgets.

Figure 1: Effect for Students



and thus, a few over-demanded schools concentrate the extra seats added to the system. Finally, the latter effect is more prominent when the penalty is lower.

Figure 2: Effect for Schools



5 Conclusions

We studied how centralized clearinghouses can jointly decide how to allocate additional seats and find a stable matching. To accomplish this: we introduced a model of stable matching under capacity planning, and we provided mathematical programming formulations to solve the problem. Given the complexity of the problem Bobbio et al. [2022], we proposed two heuristics that find close-to-optimal solutions efficiently. Finally, we adapted our framework to solve an instance of the Chilean school choice problem. Our results show that each additional seat can benefit multiple students. However, depending on how we penalize having unassigned students in the objective, the set of students who benefit from the extra seats changes. Indeed, we have shown that if we consider a large penalty, the optimal solution prioritizes *access*, i.e., assigning students that were previously unassigned. In contrast, if that penalty is low, the optimal solution prioritizes *fairness*, i.e., improving as much as possible students' preference of assignment. Hence, which penalty to consider is a policy-relevant decision that depends on the objective of the clearinghouse.

References

- A. Abdulkadiroğlu and T. Sönmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, may 2003.
- A. Abdulkadiroğlu, P. A. Pathak, A. E. Roth, and T. Sönmez. The boston public school match. *American Economic Review*, 95(2):368–371, apr 2005.
- A. Abdulkadiroğlu, P. A. Pathak, and A. E. Roth. Strategy-proofness versus efficiency in matching with indifference: Redesigning the NYC high school match. *American Economic Review*, 99(5):1954–1978, dec 2009.
- A. Abdulkadiroğlu, Y.-K. Che, and Y. Yasuda. Resolving conflicting preferences in school choice: The “boston mechanism” reconsidered. *American Economic Review*, 101(1):399–410, feb 2011.
- K. Abe, J. Komiyama, and A. Iwasaki. Anytime capacity expansion in medical residency match by monte carlo tree search. *arXiv preprint arXiv:2202.06570*, 2022.
- K. C. Ágoston, P. Biró, and I. McBride. Integer programming methods for special college admissions problems. *Journal of Combinatorial Optimization*, 32(4):1371–1399, 2016.
- K. C. Ágoston, P. Biró, E. Kováts, and Z. Jank. College admissions with ties and common quotas: Integer programming approach. *European Journal of Operational Research*, 2021. ISSN 0377-2217.
- N. Arnosti. Short lists in centralized clearinghouses. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. ACM, jun 2015.
- I. Ashlagi and P. Shi. Optimal allocation without money: An engineering approach. *Management Science*, 62(4):1078–1097, apr 2016.
- I. Ashlagi, A. Nikzad, and A. Romm. Assigning more students to their top choices: A comparison of tie-breaking rules. *Games and Economic Behavior*, 115:167–187, may 2019.
- M. Baïou and M. Balinski. The stable admissions polytope. *Mathematical programming*, 87(3):427–439, 2000.
- M. Balinski and T. Sönmez. A tale of two mechanisms: student placement. *Journal of Economic theory*, 84(1):73–94, 1999.
- F. Bobbio, M. Carvalho, A. Lodi, and A. Torrico. Capacity variation in the many-to-one stable matching problem. 2022.
- A. L. Bodoh-Creed. Optimizing for distributional goals in school choice problems. *Management Science*, 66(8):3657–3676, aug 2020.
- C. Calsamiglia and M. Güell. Priorities in school choice: The case of the boston mechanism in barcelona. *Journal of Public Economics*, 163:20–36, 2018.
- F. Caro, T. Shirabe, M. Guignard, and A. Weintraub. School redistricting: embedding GIS tools with integer programming. *Journal of the Operational Research Society*, 55(8):836–849, aug 2004.
- J. Correa, N. Epstein, R. Epstein, J. Escobar, I. Rios, N. Aramayo, B. Bahamondes, C. Bonet, M. Castillo, A. Cristi, B. Epstein, and F. Subiabre. School choice in chile. *Operations Research*, 70(2):1066–1087, mar 2022. doi: 10.1287/opre.2021.2184.
- M. Delorme, S. García, J. Gondzio, J. Kalcsics, D. Manlove, and W. Pettersson. Mathematical models for stable matching problems with ties and incomplete lists. *European Journal of Operational Research*, 277(2):426–441, 2019.

- L. E. Dubins and D. A. Freedman. Machiavelli and the gale-shapley algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981.
- L. Ehlers. School choice with control. 2010.
- D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
- D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11(3):223–232, 1985.
- D. Gusfield and R. W. Irving. *The stable marriage problem: structure and algorithms*. MIT press, 1989.
- I. E. Hafalir, M. B. Yenmez, and M. A. Yildirim. Effective affirmative action in school choice. *Theoretical Economics*, 8(2):325–363, may 2013.
- A. S. Kelso Jr and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society*, pages 1483–1504, 1982.
- F. Kojima, A. Tamura, and M. Yokoo. Designing matching mechanisms under constraints: An approach from discrete convex analysis. *Journal of Economic Theory*, 176:803–833, 2018.
- S. D. Kominers. Respect for improvements and comparative statics in matching markets. Technical report, Working Paper, 2019.
- R. Kurata, N. Hamada, A. Iwasaki, and M. Yokoo. Controlled school choice with soft bounds and overlapping types. *Journal of Artificial Intelligence Research*, 58(1):153–184, jan 2017.
- A. Kwanashie and D. F. Manlove. An integer programming approach to the hospitals/residents problem with ties. In *Operations research proceedings 2013*, pages 263–269. Springer, 2014.
- D. Manlove. *Algorithmics of matching under preferences*, volume 2. World Scientific, 2013.
- G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I - convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.
- T. Nguyen and R. Vohra. Near-feasible stable matchings with couples. *American Economic Review*, 108(11):3154–69, 2018.
- P. A. Pathak and T. Sönmez. Leveling the playing field: Sincere and sophisticated players in the boston mechanism. *American Economic Review*, 98(4):1636–1652, aug 2008.
- I. Rios, T. Larroucau, G. Parra, and R. Cominetti. Improving the chilean college admissions system. *Management Science*, 69(4):1186–1205, July-August 2021.
- A. Romm. Implications of capacity reduction and entry in many-to-one stable matching. *Social Choice and Welfare*, 43(4):851–875, 2014.
- A. E. Roth. The economics of matching: Stability and incentives. *Mathematics of operations research*, 7(4):617–628, 1982.
- A. E. Roth. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70(4):1341–1378, July 2002.
- A. E. Roth and M. A. O. Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge Univ. Press, Cambridge, MA, 1990.
- A. E. Roth, U. G. Rothblum, and J. H. Vande Vate. Stable matchings, optimal assignments, and linear programming. *Mathematics of operations research*, 18(4):803–828, 1993.

- U. G. Rothblum. Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, 54(1):57–67, 1992.
- P. Shi. Assortment planning in school choice. 2016.
- T. Soma and Y. Yoshida. A generalization of submodular cover via the diminishing return property on the integer lattice. *Advances in neural information processing systems*, 28, 2015.
- T. Sönmez. Manipulation via capacities in two-sided matching markets. *Journal of Economic theory*, 77(1):197–204, 1997.
- T. Sonmez and M. Yenmez. Affirmative action with overlapping reserves. 2019.
- J. H. V. Vate. Linear programming brings marital bliss. *Operations Research Letters*, 8(3):147–153, 1989.

Appendix

To ease the exposition, we avoid using the symbol $\succ$ when presenting a preference list, instead we simply separate agents by “,” and use the convention that the leftmost agents are the most preferred. For instance, we will represent the preference list $w \succ w' \succ w''$ as w, w', w'' . The “...” at the end of a preference list will be used to represent an arbitrary completion of the preferences.

6 Proof of Lemma 2.1

We begin by noting that Baïou and Balinski [2000] show that the feasible region of the Integer Program (1) corresponds to the set of stable matchings. Therefore, in the following proof, we only need to focus on proving the equivalence between the student-optimal matching and a matching minimizing the sum of the student’s rank over the set of stable matchings.

If: In a student-optimal stable matching, each student is assigned to the best school they could achieve in any stable matching Gale and Shapley [1962]. Thus, each unassigned student is unassigned in every stable matching. Moreover, by the Rural Hospital Theorem, the same students are assigned in all stable matchings. Suppose that μ is the student-optimal stable matching but its corresponding binary encoding is not optimal to the Integer Program (1). Let x' be an optimal solution to the Integer Program (1) and let $\mu' = \{(s, c) \in \mathcal{E} : x'_{s,c} = 1\}$ be the associated matching. Hence, the following inequality holds: $\sum_{(s,c) \in \mu} r_{s,c} > \sum_{(s,c) \in \mu'} r_{s,c}$. This means that there is at least one student s' who prefers the stable matching μ' to the stable matching μ , which is a contradiction.

Only if: Let μ be a (stable) matching corresponding to an optimal solution of the Integer Program (1). As before, by the Rural Hospital Theorem, we observe that the set of students unassigned in μ is the same set of students unassigned in every stable matching. Hence, the set of assigned students in μ , is the same set for every stable matching. Let us suppose, again by contradiction, that μ is not a student-optimal stable matching. Let μ' be a student-optimal stable matching. Denote by S' the set of students whose assignment to schools differs in the two matchings. By construction, the objective value of the Integer Program (1) for $\mathcal{S} \setminus S'$ is the same in both μ and μ' :

$$\sum_{(s,c) \in \mu: s \in \mathcal{S} \setminus S'} r_{s,c} = \sum_{(s,c) \in \mu': s \in \mathcal{S} \setminus S'} r_{s,c}.$$

Furthermore, S' is the disjoint union of the following two sets of students: S'_1 , the set of assigned students who prefer their school in μ' , and S'_2 , the set of assigned students who prefer their school in μ . By Gale and Shapley [1962], μ' is a stable matching in which all students are assigned the best

school they could achieve in any stable matching. Therefore, the set S'_2 is empty. Hence, $S' = S'_1$, and by hypothesis

$$\sum_{(s,c) \in \mu: s \in S'_1} r_{s,c} < \sum_{(s,c) \in \mu': s \in S'_1} r_{s,c}.$$

However, from the definition of S'_1 , $r_{s,\mu'(s)} \leq r_{s,\mu(s)}$ for every $s \in S'_1$, which leads to a contradiction. $\square$

7 Missing Proofs in Section 2.1

Example 7.1. Consider an instance with three schools $\mathcal{C} = \{c_1, c_2, c_3\}$, four students $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, and capacities $q_{c_1} = q_{c_2} = 1$, $q_{c_3} = 2$. In addition, consider preferences given by:

$$\begin{aligned} c &: s_1 \succ s_2 \succ s_3 \succ s_4, \forall c \in \mathcal{C} \\ s_1 &: c_1 \succ c_2 \succ c_3 \\ s_2 &: c_2 \succ c_1 \succ c_3 \\ s_3 &: c_1 \succ c_3 \succ c_2 \\ s_4 &: c_2 \succ c_3 \succ c_1. \end{aligned}$$

Notice that, with no capacity expansion, the student-optimal stable matching is

$$\mu^* = \{(s_1, c_1), (s_2, c_2), (s_3, c_3), (s_4, c_3)\},$$

which leads to an objective of 6. On the other hand, if we have a budget $B = 1$, note that we can allocate it to either c_1 and obtain the matching $\mu' = \{(s_1, c_1), (s_2, c_2), (s_3, c_1), (s_4, c_3)\}$, or to school c_2 and obtain the matching $\mu'' = \{(s_1, c_1), (s_2, c_2), (s_3, c_3), (s_4, c_2)\}$. In both cases, one student moves from their second choice to their top choice, and thus in both cases the sum of preferences of assignment is 5. Hence, we conclude that this problem has more than one optimal solution.

Proof of Proposition 2.4. Consider $\mathcal{S} = \{s_1, s_2, s_3\}$ and $\mathcal{C} = \{c_1, c_2\}$, with $q_c = 1$ for all c . In addition, consider the following preferences and priorities:

$$\begin{aligned} s_1 &: c_1 & c_1 &: s_1, s_2, s_3 \\ s_2 &: c_1, c_2 & c_2 &: s_2, s_1, s_3 \\ s_3 &: c_1 \end{aligned}$$

Then, if $B = 1$, the extra seat will go to c_1 , and s_3 will remain unassigned. $\square$

Proof of Theorem 2.3. Let μ^* be the stable-matching corresponding to the solution $(\mathbf{x}^*, \mathbf{t}^*)$. To find a contradiction, suppose there exists another stable matching μ' that has a higher cardinality, i.e., $|s \in \mathcal{S} : \mu'(s) \neq \emptyset| < |s \in \mathcal{S} : \mu^*(s) \neq \emptyset|$.

By optimality of $(\mathbf{x}^*, \mathbf{t}^*)$, we know that

$$\sum_{s \in \mathcal{S}} r_{s,\mu^*(s)} < \sum_{s \in \mathcal{S}} r_{s,\mu'(s)}.$$

On the other hand, we know that

$$\begin{aligned}
\sum_{s \in \mathcal{S}} r_{s, \mu^*(s)} - \sum_{s \in \mathcal{S}} r_{s, \mu'(s)} &= \sum_{s \in \mathcal{S}: \mu^*(s) \in \mathcal{C}} r_{s, \mu^*(s)} - \sum_{s \in \mathcal{S}: \mu'(s) \in \mathcal{C}} r_{s, \mu'(s)} + \sum_{s \in \mathcal{S}: \mu^*(s) = \emptyset} r_{s, \emptyset} - \sum_{s \in \mathcal{S}: \mu'(s) = \emptyset} r_{s, \emptyset} \\
&= \sum_{s \in \mathcal{S}: \mu^*(s) \in \mathcal{C}} r_{s, \mu^*(s)} - \sum_{s \in \mathcal{S}: \mu'(s) \in \mathcal{C}} r_{s, \mu'(s)} \\
&\quad + \bar{r} \cdot [|\{s \in \mathcal{S} : \mu^*(s) = \emptyset\}| - |\{s \in \mathcal{S} : \mu'(s) = \emptyset\}|] \\
&> - \sum_{s \in \mathcal{S}} |\succ_s| + \bar{r} \cdot [|\{s \in \mathcal{S} : \mu^*(s) = \emptyset\}| - |\{s \in \mathcal{S} : \mu'(s) = \emptyset\}|] \\
&> 0
\end{aligned} \tag{7}$$

The first equality follows from assuming that $r_{s, \emptyset} = \bar{r}$ for all $s \in \mathcal{S}$. The first inequality follows from the fact that, given a student s that is assigned, the maximum improvement is to move from their last preference, $|\succ_s|$, to their top preference, and therefore $r_{s, \mu^*(s)} - r_{s, \mu'(s)} > 1 - |\succ_s| > -|\succ_s|$. Then, we have that $\sum_{s \in \mathcal{S}: \mu^*(s) \in \mathcal{C}} r_{s, \mu^*(s)} - \sum_{s \in \mathcal{S}: \mu'(s) \in \mathcal{C}} r_{s, \mu'(s)} \geq -\sum_{s \in \mathcal{S}} |\succ_s|$. Finally, the last inequality follows from the fact that

$$|\{s \in \mathcal{S} : \mu^*(s) = \emptyset\}| - |\{s \in \mathcal{S} : \mu'(s) = \emptyset\}| \geq 1$$

and that $\bar{r}$ is arbitrarily large, and thus we can take a value $\bar{r} > \sum_{s \in \mathcal{S}} |\succ_s|$. As a result, we obtain that

$$\sum_{s \in \mathcal{S}} r_{s, \mu^*(s)} - \sum_{s \in \mathcal{S}} r_{s, \mu'(s)} > 0,$$

which contradicts the optimality of $(\mathbf{x}^*, \mathbf{t}^*)$. $\square$

We now focus on the proof of Theorem 2.2. Towards this goal, we first prove some lemmas. Recall that the student-optimal DA algorithm (see Appendix 9) does not depend on the order that the students apply to schools. For the remainder of this section, consider a fix ordering $s_1, \dots, s_{|\mathcal{S}|}$. Now, we analyze the effect of an additional seat to the assignment of the students. Given an additional seat in some school, we define the *chain of improvements* as the sequence of students who change their assignment (i.e., either improve or enter) with respect to the instance without the additional seat. This sequence of students is determined by the ordering in which the students apply in the DA algorithm. Note that this sequence is optimal for the students since we are considering the student-optimal DA method. Also, observe that an additional seat in a different school may lead to a different chain of improvements. Before proving Theorem 2.2, our goal is to show some properties of these chains of improvements.

Given a budget $B \geq 0$ and a feasible allocation $\mathbf{t} \in \mathbb{Z}_+^{\mathcal{C}}$, a school $c \in \mathcal{C}$ is said to be over-demanded in an assignment μ if $|\mu(c)| = q_c + t_c$ and there exists a student s' such that $c \succ_{s'} \mu(s')$. A school is under-demanded if it is not over-demanded.

Lemma 7.2. *Any chain of improvements ends in a student that was initially unassigned or in a student who was previously assigned to an under-demanded school.*

Proof. Denote by μ the initial matching for $B = 0$. Suppose that the lemma is not true, i.e., there exists a chain of improvements that ends in a student $s \in \mathcal{S}$ that was assigned in μ to an over-demanded school $c \in \mathcal{C}$. Since s is part of the chain of improvements, we know that s is assigned to a preferred school $c' \succ_s c$ in the new assignment. Given that c was over-demanded in μ , we know there exists a student $s' \in \mathcal{S}$ such that $c \succ_{s'} \mu(s')$ and the seat that was previously used by s in c is available. Thus, student s' would be part of the chain of improvements. This is a contradiction, since we assumed that the chain ended with student s . $\square$

Lemma 7.3. *Any chain of improvements reaches at most one initially unassigned student.*

Proof. Since every student in a chain of improvements uses only one seat in some school, we know the seat that the student frees can be used by only one student, who in turn liberates only one seat, and so on. Then, by Lemma 7.2, we know that the chain continues the same way until it reaches an under-demanded school or an initially unassigned student. In both cases, the chain ends in at most one previously unassigned student. $\square$

Now, our goal is to prove that B additional seats lead to at most B chains of improvements. Given a budget $B > 0$, we denote by $(c_1, \dots, c_B)$ the sequence of additional seats (possibly with repetition) where in each step $\ell \in \{1, \dots, B\}$ we allocate an additional seat to school $c_\ell \in \mathcal{C}$ and then we use the DA algorithm to find the student-optimal stable matching considering that extra seat.

Lemma 7.4. *Fix a budget $B > 0$. Any permutation of a given sequence of additional seats $(c_1, \dots, c_B)$ leads to the same final matching. Moreover, a sequence of B additional seats generates at most B chains of improvements.*

Proof. Note that any permutation of a sequence of additional seats $(c_1, \dots, c_B)$ leads to the same allocation vector $\mathbf{t} \in \mathbb{Z}_+^{\mathcal{C}}$ with $t_c = |\{c' \in (c_1, \dots, c_B) : c' = c\}|$. Since the outcome of the DA algorithm is independent on the order in which students are processed, then it is irrelevant the order in which the sequence of additional seats is processed. In fact, this is equivalent to using the DA algorithm in the expanded instance $\Gamma_{\mathbf{t}}$.

Let us prove now the second part of the lemma. By using Lemmas 7.2 and 7.3 we know that a chain of improvements starts with a student and ends in either an under-demanded school or an initially unassigned student. Then, each additional seat in a sequence $(c_1, \dots, c_B)$ generates a single chain of improvements, and since we know that at most B additional seats will be allocated, we know that there will be at most B chains of improvements. $\square$

Proof of Theorem 2.2. For the purpose of the proof, let us consider an arbitrary optimal allocation $\mathbf{t}^*$. Let $(c_1^*, \dots, c_B^*)$ be any sequence of additional seats defined by $\mathbf{t}^*$. First, note that a chain of improvements will never start in a student s that was initially unassigned because $r_{s, \emptyset} < r_{s, c}$ for all $c \neq \emptyset$. Hence, each additional seat c_ℓ^* generates a chain of improvements that starts with an initially assigned student. In addition, from Lemma 7.3 we know that each additional seat c_ℓ^* creates a chain that reaches at most one initially unassigned student. These two observations above imply that each additional seat generates a chain in which the number of students that improve in that chain is at least the number of students who enter in that chain. Finally, from Lemma 7.4 we know that $(c_1^*, \dots, c_B^*)$ generates at most B chains of improvements. Thus, by summing over all the chains the number students who enter and improve, we conclude that $|\mathcal{S}_I(B)| \geq |\mathcal{S}_E(B)|$. $\square$

8 Counter-examples for Proposition 2.5

Recall function $f : \mathbb{Z}_+^{\mathcal{C}} \rightarrow \mathbb{R}_+$ defined for a given $\mathbf{t} \in \mathbb{Z}_+^{\mathcal{C}}$ as

$$f(\mathbf{t}) = \min \left\{ \sum_{(s, c) \in \mu} r_{s, c} : \mu \text{ is a stable matching in instance } \Gamma_{\mathbf{t}} \right\}.$$

Non submodularity. Consider a set $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ of students and a set $\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5\}$ of schools. The preference lists are as follows

$$\begin{array}{ll} s_1 : c_1, c_2, \dots & c_1 : s_1, s_2, \dots \\ s_2 : c_2, c_3, \dots & c_2 : s_2, s_3, \dots \\ s_3 : c_2, c_5, c_4, \dots & c_3 : s_2, s_1, \dots \\ s_4 : c_5, \dots & c_4 : s_3, s_4, \dots \\ & c_5 : s_4, s_3, \dots \end{array}$$

School c_1 has capacity 0 and the other schools have capacity 1. We choose the following two allocations: $\mathbf{t} = (1, 0, 0, 0, 0)$ and $\mathbf{t}' = (0, 1, 0, 0, 0)$. Therefore, we obtain

$$f(\mathbf{t} \vee \mathbf{t}') + f(\mathbf{t} \wedge \mathbf{t}') = f(1, 1, 0, 0, 0) + f(0, 0, 0, 0, 0) = 4 > 3 = 2 + 1 = f(\mathbf{t}) + f(\mathbf{t}').$$

Non super-modularity. Consider a set $\mathcal{S} = \{s_1, s_2, s_3\}$ of students and a set $\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5\}$ of schools. The preference lists are as follows

$$\begin{array}{lll} s_1 : c_1, c_3, \dots & c_h : s_1, s_2, \dots & \text{for all } h \in \{1, 2, 3\}. \\ s_2 : c_2, c_4, \dots & & \\ s_3 : c_3, c_4, c_5, \dots & & \\ s_4 : c_5, \dots & & \end{array}$$

Schools c_1 and c_2 have capacity 0 and the other schools have capacity 1. We choose the following two allocations: $\mathbf{t} = (1, 0, 0, 0, 0)$ and $\mathbf{t}' = (0, 1, 0, 0, 0)$. Therefore, we obtain

$$f(\mathbf{t} \vee \mathbf{t}') + f(\mathbf{t} \wedge \mathbf{t}') = f(1, 1, 0, 0, 0) + f(0, 0, 0, 0, 0) = 4 < 5 = 3 + 2 = f(\mathbf{t}) + f(\mathbf{t}').$$

9 The Deferred Acceptance Algorithm

In this section, we recall the Deferred Acceptance algorithm introduced in Gale and Shapley [1962].

INPUT: An instance $\Gamma = \langle \mathcal{S}, \mathcal{C}, \succ, \mathbf{q} \rangle$.

OUTPUT: Student-oriented matching.

Step 1: Each student starts by applying to her most preferred school. Schools temporarily accept the most preferred applications and reject the less preferred applications which exceed their capacity.

Step 2: Each student s who has been rejected, proposes to her most preferred school to which she has not applied yet; if she has proposed to all schools, then she does not apply. If the capacity of the school is not met, then her application is temporarily accepted. Otherwise, if the school prefers her application to one of a student s' who was temporarily enrolled, s is temporarily accepted and s' is rejected. Vice-versa, if the school prefers all the students temporarily enrolled to s , then s is rejected.

Step 3: If all students are enrolled or have applied to all the schools they rank, return the current matching. Otherwise, go to Step 2.

10 McCormick linearization

In this section we describe the McCormick convex envelope McCormick [1976] used to obtain a linear relaxation for bi-linear terms; if one of the terms is binary, the linearization provides an equivalent formulation. Consider a bi-linear term of the form $x_i \cdot x_j$ with the following bounds for the variables x_i and x_j : $l_i \leq x_i \leq u_i$ and $l_j \leq x_j \leq u_j$. Let us define $y = x_i \cdot x_j$, $m_i = (x_i - l_i)$, $m_j = (x_j - l_j)$, $n_i = (u_i - x_i)$ and $n_j = (u_j - x_j)$. Note that $m_i \cdot m_j \geq 0$, from which we derive the under-estimator $y \geq x_i \cdot l_j + x_j \cdot l_i - l_i \cdot l_j$. Similarly, it holds that $n_i \cdot n_j \geq 0$, from which we derive the under-estimator $y \geq x_i \cdot u_j + x_j \cdot u_i - u_i \cdot u_j$. Analogously, over-estimators of y can be defined. Make $o_i = (u_i - x_i)$, $o_j = (u_j - x_j)$, $p_i = (x_i - l_i)$ and $p_j = (x_j - l_j)$. From $o_i \cdot o_j \geq 0$ we obtain the over-estimator $y \leq x_i \cdot l_j + x_j \cdot u_i - u_i \cdot l_j$, and from $p_i \cdot p_j \geq 0$ we obtain the over-estimator $y \leq x_j \cdot l_i + x_i \cdot u_j - u_j \cdot l_i$. The four inequalities provided by the over and under estimators of y , define the McCormick convex (relaxation) envelope of $x_i \cdot x_j$.

11 Theorem 3.3

In this section we present the proof of Theorem 3.3, and a counter-example to the inverse inclusion of the statement in Theorem 3.3.

Proof of Theorem 3.3. The constraints that do not involve the linearization terms $w_{s,c}$ or $w_{s,c,c'}$ are trivially satisfied by a feasible solution in both formulations. Therefore, we will restrict our analysis to the remaining constraints. Let $(\mathbf{x}, \mathbf{t}, \mathbf{w})$ be a feasible solution of the relaxed aggregated linearized program. It is easy to verify that by defining $\bar{w}_{s,c,c'} = w_{s,c} \cdot x_{s,c'}$ for every $s \in \mathcal{S}$, $c \in \mathcal{C}$ and $c' \succeq_s c$, the constraints of the relaxed non-aggregated linearization are all met. $\square$

Example 11.1. Let us consider the set of students $\mathcal{S} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ and the set of schools $\mathcal{C} = \{c_1, c_2, c_3, c_4\}$. The rankings of the students are as follows:

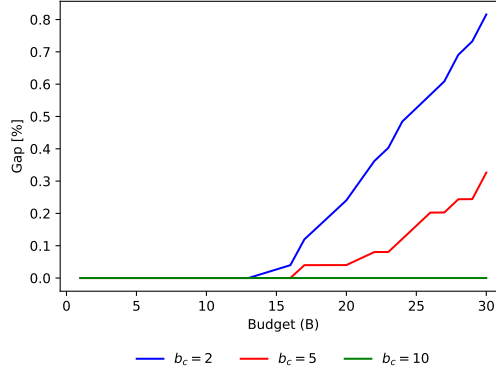
$$\begin{array}{ll} s_1 : c_3, c_4, c_1, c_2 & c_1 : s_1, s_3, s_2, s_5, s_6, s_4 \\ s_2 : c_2, c_1, c_4, c_3 & c_2 : s_4, s_1, s_6, s_5, s_2, s_3 \\ s_3 : c_2, c_1, c_4, c_3 & c_3 : s_2, s_1, s_5, s_6, s_3, s_4 \\ s_4 : c_1, c_3, c_2, c_4 & c_4 : s_6, s_3, s_2, s_4, s_5, s_1 \\ s_5 : c_3, c_1, c_4, c_2 & \\ s_6 : c_1, c_3, c_4, c_2. & \end{array}$$

Finally, schools' capacities are $q_{c_1} = q_{c_2} = 1$ and $q_{c_3} = q_{c_4} = 2$. Given that we have to allocate optimally one extra position, the optimal solution for the relaxed aggregated linearization is $x_{s_1 c_3} = 1, x_{s_2 c_2} = 1, x_{s_3 c_1} = 0.16, x_{s_3 c_2} = 0.66, x_{s_4 c_1} = 0.5, x_{s_4 c_2} = 0.16, x_{s_4 c_3} = 0.16, x_{s_5 c_3} = 0.83, x_{s_6 c_1} = 0.5, x_{s_6 c_4} = 0.33$ and $\mathbf{t} = (0.16, 0.83, 0, 0)$ with the cost equal to 1.33.

On the other hand, the optimal solution for the relaxed non-aggregated linearization is $x_{s_1 c_3} = 0.83, x_{s_1 c_4} = 0.08, x_{s_2 c_2} = 1, x_{s_3 c_1} = 0.11, x_{s_3 c_2} = 0.66, x_{s_3 c_4} = 0.02, x_{s_4 c_1} = 0.66, x_{s_4 c_3} = 0.16, x_{s_5 c_3} = 1, x_{s_6 c_1} = 0.52, x_{s_6 c_4} = 0.30$ and $\mathbf{t} = (0.30, 0.66, 0, 0.03)$ with cost equal to 1.03.

Experimental results suggest the following hypothesis: Under the additional constraint that for every student s , $\sum_{c \in \mathcal{C}} x_{s,c} \geq 1$ holds (i.e., every student is assigned at least to a school), the inverse of Theorem 3.3 holds.

Figure 3: Bounds on Expansion



12 Further Considerations on the Chilean Data

Table 1: Instance for Evaluation

Region	Students	Schools	Applications
Magallanes Pre-K	1389	43	4483
Overall Pre-K	84626	3465	256120
Overall (all levels)	274990	6421	874565

In this section, we present extra computational results that we obtain using the Chilean school choice dataset.

12.1 Practical Implementation.

A valid concern from policy-makers is that our approach would assign most extra seats to a small number of over-demanded schools. As a result, our solution would not be feasible in practice since classroom capacities limit the possible expansion. To rule out this concern, we adapt our model and include the new set of constraints

$$t_c \leq b_c, \quad \text{for all } c \in \mathcal{C},$$

where b_c is the maximum number of additional seats that can be allocated to school c .

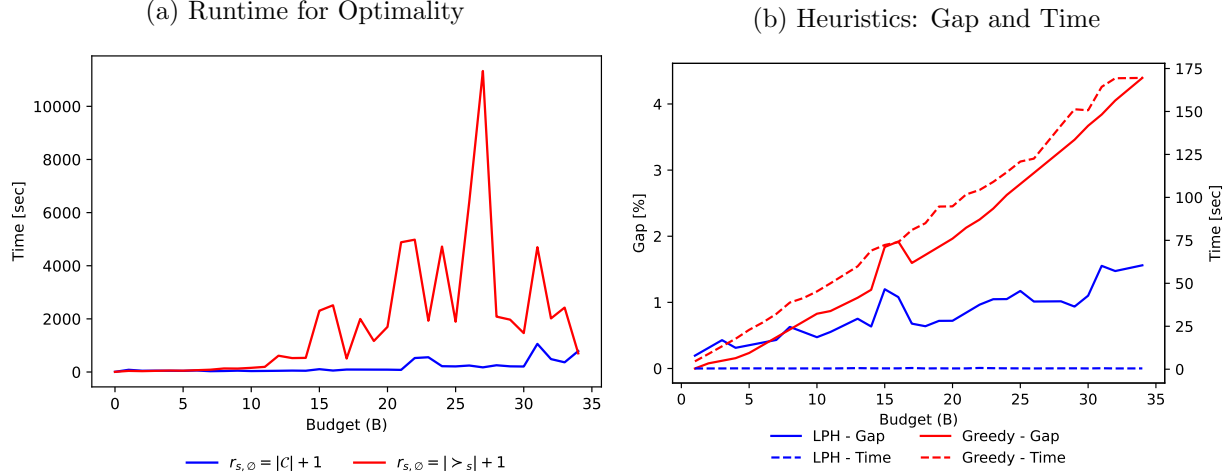
In Figure 3, we compare the gap between the optimal (unconstrained) solution and the values obtained when considering $b_c \in \{2, 5, 10\}$ for all $c \in \mathcal{C}$. First, we observe that the gap increases for $b_c \in \{2, 5\}$ as we increase the budget, while it does not change for $b_c = 10$. This result suggests that the problem has many optimal solutions, and thus we can select one that does not over-expand some schools. Second, we observe that the overall gap is relatively low (max of 0.8%), which suggests that we can include the practical limitations for schools without major loss of performance. In Appendix 13, we discuss some model extensions to incorporate other relevant aspects from a practical standpoint.

12.1.1 Heuristics.

In Figure 4a, we plot the time to achieve zero optimality gap for $B \in \{0, \dots, 35\}$. We observe that the computational time for the case with $r_{s,\emptyset} = |\succ_s| + 1$ is considerably larger than that for

$r_{s,\emptyset} = |\mathcal{C}| + 1$. Moreover, we observe that the solution time for $r_{s,\emptyset} = |\succ_s| + 1$ can be considerably large for some budget values, and this problem may be exacerbated if we consider larger instances (e.g., larger regions or the entire country). For this reason, it is relevant to study the performance of the heuristics described in Section 3.2.

Figure 4: Heuristics



For each value of the budget, in Figure 4b we report the gap obtained relative to the optimal policy⁷ and the execution time of each heuristic. As shown in our simulation results, Greedy performs better than LPH for low values of the budget, but this reverses as the budget increases. In addition, we observe that the time of LPH is significantly better than that of Greedy. Hence, we conclude that LPH can be an effective approach for large instances and large values of B .

13 Extensions

Our model can be easily extended to capture several relevant variants of the problem. In what follows we name some direct extensions:

- Adding cost for capacity expansion: if there is a unit-cost p_c of increasing the capacity of school c , this can be included in the objective function and all the other elements of the model remain the same.
- Adding secured enrollment: the Chilean system guarantees that students that are currently enrolled and apply to switch will be assigned to their current school if they are not assigned to a more preferred one. This can be easily captured in our setting by introducing a parameter $m_{s,c}$, which is equal to 1 if student s is currently enrolled in school c , and 0 otherwise, and defining $M = \{s \in S : m_{s,c} = 1 \text{ for some } c \in C\}$. Then, we would only have to update a couple of constraints:

$$\begin{aligned} \sum_{s \in S} x_{s,c} \cdot (1 - m_{s,c}) &\leq q_c + t_c, \quad \text{for all } c \in C \\ \sum_{c \in C} x_{s,c} &= 1, \quad \text{for all } s \in M. \end{aligned} \tag{11}$$

⁷We consider the problem with penalty $r_{s,\emptyset} = |\succ_s| + 1$. The results are similar if we consider $r_{s,\emptyset} = |\mathcal{C}| + 1$

The first constraint ensures that students currently enrolled do not count towards the capacity of the school they are currently enrolled. The second constraint ensures that all students that are currently enrolled are assigned to some schools (potentially, to the same school they are currently enrolled).

- Room assignment: schools report the number of vacancies they have for each level. This decision depends on the classrooms they have and their capacity. However, schools decide (before the assignment) what level goes in each classroom, and this determines the number of reported vacancies for that level. This may introduce some inefficiencies, since some levels may be more demanded, and thus assigning a larger classroom may benefit both students in that school but also in others.
- Quota assignment: many school choice systems have different quotas to serve under-represented students or special groups. For instance, in Chile there are quotas for low-income students (15% of total seats), for students with disabilities or special needs, and for students with high-academic performance. Moreover, some of these quotas may overlap, i.e., some students may be eligible for multiple quotas, and in most cases students count in only one of them. The number of seats available for each quota are pre-defined by each school, and schools have some freedom to define these quotas. Hence, our problem could be adapted to help schools define what is the best allocation of seats to quotas in order to improve students' welfare.

14 Strategy-proofness Analysis

A mechanism is strategy-proof if truthful revelation is a dominant strategy for all agents. Roth [1982] and Dubins and Freedman [1981] prove that the student-proposing version of DA is strategy-proof for students. However, it is unclear whether this property holds when we include capacity decisions. For this reason, we now analyze the existence of a strategy-proof mechanism in the context of capacity expansion.

We say that agents (students or schools) have *ex-ante knowledge* about the capacity expansions if they know the expanded capacity of each school before applying. Similarly, we say that agents have *interim knowledge* if they know that the clearinghouse will add at least one seat, but they do not know the exact amount nor the allocation of those extra seats. Finally, we say that agents have *ex-post knowledge* if they know about capacity expansions only after they apply. Our first results, formalized in Theorems 14.1 and 14.2, show that the timing in which the clearinghouse reveals the information about extra capacities plays an important role. Specifically, we prove that the mechanism is strategy-proof for students if they have ex-ante or ex-post knowledge about capacity expansions. However, this is not the case when students have interim knowledge.

Theorem 14.1. *The mechanism is strategy-proof for students if they have ex-ante or ex-post knowledge about capacity expansions. In contrast, the algorithm is not strategy-proof for students if they have interim knowledge about capacity expansions.*

Proof. If students have ex-ante or ex-post knowledge about capacity expansions, then it is direct that the mechanism is strategy-proof for students. The reason is that, in the ex-ante and ex-post cases, capacities are fixed (from the perspective of students), and thus the problem is equivalent to the standard setting, in which we know that student-proposing DA is strategy-proof for students. Then, since our mechanism finds the student-optimal stable-matching, we conclude that our mechanism is strategy-proof for students following the same argument in Roth [1982], Dubins and Freedman [1981].

For the second part, we consider the following counter-example:

$$\begin{array}{ll}
s_1 : c_1 \succ \dots & c_1 : s_1 \succ s_3 \succ \dots \\
s_2 : c_2 \succ \dots & c_2 : s_2 \succ s_3 \succ \dots \\
s_3 : c_1 \succ c_2 \succ c_3 \succ \dots & c_3 : s_3 \succ \dots \\
s'_1 : c'_1 \succ \dots & c'_1 : s'_1 \succ s'_2 \succ \dots \\
s'_2 : c'_1 \succ c'_2 \succ \dots & c'_2 : s'_2 \succ \dots
\end{array}$$

where the "...", represent an arbitrary completion of the preferences. If agents are truthful, the optimal allocation is to assign the extra seat to school c_1 , which will admit student s_3 ; thus, the final matching would be $\{(s_1, c_1), (s_2, c_2), (s_3, c_1), (s'_1, c'_1), (s'_2, c'_2)\}$. If student s'_2 misreports her preferences by reporting

$$s'_2 : c'_1 \succ c_1 \succ c_2 \succ c'_2 \succ c_3,$$

the extra seat would be allocated to c'_1 and s'_2 would get her favorite school. □

Theorem 14.1 shows that the timing of the disclosure of information about extra-capacities is critical to avoid manipulation from the students. Unfortunately, as we show in Theorem 14.2, the mechanism is not strategy-proof for schools in any case.

Theorem 14.2. *The mechanism is not strategy-proof for schools, regardless of the timing by which the clearinghouse reveals the extra seats.*

Proof. If schools have ex-ante or ex-post knowledge about capacity expansions, then it is direct that the mechanism is not strategy-proof for schools. The reason is that, in the ex-ante and ex-post cases, capacities are fixed (from the perspective of schools), and thus the problem is equivalent to the standard setting, in which we know that student-proposing DA is not strategy-proof for schools Sönmez [1997].

If schools have interim knowledge of the extra spots, then consider the following example consisting of three schools c_1, c_2, c_3 all with capacity 2, and three students s_1, s_2, s_3 . Their preferences are as follows:

$$\begin{array}{ll}
s_1 : c_2 \succ c_1 \succ c_4 \succ c_5 & c_1 : \{s_1, s_2\} \succ \{s_1, s_3\} \succ s_1 \succ \{s_3, s_2\} \succ s_2 \succ s_3 \\
s_2 : c_1 \succ c_2 \succ c_3 & c_2 : s_3 \succ s_2 \succ s_1 \\
s_3 : c_1 \succ c_2 \succ c_3 & c_3 : s_3 \succ s_2 \\
s_4 : c_4 & c_4 : s_4 \succ s_1 \\
s_5 : c_5 & c_5 : s_5 \succ s_1
\end{array}$$

In this case, if the schools have interim knowledge about capacity expansions, school c_1 can manipulate the matching by reporting a false capacity of 0. Note that in this case, it is optimal to allocate the extra capacity to school c_1 , obtaining the students-optimal stable matching $\{(s_1, c_1), (s_2, c_3), (s_2, c_2), (s_4, c_4), (s_5, c_5)\}$ with an average rank of 9. Note that as a consequence of the manipulation school c_1 improves her matching, which otherwise would be with students s_2, s_3 . If there is no manipulation, every student would be matched to her most preferred school. Note that the preferences of the agents are responsive. □

$ \mathcal{S} $	$ \mathcal{C} $	Integer Quadratic Program		Aggregated Linearization	
		rows	columns	rows	columns
1000	5	1006	5005	21006	10005
	8	1009	8008	33009	16008
	10	1011	10010	41011	20010
	15	1016	15015	61016	30015
2000	5	2006	10005	42006	20005
	8	2009	16008	66009	32008
	10	2011	20010	82011	40010
	15	2016	30015	122016	60015

Table 2: Size of the formulations.

15 Evaluation of Methods on Random Instances

In this section, we empirically evaluate the performance of our methods to assess which formulations and heuristics work better. To perform this analysis, we assume that students have complete preference lists and that the sum of schools’ capacities equals the number of students. Since the number of variables and constraints increases with $|\mathcal{E}|$, considering complete preference lists increases the dimension of the problem and thus makes it harder to solve, providing a “worst-case scenario” in terms of computing time.

Experimental Setup. We create 30 instances for each combination of the following parameters: $|\mathcal{S}| \in \{1000, 2000\}$, $|\mathcal{C}| \in \{5, 8, 10, 15\}$, $B \in \{1, 2, 5, 10, 20, 30\}$. In particular, for each instance, we generate preference lists and capacities uniformly at random, ensuring that no school has zero capacity. The formulations were coded in Python 3.7.3 and solved through Gurobi 9.1.2, restricted to a single CPU thread and 1 hour of time limit. The scripts were run on an Intel(R) Xeon(R) Gold 6226 CPU on 2.70GHz, running Linux 7.9.⁸ In Table 2, we summarize the initial size of our mathematical programming formulations, namely, the number of constraints (rows) and variables (columns) at the beginning of the computation. Note that the models’ size does not depend on the budget for extra seats and only on the number of students and schools. Also, note that the IQCP formulation is more compact than the aggregated linearization model for the same set of parameters $|\mathcal{S}|$ and $|\mathcal{C}|$.

Experiments. In Table 3, we present the results of our experiments, where we compare the quadratic program (IQP), the aggregated linearization (Agg-Lin), and the two heuristics (Greedy and LPH). The columns with “gap” report the average percentage gap between the best upper bound found by the two exact methods and the solution found by the heuristic; lower values mean a better capacity to find a solution close to the optimum. When running the mathematical programs in Gurobi, we feed the best solution found by the heuristics as a warm start. For IQP and Agg-Lin, we report the following statistics: (i) average percentage root gap between the root value and the best upper bound found by all methods, “rt-g” ; (ii) average number of explored nodes, “nodes” ; (iii) number of instances that reached the time limit of 1 hour, “#g” ; (iv) average last percentage gap found by Gurobi, “%g” ; (v) average gap between the upper bound found and the best upper bound found by the two programs “b-g” ; and (vi) average computing time in seconds, “time” .

⁸All code and synthetic instances will be publicly available upon acceptance

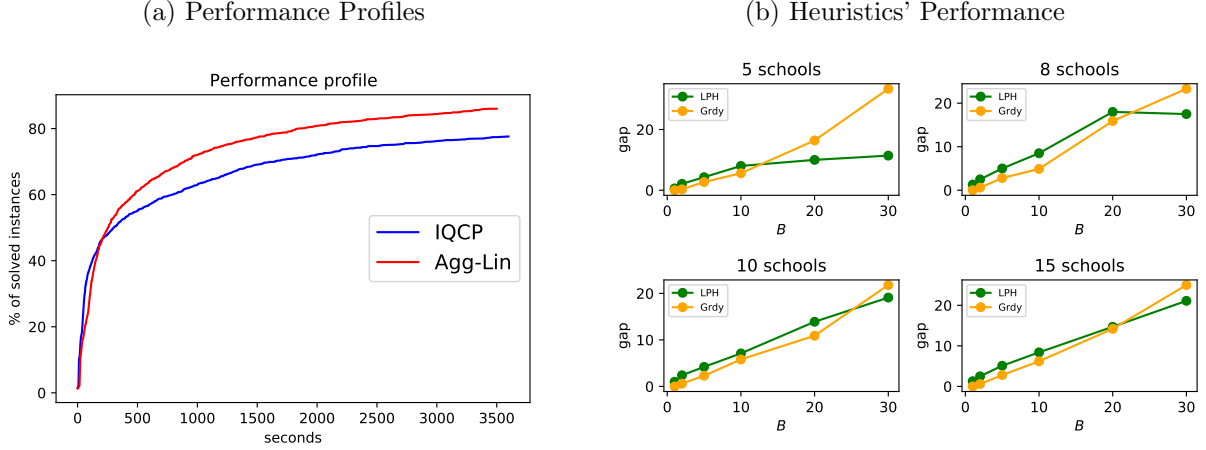


Figure 5: Figure on the left: Performance profiles of the integer quadratic program and the aggregated linearization (experiments on 1440 instances). Figure on the right: The average gap (in %) between the two heuristics and the best upper bound found by the exact methods; the gap is represented as a function of B . The number of students is 2000 and each data-point is the average over 30 instances.

First, we observe that Agg-Lin outperforms IQP starting in $|\mathcal{S}| \geq 1000$, $|\mathcal{C}| = 15$ and $B \geq 1$. Moreover, given that the difference in performance is negligible in other smaller instances, we conclude that Agg-Lin should be preferred. This result is in line with Figure 5a, which shows the fraction of instances solved to optimality (given a mipgap tolerance of $1e-4$) within one hour by each method. Second, by looking at the computing times and gaps for the exact methods, we observe that the main driver of the computing time is the number of schools. Finally, we find that increasing B reduces the computing time of the exact models. In particular, we observe a drastic reduction in the computing time as we move from $B = 1$ to $B = 2$. One possible explanation is that the additional extra seat makes infeasible the branching in the second summation of the stability constraint.

Regarding our heuristics, we find that Greedy performs better than LPH when B is relatively small, but this result reverses as B increases. This result is in line with Figure 5b, where we show the average gap between the heuristics and the best upper bound considering $|\mathcal{S}| = 2000$ and $|\mathcal{C}| \in \{5, 8, 10, 15\}$ (the results are similar with $|\mathcal{S}| = 1000$). Finally, note that LPH has a more consistent performance across different budgets B than Greedy.

S	C	B	LPH	Greedy	Integer Quadratic Program (IQP)						Aggregated linearization (Agg-Lin)					
			gap	gap	rt-g	nodes	#g	%g	b-g	time	rt-g	nodes	#g	%g	b-g	time
1000	5	1	1.4	0.0	44.7	110.6	0	0.0	0.0	21.75	54.0	315.5	0	0.0	0.0	26.23
		2	2.9	0.3	51.5	373.4	0	0.0	0.0	14.23	50.7	107.3	0	0.0	0.0	22.19
		5	6.0	3.6	44.1	126.5	0	0.0	0.0	10.57	43.9	89.1	0	0.0	0.0	21.05
		10	11.8	11.0	41.6	146.8	0	0.0	0.0	11.02	41.6	172.6	0	0.0	0.0	22.93
		20	18.0	28.9	28.3	21.6	0	0.0	0.0	7.95	28.3	102.2	0	0.0	0.0	17.91
		30	25.2	59.1	18.0	15.1	0	0.0	0.0	5.65	18.0	158.7	0	0.0	0.0	12.93
1000	8	1	1.6	0.0	66.2	25068.6	4	2.1	0.0	1488.81	64.7	1929.9	0	0.0	0.0	181.38
		2	3.2	0.8	61.1	819.8	0	0.0	0.0	60.67	60.6	1324.9	0	0.0	0.0	146.83
		5	6.0	3.1	58.6	653.4	0	0.0	0.0	56.41	58.4	1185.1	0	0.0	0.0	111.91
		10	10.6	8.3	55.7	620.9	0	0.0	0.0	60.73	55.6	1204.2	0	0.0	0.0	108.18
		20	18.9	18.3	44.6	347.0	0	0.0	0.0	64.27	44.6	702.1	0	0.0	0.0	63.78
		30	31.7	51.0	27.7	77.6	0	0.0	0.0	22.52	27.7	432.7	0	0.0	0.0	33.15
1000	10	1	2.1	0.0	66.7	27688.8	13	9.5	0.0	2427.38	65.6	8646.9	6	5.2	0.0	1324.37
		2	2.8	0.9	65.8	1015.3	0	0.0	0.0	207.02	64.9	1825.4	0	0.0	0.0	331.98
		5	5.8	3.7	62.7	956.3	0	0.0	0.0	170.84	62.4	1635.7	0	0.0	0.0	227.79
		10	10.0	8.2	58.3	819.7	0	0.0	0.0	308.82	58.1	1395.7	0	0.0	0.0	187.56
		20	20.4	19.9	44.9	464.7	0	0.0	0.0	148.93	44.9	772.6	0	0.0	0.0	112.19
		30	34.7	51.4	32.1	42.2	1	0.1	0.0	44.59	32.1	436.3	0	0.0	0.0	64.01
1000	15	1	2.2	0.0	71.8	11773.6	28	29.5	0.0	3519.41	70.9	6765.4	27	30.5	0.0	3426.71
		2	4.2	0.8	70.2	1699.7	0	0.0	0.0	1427.29	69.2	2352.5	1	0.5	0.0	964.12
		5	7.8	3.6	67.4	2070.9	1	0.1	0.0	1499.12	66.7	3007.2	0	0.0	0.0	822.86
		10	12.7	8.9	65.6	1802.3	1	0.6	0.0	1442.69	65.3	3305.5	0	0.0	0.0	710.30
		20	23.3	20.1	51.4	864.2	0	0.0	0.0	652.50	51.4	1480.8	0	0.0	0.0	349.08
		30	31.4	32.7	44.3	562.8	0	0.0	0.0	450.67	44.3	1134.5	0	0.0	0.0	248.97
2000	5	1	0.6	0.0	59.4	9444.3	0	0.0	0.0	538.29	57.5	1473.8	0	0.0	0.0	108.17
		2	2.1	0.3	54.4	1691.5	0	0.0	0.0	81.46	53.3	356.6	1	0.0	0.0	117.00
		5	4.3	2.7	56.2	1762.2	0	0.0	0.0	59.17	55.9	716.3	0	0.0	0.0	139.85
		10	8.0	5.6	49.2	1370.4	0	0.0	0.0	58.33	49.0	611.6	0	0.0	0.0	137.41
		20	10.0	16.4	41.6	889.6	0	0.0	0.0	53.26	41.6	434.3	0	0.0	0.0	115.28
		30	11.4	33.4	25.3	187.7	0	0.0	0.0	36.10	25.3	230.7	0	0.0	0.0	92.03
2000	8	1	1.3	0.0	67.4	11057.6	17	17.4	0.0	2793.14	66.7	3856.3	7	5.3	0.0	1665.35
		2	2.5	0.6	67.0	17375.8	2	0.5	0.0	1386.71	66.7	2094.3	0	0.0	0.0	861.73
		5	5.0	2.8	62.5	12946.5	4	1.8	0.0	1211.51	62.4	2911.9	2	0.4	0.0	1059.36
		10	8.5	4.9	60.7	13625.2	1	0.3	0.0	944.79	60.7	3414.9	2	0.6	0.0	1099.34
		20	18.0	15.9	48.9	9011.9	2	0.8	0.0	817.34	48.9	2365.6	0	0.0	0.0	666.67
		30	17.5	23.3	44.3	6149.3	1	0.4	0.0	391.54	44.3	2442.0	1	0.4	0.0	410.77
2000	10	1	1.0	0.0	71.0	1355.9	29	38.8	0.0	3528.57	70.7	3569.4	22	25.7	0.0	3140.23
		2	2.4	0.6	68.9	24435.4	21	16.0	0.0	3017.29	68.7	3393.4	4	2.4	0.0	1853.09
		5	4.2	2.3	67.4	18647.5	19	16.3	0.0	3066.72	67.3	3610.9	10	7.1	0.0	2270.44
		10	7.1	5.8	60.7	17816.0	9	8.0	0.1	2248.52	60.8	4481.9	6	3.8	0.0	1904.39
		20	13.9	10.9	56.9	9829.0	13	10.6	0.0	2146.54	56.9	3549.1	6	3.3	0.1	1640.89
		30	19.1	21.8	46.2	5591.9	7	6.0	0.2	1394.91	46.2	3359.9	1	0.5	0.0	830.71
2000	15	1	1.3	0.0	73.3	5793.4	30	54.4	0.0	3600.18	73.2	1424.3	30	48.5	0.0	3601.82
		2	2.5	0.6	72.3	2563.4	30	47.9	0.2	3600.14	72.2	2793.9	26	22.3	0.0	3514.34
		5	5.1	2.8	69.1	1680.6	30	29.9	0.1	3600.63	69.0	3708.1	22	18.4	0.1	3347.38
		10	8.4	6.2	63.9	649.9	26	22.1	0.4	3349.68	64.0	4909.3	13	9.9	0.3	2833.55
		20	14.7	14.2	56.1	535.1	21	15.0	1.8	2888.56	56.1	4524.6	8	4.3	0.0	2140.14
		30	21.1	25.0	48.6	506.1	14	7.0	1.4	2359.37	48.6	5229.9	7	4.9	0.8	1815.62

Table 3: Average results for each triplet $|S|$, $|C|$ and B .