

# Algorithms for Affirmative Action

Nick Arnosti<sup>a</sup>

<sup>a</sup> College of Science and Engineering, University of Minnesota, Minneapolis, Minnesota 55455

Contact: [arnosti@umn.edu](mailto:arnosti@umn.edu),  <https://orcid.org/0000-0002-6685-1428> (NA)

Received: July 6, 2023

Revised: October 2, 2024; January 7, 2025


Accepted: January 25, 2025

Published Online in Articles in Advance:  
April 9, 2025

<https://doi.org/10.1287/ited.2023.0039>

Copyright: © 2025 The Author(s)

**Abstract.** This paper illustrates how fundamental concepts from optimization—such as greedy algorithms, matroids, maximum weight matching, and NP-completeness—arise in domains where policymakers wish to select a set of applicants while ensuring representation for specific groups. Examples of such settings include visa lotteries in the United States, the election for Chile’s constitutional assembly, affordable housing lotteries in New York City, selection for Indian civil service positions, and admission to Indian and Brazilian universities. By providing these examples alongside sample exercises, I aim to offer educators tools to make optimization theory accessible to students at all levels, while highlighting its policy relevance.

 **Open Access Statement:** This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License. You are free to download this work and share with others for any purpose, except commercially, and you must attribute this work as “INFORMS Transactions on Education. Copyright © 2025 The Author(s). <https://doi.org/10.1287/ited.2023.0039>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc/4.0/>.”

**Funding:** This work was supported by the National Science Foundation [Grant 2339912; CAREER: Efficient and Equitable Housing Allocation].

**Supplemental Material:** The online data files are available at <https://doi.org/10.1287/ited.2023.0039>.

**Keywords:** affirmative action • diversity • course materials

## 1. Introduction

How did a procedural modification to the H1B visa lottery made during the Trump administration direct thousands of visas to applicants with advanced degrees? Are low-income renters disproportionately impacted by New York’s “community preference” policy for affordable housing? How did Chile ensure representation of women in its 2021 constitutional assembly? How did affirmative action policies for Brazilian universities and Indian civil service positions end up *disadvantaging* some of the populations they were intended to help?

These questions are all addressed in the Diversity and Affirmative Action unit of my class Engineering the Allocation of Public Resources. They all relate to settings where an organization must select from a set of applicants and wishes to ensure that the selected applicants include adequate representation of particular groups. In the course of addressing these questions, the class touches on topics including greedy algorithms, graph theory, matroid theory, and NP-completeness.

The purpose of this article is to provide a starting point for instructors interested in connecting algorithms and optimization to public policy. I primarily have in mind instructors teaching technical classes who want to connect their material to interesting applications, but the article is also appropriate for law or public policy instructors who want students to get “under the hood” and see how diversity initiatives

are actually implemented. This article provides an overview of several diversity and affirmative action policies in use today, discusses the algorithms used to implement these policies, and offers sample exercises and additional references.

Sections 2 and 3 use U.S. visa allocation as a motivating application to introduce *quotas* and *reserves*, which are used to ensure minimum (or maximum) representation of specific groups. These terms have been used somewhat inconsistently in the literature; I adopt the terminology of Arnosti et al. (2024). Section 4 briefly describes how ideas from the preceding sections apply to affirmative action policies used to allocate affordable housing in New York, elected assembly seats in Chile, civil service positions in India, and university positions in Brazil.

### 1.1. Course Context

My course aims to illustrate connections between optimization and game theory, and societal issues that are more often discussed in public policy classes. Concepts such as *efficiency*, *fairness*, and *incentives* are given precise mathematical definitions. Mathematical concepts such as greedy algorithms, matroids, NP-completeness, and bipartite matching are shown to arise in unexpected contexts.

The first two units of the course study different notions of *individual fairness*: treating people equally

(symmetry) and respecting priority claims. The content in this article is drawn from Unit 3, which studies *group fairness* (or diversity). Mathematically, this means ensuring a desired representation of specific groups of applicants. Designing algorithms to achieve this goal is nontrivial, especially in cases with multiple overlapping groups of interest. My class studies algorithms that have been deployed in practice, and explores cases when these algorithms work well, and cases when they fail to achieve their intended goals.

## 1.2. Student Background

Course enrollment has ranged from 17 to 33 students. Most of these are undergraduate seniors, but the course is listed at the master's level, and typically includes several graduate students (master's and PhD), as well as a few precocious sophomores and juniors. Although most enrolled students are from the Department of Industrial and Systems Engineering, students from statistics, computer science, economics, civil engineering, biomedical engineering, and chemical engineering have all taken the course. To avoid excluding students from other departments, I do not list any formal prerequisites for the course.

Several facts make it possible to teach such a diverse group of students. First, most of the material in the course is not taught in "standard" undergraduate curricula, and thus has not been seen by enrolled graduate students. Indeed, many topics from the course are areas of ongoing research, and typically taught only to PhD students. Despite this, the material is sufficiently elementary to be taught to undergraduates from first principles. Although prior exposure to basic probability and optimization is helpful, students without this background can succeed in the class.

A key barrier to introducing these ideas to undergraduates is that they appear primarily in academic papers, which are filled with abstract mathematical notation and few (if any) examples. While developing the course, I made an effort to introduce concepts through examples, activities, and diagrams before presenting definitions and theorem statements, which are more abstract. I find that abstract definitions are often understandable only after students have "gotten their hands dirty." Mathematical precision is an essential component of the class, but I focus on consequences of theorems, rather than their proofs.

## 1.3. Pacing and Content Selection

My unit on diversity and affirmative action lasts approximately four weeks (eight two-hour classes). However, a subset of the content could be covered as a stand-alone unit in as little as one or two classes. For example, in Spring of 2023, I delivered a guest lecture for Econ 136 at Stanford University (an undergraduate elective), which covered most of the content from Section 2. The content

of Section 3 could similarly be covered in a single class, although I believe that both Sections 2 and 3 are best taught over two classes. The applications of New York's affordable housing, Chile's constitutional assembly, and India's affirmative action policies (discussed in Sections 4.1, 4.2, and 4.3) can each be covered in a single two-hour class period, and instructors could choose which of these applications to cover in an a la carte manner depending on the learning objectives of the course, the time available, and personal interest.

## 1.4. Course Materials

To help instructors incorporate these topics into their courses, I have included a number of figures taken directly from my lecture slides (with minor formatting modifications). In the interest of being concise, most course content is not included in this article, but is available my website, [nickarnosti.com/teaching/](http://nickarnosti.com/teaching/).

## 1.5. Evaluation

I use several approaches to assess whether my lessons have been effective. In class, I introduce activities and exercises, which allow me to circulate through the classroom and observe common mistakes and misconceptions. I also include a daily handout with a few problems on it, which serves both to take attendance and to gauge student understanding. I assign short homework assignments due two times each week (the night before each class). Some of these (called "concept checks") are submitted and automatically graded using Canvas, with results shared with each student immediately upon submission. Students are not allowed to work together on concept checks, so the data from these mini-assessments allows me to see how many students internalized concepts from the previous lesson, and make adjustments to the next class as necessary. In addition, I provide students with a Google Form that they can use at any time throughout the semester to offer anonymous feedback.

Student reactions to the course are consistently positive. Across three years, the average instructor evaluation is 5.7 out of 6, and every year I get at least one comment from a student saying that it has been their favorite college course.

## 2. Quotas

We consider a setting with a set of applicants,  $\mathcal{A}$ , and various categories of applicants,  $C \subseteq \mathcal{A}$ . An *upper* (or *maximum*) quota  $u_C$  on a category  $C$  indicates the maximum number of applicants from this category who can be accepted. A *lower* (or *minimum*) quota  $l_C$  indicates the minimum number of applicants from  $C$  who can be accepted. This section introduces the mathematics of maximum and minimum quotas using the American Diversity Visa Lottery and the Israeli Pre-Military Academy (PMA) match algorithm as motivating examples.

Although finding a set of applicants that complies with all quotas is an NP-complete problem, in special cases, this problem has structure that allows it to be solved in polynomial time. Specifically, this can be done whenever the set of target groups is *nested* (also referred to as *laminar* or *hierarchical*), as explained in more detail below. Within the context of an optimization course, this content can be used to motivate greedy algorithms, computational complexity and NP-hardness, and integer programming.

## 2.1. Upper (Maximum) Quotas

**2.1.1. Diversity Visa Lottery (Example 1).** Upper quotas arise in the Diversity Visa Lottery, which allocates Green Cards for the United States. Up to 55,000 of these visas are awarded annually. At most 7% of this total may go to applicants born in any single country. In addition, there are caps on the total number of successful applicants that have been born in each region (roughly corresponding to continents), which vary by region and year based on recent immigration patterns.<sup>1</sup>

We model this as a set of applicants  $\mathcal{A}$ , who are given random lottery numbers that determine their priority. Each applicant is associated with their country and region of birth, and there are upper quotas on the number of applicants from each country and region that can be accepted. (For now, we take the lower quotas  $l_C$  to be zero.)

**Definition 1.** A *feasible selection* is a set of applicants  $S \subseteq \mathcal{A}$  that satisfies all quotas:  $l_C \leq |S \cap C| \leq u_C$  for each  $C \in \mathcal{C}$ .

These quotas imply that U.S. Citizenship and Immigration Services (USCIS) cannot simply accept the applicants with the best lottery numbers, as illustrated in Figure 1. One natural algorithm—which many students will use intuitively—is a top-down or “greedy” approach, which considers applicants in priority order and accepts them so long as doing so does not violate an upper quota.

### Algorithm 1 (Greedy)

Initialize  $S = \emptyset$ .

Consider applicants in priority order. When considering applicant  $i$ , if the set  $S \cup \{i\}$  does not violate any maximum quotas, set  $S = S \cup \{i\}$ .

Accept applicants in  $S$ , and reject all others.

The greedy algorithm is natural, but is it the “best” algorithm we can use? In many contexts, greedy choices produce suboptimal outcomes. We have not yet introduced an objective function, but one natural goal is to accept as many applicants as possible.

**Definition 2.** A feasible selection *maximizes selected applicants* if no feasible selection accepts more applicants.

When using a greedy algorithm, might we accept fewer applicants than would otherwise be possible? To explore this question, and to give students practice applying Algorithm 1, I provide the exercise in Figure 1. In this example, it is easy to see no more than six applicants can be accepted. In fact, the greedy algorithm always selects six applicants, as the class collectively discovers.

**2.1.2. Israeli Preliminary Academies (Example 2).** Does the greedy algorithm always maximize selected applicants, or is there something special about the example in Figure 1?

We can explore this question using a new application: preliminary academies in Israel. These are essentially gap-year programs that students attend after graduating from high school and before completing their mandatory military service. Each PMA has a distinct curriculum and culture, and limited capacity. After conducting interviews, applicants and PMAs rank each other, and an assignment is found using an adaptation of the deferred acceptance algorithm of Gale and Shapley (1962). The details of this algorithm are beyond the scope of this article; interested readers can learn more by reading Gonczarowski et al. (2019), which is my source for all information about this application.

**Figure 1.** Maximum Quotas Example 1

**Quotas:**

- At **most** 2 applicants from each country.
- At **most** 3 applicants from each region.
- At **most** 6 applicants in total.

**Lottery Order:**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	B	A	D	B	D	B	D	D	A	C	B	A	D	C	B

**Countries of Origin:**

- Afghanistan
- Bhutan
- Cameroon
- Djibouti

1. Which applicants should we select?
2. Compare answers with your neighbor.

---

3. Choose a different lottery order (order numbers 1-16).
4. For your lottery order, does the Greedy algorithm select 6 applicants?

**Notes. Objectives.** Students will practice applying the greedy algorithm. They will see that in this example, for any possible priority order, this algorithm maximizes selected applicants.

**Notation.** Applicant 1 has highest priority, and Applicant 16 the lowest. The letter of an applicant denotes their country of origin, and the color denotes the region where that country is located.

**Commentary.** Most students will likely answer that Applicants 1, 2, 3, 4, 6, and 11 should be accepted. Ask them how they found this answer. Some have likely used the greedy Algorithm 1. You should raise the concern that making greedy decisions could be at odds with other goals, such as maximizing the total number of selected applicants. To explore this, ask each student to write down a permutation of the numbers 1–16, with the first number they write interpreted as the highest-priority applicant, and the last the lowest. Then have them determine the outcome of the greedy algorithm with their priority order. Everyone should end with a total of six accepted applicants.

**Figure 2.** Maximum Quotas Example 2

Category	Max			
Female	2	1. Female, Jerusalem	5. Male, Jerusalem	1. Which applicants are chosen by Greedy? 2. Can you find an ordering where Greedy fails to select 4 applicants?
Male	2	2. Male, Tel Aviv	6. Female, Jerusalem	
Jerusalem	2	3. Female, Tel Aviv	7. Male, Jerusalem	
Tel Aviv	2	4. Male, Tel Aviv		

*Notes. Objectives.* Students will gain practice applying the greedy algorithm, and notice that in some instances, this algorithm may fail to maximize the total number of selected applicants.

*Solution.* When applicants are ranked from 1–7, the greedy algorithm selects Applicants 1, 2, 3, and 5. In general, the greedy algorithm may select either three or four applicants, depending on the ranking of applicants. For example, it selects only three applicants if Applicants 5 and 7 have the highest priority.

For our purposes, the important fact is that in addition to ranking applicants, PMAs also specify maximum and minimum quotas on the number of applicants in certain categories. This information is used by the algorithm to make choices on behalf of each PMA. If a PMA does not specify any minimum quotas, then the algorithm used to make choices on its behalf is equivalent to the greedy Algorithm 1.

Figure 2 presents an example motivated by this context, in which a PMA wishes to select an incoming class that is balanced by gender and geography. In this example, the number of applicants selected by the greedy algorithm *does* depend on the priority order; in some cases, the greedy algorithm selects only three applicants, even though it is possible to select four.

**2.1.3. Nested Categories.** The example in Figure 2 suggests that there was indeed something special about our initial example, which ensured that the greedy algorithm maximized selected applicants. In Figure 1, the categories of applicants are *nested*, meaning that for every two categories, either they are disjoint or one contains the other. This condition is visualized in Figure 3, and appears in different papers under different names. For example, Yokoi (2017) and Gonczarowski et al. (2019) use the word “laminar” instead of “nested,” and Budish et al. (2013) use the term “hierarchy.”

It turns out that if quota categories are nested, then for any priority order, the greedy algorithm maximizes selected applicants. In fact, it gets even better. Not only does Greedy maximize the total *number* of selected

applicants, it also maximizes the *priority* of the selected applicants, in the following sense.

**Definition 3.** Selection  $S$  *priority dominates*  $S'$  if there is a one-to-one function  $\phi : S' \rightarrow S$  such that for each  $s \in S'$ ,  $\phi(s)$  has weakly higher priority than  $s$ .

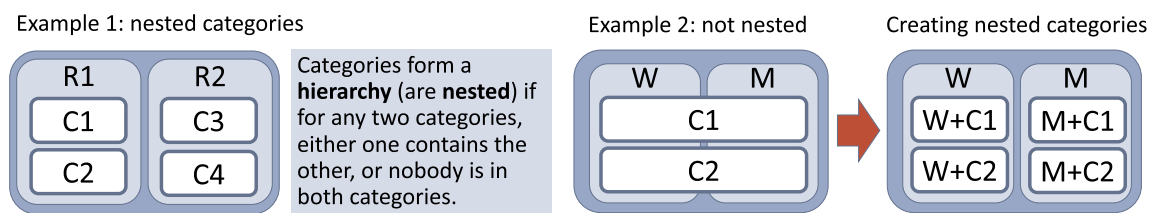
**Theorem 1.** *With maximum quotas, if quota categories are nested, then Algorithm 1 always finds a selection that priority dominates all other feasible selections.*

This result is a special case of Theorem 2 below. Because  $\phi$  is required to be one-to-one, if  $S$  priority dominates  $S'$ , then  $|S| \geq |S'|$ . Therefore, Theorem 1 implies that when categories are nested, the greedy algorithm finds a selection that maximizes selected applicants.

Many students find the definition of priority domination unintuitive. I also offer two alternative (but equivalent) definitions, which may be more intuitive for some students. The first is that selection  $S$  priority dominates  $S'$  if  $S$  contains at least as many of the  $k$  highest-priority applicants as  $S'$  does, for every  $k \in \{1, 2, \dots, |\mathcal{A}|\}$ . The second is that  $S$  priority dominates  $S'$  if  $|S| \geq |S'|$ , and the  $k$ th highest-priority applicant in  $S$  has at least as high priority as the  $k$ th highest-priority applicant in  $S'$ , for every  $k \in \{1, 2, \dots, |S'|\}$ .

Regardless of which definition is used, it is helpful to give students an opportunity to practice with this concept. Figure 4 shows one exercise that I use. An important observation is that priority domination only induces a partial order: for some pairs of selections, neither dominates the other. The interpretation I give

**Figure 3.** Illustration of Nested Categories



*Note. Objectives.* Provide students with a definition that formalizes the key difference between Examples 1 and 2 (in the former, categories are nested, and in the latter they are not), provide a visualization of nested and nonnested categories, and illustrate how nested categories can be constructed by considering intersections of the original categories.

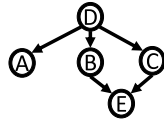
Figure 4. Priority Domination

### Priority Domination Practice

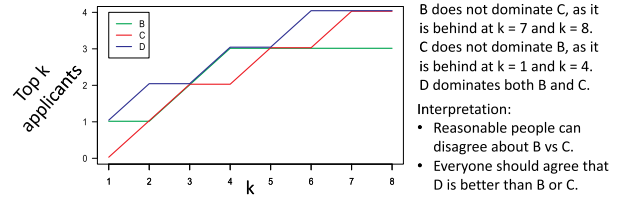
Applicants are ranked  $1 > 2 > 3 > 4 > 5 > 6 > 7 > 8$

For each pair of selections, does one priority dominate the other?

- |                           |        |        |
|---------------------------|--------|--------|
| Selection A: {1, 4, 5, 8} | A vs B | B vs D |
| Selection B: {1, 3, 4}    | A vs C | B vs E |
| Selection C: {2, 3, 5, 7} | A vs D | C vs D |
| Selection D: {1, 2, 4, 6} | A vs E | C vs E |
| Selection E: {2, 3, 6}    | B vs C | D vs E |



### Priority Domination Visualization



Notes. Objectives. Give students practice with the concept of priority domination.

Solution. Given by green circles and/or the directed graph (note that priority domination is transitive).

is that in these cases, reasonable people could disagree about which selection is preferable.

## 2.2. Lower (Minimum) Quotas

In practice, PMAs specify both maximum and *minimum* quotas. The presence of minimum quotas complicates matters significantly. With only upper quotas, a feasible selection trivially exists, as it is always feasible to accept nobody. With the addition of lower quotas, this is not the case. Even if a feasible selection exists, finding it is nontrivial. In fact, this problem turns out to be NP-complete.<sup>2</sup>

2.2.1. Heuristic Algorithms. In practice, a variety of heuristics are used to try to solve this problem. These can broadly be divided into two categories:

1. Ignore minimum quotas, and apply Algorithm 1. If the resulting selection satisfies all minimum quotas, use this selection. Otherwise, make ad hoc modifications to try to achieve feasibility.
2. Try to satisfy minimum quotas before accepting applicants who do not contribute to these quotas.

Within each of these high-level approaches, there are many variations, which differ in the way they modify the output of Algorithm 1 (in the first case) or the way they try to satisfy minimum quotas (in the second). For example, Israeli PMAs follow the second approach. They conduct two passes in priority order:

the first accepts applicants who contribute to one or more unmet minimum quotas, and the second accepts any applicant who does not cause a maximum quota violation. Other variants of this approach consider quota categories in a specific order. Two such algorithms are described in Figure 5. In general, the order in which categories are considered is consequential.

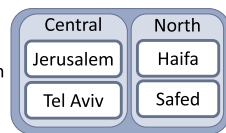
There are several challenges when using these heuristics. One is that when choosing among heuristics, it can be difficult to anticipate the way in which their outcomes will differ. Furthermore, it may be difficult to justify why a particular heuristic was chosen. An additional challenge is that these heuristics may fail to find feasible selections on some input.

If quota categories are nested, then there are efficient algorithms to determine whether a feasible selection exists. Furthermore, if it does, these algorithms can find a feasible selection that priority dominates all others. One example algorithm fills minimum quotas for more specific categories before those for less specific ones (as algorithm CR does in Figure 5). An alternative algorithm that produces the same outcome was proposed by Gonczarowski et al. (2019). This algorithm sequentially selects the highest-priority applicant from the set of applicants who contribute to the largest number of unmet minimum quotas and do not cause violation of a maximum quota.

Figure 5. Minimum Quotas

### Minimum Quotas

At **most** 7 applicants  
 At **least** 3 applicants from each region  
 At **least** 1 applicant from each city



Priority Order:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
J	T	J	H	T	H	T	H	H	J	S	T	J	H	S	T

- |   |   |
|---|---|
| <b>Algorithm CR.</b> From top of priority list, | <b>Algorithm RC.</b> From top of priority list, |
| 1. Fill city minimum quotas                     | 1. Fill regional minimum quotas                 |
| 2. Fill regional minimum quotas                 | 2. Fill city minimum quotas                     |
| 3. Accept until reaching maximum                | 3. Accept until reaching maximum                |

What is outcome of each algorithm? Which outcome seems better?  
 Identify a priority order such that Algorithm RC fails to find a feasible selection.

Notes. Objectives. Introduce two algorithms for minimum quotas. When quota categories are nested, starting with the most specific quotas (as algorithm CR does) finds a feasible selection whenever one exists, and selects applicants with as high priority as possible.

Notation. Applicant 1 has highest priority, and Applicant 16 the lowest. Letters denote applicants' cities of origin, and colors denote the regions where those cities are located.

Solution. Algorithm CR selects Applicants 1, 2, 3, 4, 5, 6, and 11. Algorithm RC selects Applicants 1, 2, 3, 4, 6, 8, and 11, which is priority dominated. Algorithm RC fails to find a feasible selection if the top three northern applicants are from the same city and the top three central applicants are from the same city.

**2.2.2. Intersectional Identities.** To circumvent the NP-hardness result, we might choose to create nested quota categories. Figure 3 illustrates how, starting from categories that are not nested, it is possible to consider intersectional categories that are nested. Quotas can be chosen for these intersectional categories to ensure that the original quotas are satisfied whenever the intersectional quotas are. However, creating this nested structure may overconstrain the problem, creating infeasible outcomes and/or requiring the selection of low-priority applicants. For example, a constraint that a panel of four people must contain two women, two men, two people over 40, and two under 40 could be enforced by requiring one person from each of the four (gender, age) intersectional identities. However, this requirement may be infeasible even if the original constraints are not.

**2.2.3. Top-Down Processing.** In cases where imposing a nested structure is not a desirable solution, an alternative is to use an algorithm with a running time that is exponential in the worst case, but guarantees output with certain desirable properties.

**Algorithm 2** (Top-Down)

Initialize  $S = \emptyset$ .

Consider applicants in priority order. When considering applicant  $i$ , if there exists a feasible selection that contains  $S \cup \{i\}$ , set  $S = S \cup \{i\}$ .

Accept applicants in  $S$ , and reject all others.

When there are no minimum quotas, Algorithm 2 reduces to the greedy Algorithm 1.

The top-down algorithm offers several advantages over its heuristic counterparts. By definition, it finds a feasible selection of applicants whenever one exists. When categories are nested, Yokoi (2017) establishes the following result.

**Theorem 2** (Yokoi 2017, Lemma 3 and Example 1). *With maximum and minimum quotas, if quota categories are nested and there is at least one feasible selection, then Algorithm 2 always finds a selection that priority dominates all other feasible selections.*

When categories are not nested, there might not be a feasible selection that priority dominates all others. However, the selection found by Algorithm 2 is never priority dominated by another feasible selection. Arnosti et al. (2024) argue that this algorithm offers a particularly simple explanation of why unsuccessful applicants were not selected: choosing them would mean either violating some quota or rejecting a higher-priority applicant who is currently accepted.

Note that, in general, the top-down algorithm may not maximize selected applicants. For example, suppose that there are many categories, the highest-priority applicant belongs to all categories, every applicant

belongs to at least one category, and there is a maximum quota of one on each category. The top-down algorithm selects only the first applicant, even though it is possible to select many more.

Implementing Algorithm 2 requires determining whether there is a feasible selection containing  $S \cup \{i\}$ , which may be computationally challenging. For most practical problems, this can be done using optimization-based software. This can be a good opportunity to introduce integer programming. If the set of quota categories is  $\mathcal{C}$ , each category  $C \in \mathcal{C}$  is associated with upper quota  $u_C$  and lower quota  $l_C$ , and we wish to add individual  $i$  to set  $S$ , we can create decision variables  $\{y_j\}_{j \in \mathcal{A}}$  and check whether the following system is feasible:

$$l_C \leq \sum_{j \in \mathcal{C}} y_j \leq u_C \text{ for } C \in \mathcal{C}, \quad (1)$$

$$y_j \in \{0, 1\} \text{ for } j \in \mathcal{A}, \quad (2)$$

$$y_j = 1 \text{ for } j \in S \cup \{i\}. \quad (3)$$

Rather than calling a feasibility checker  $n$  times (once for each applicant  $i \in \mathcal{A}$ ), the final selection of the top-down algorithm can be found in a single call by introducing an objective: if  $\pi(j)$  gives the priority of individual  $j$  (with  $\pi(j) = 1$  highest priority and  $\pi(j) = |\mathcal{A}|$  lowest), then we wish to maximize  $\sum_j 2^{|\mathcal{A}| - \pi(j)} y_j$  subject to (1) and (2). Having the objective double each time the priority of the individual increases by one position ensures that it is more valuable to select one high-priority individual than all of the individuals below them, and therefore that the solution to this optimization problem will be the outcome of the top-down Algorithm 2. Figure 6 summarizes which goals are achievable for nested and general quotas (assuming  $P \neq NP$ ).

**2.2.4. References.** Additional information about the Diversity Immigrant Visa Program can be found at U.S. Department of State (2023). Details about the matching process for Israeli PMAs are presented in

**Figure 6.** Summary: Maximum and Minimum Quotas

	Nested	In General (Poly Time)	In General (Exp Time)
Finds Feasible Selection	✓	✗	✓
Maximizes Selected Applicants	✓	✗	✓
Priority Dominates Other Feasible Selections	✓	✗	✗

*Notes. Objectives.* Summarize the conclusions from study of quotas. If categories are nested, then the problem is “easy.” Otherwise, there may be a trade-off between algorithms that run quickly and sometimes fail, and those that may run slowly but always find a feasible selection.

Gonczarowski et al. (2019). Mathematical results on quotas can be found in Yokoi (2017) and Arnosti et al. (2024). Sonmez and Yenmez (2019) study what they call “one-to-all” reserves, which are equivalent to lower quotas. They focus on algorithms for the case with just two categories. Lecture slides on quotas can be found on my website, [nickarnosti.com](http://nickarnosti.com).

### 3. Reserves

When using quotas, applicants count toward quotas of all groups to which they belong. For example, if there is a minimum quota on the number of women selected and on the number of minorities selected, then a female minority counts toward both quotas.

This may be undesirable for two reasons. Mathematically, it can result in computationally intractable problems, as discussed in Section 2. Practically, it can result in situations where a small number of applicants belonging to multiple targeted groups are selected, in order to enable the selection of others belonging to none. For example, consider the process of selecting a committee of three people, which must contain at least one woman and one minority. The selection of a woman minority candidate fulfills both requirements, leaving two spots available for any other candidates. Although this may be acceptable in some contexts, in others it may be seen as circumventing the intent of the policy.

An alternative approach is to say that each applicant can count toward at most *one* of the categories for which they are eligible: the hypothetical female minority candidate can be counted as either a female or a minority, but not both. This is called a *reserve* system. When using a reserve system, practitioners must decide how to “count” candidates who are eligible for multiple categories.

I motivate this topic using H1B visa allocation, as described in Section 3.1. I spend one two-hour class on this application, during which I introduce several algorithms. A key conceptual takeaway is that a policy that reserves visas for a particular group is not fully specified: this policy can be implemented using different algorithms, whose outcomes differ substantially. The

next class presents the more general model of overlapping reserve categories from Section 3.2. This model can be used to motivate and explore concepts from optimization and graph theory including the difference between maximal and maximum matchings, algorithms for finding maximum matchings, and matroid theory.

#### 3.1. H1B Visa Allocation

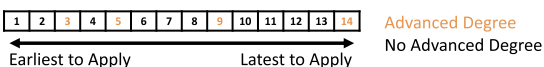
H1B visas are temporary employment-based visas, which allow employers to hire foreign laborers into “specialized occupations” that require “the application of a body of highly specialized knowledge and the attainment of at least a bachelor’s degree or its equivalent.”<sup>3</sup> Each year, 65,000 visas are issued to eligible foreign workers, and an additional 20,000 are set aside for applicants who have earned advanced degrees from an institution in the United States (hereafter, “advanced degree holders”). Initially, priority was first come, first served. This invites the question, if the first applicant holds an advanced degree, do they get a “standard” visa or a “reserved” one?

The USCIS has used both procedures. In 2005 (the first year after the 20,000 reserved visas were created), early applicants all received standard H1B visas, regardless of whether they had an advanced degree or not. This left 20,000 additional visas for applicants outside of the first 65,000 who had advanced degrees. This system is referred to by Pathak et al. (2025) as an “over-and-above” implementation. In several subsequent years, the USCIS used an “exemptions-first” policy: anyone eligible for a reserved position was assigned to one of these positions until none remained. As Figure 7 illustrates, if we fix the application order, these procedures award visas to different applicants, and the differences between the procedures can be substantial. The following result states that reserve-eligible applicants always prefer the over-and-above implementation, and applicants who are not reserve-eligible prefer exemptions first.

**Theorem 3** (Pathak et al. 2025, Theorem 1). *Any reserve-eligible applicant who is selected by the exemptions-first algorithm is also selected by the over-and-above algorithm.*

Figure 7. Reserve Example

We have 14 applicants for H1B visas. We can award only 6 visas, but an additional 2 spots are available to applicants with an advanced degree from a US institution. Applicants are numbered by arrival order.



In 2005, which applicants would receive visas?  
In 2006–2008, which applicants would receive visas?  
Which method selects higher priority applicants?  
Which method selects more applicants with advanced degrees?

**Notes.** **Objectives.** Students will practice the exemptions-first and over-and-above processing rules. They will see the differences between the outcomes of these rules: Exemptions first always selects (weakly) higher-priority applicants, and (weakly) fewer reserve-eligible applicants.

**Notation.** There are 14 applicants, arriving in the order 1–14. There are six visas that all applicants are eligible for and two that only applicants with advanced degrees (in orange) are eligible for.

**Solution.** In 2005, USCIS used the over-and-above algorithm, which selects Applicants 1–6, 9, and 14. From 2006 to 2008, USCIS used the exemptions-first algorithm, which selects Applicants 1–8.

Any reserve-ineligible applicant who is selected by the over-and-above algorithm is also selected by the exemptions-first algorithm.

There are other algorithms that could be used to allocate visas. In light of Theorem 3, a natural question is whether we could find a reasonable interpretation of the law that selects even more reserve-eligible applicants than the over-and-above algorithm (or fewer than exemptions first). Pathak et al. (2025) argue that we cannot. They show that among algorithms that “comply with the statute,” over-and-above and exemptions-first algorithms are extremal: the former is best for reserve-eligible applicants, and the latter is best for those who are not reserve-eligible.

**3.1.1. Learning Objectives.** By the end of class, students should be able to do the following:

- Describe four algorithms that have implemented H1B law since 2005. (These algorithms are over-and-above, exemptions first, and two lottery procedures that have been used since 2009 but are not discussed in this article; see Pathak et al. 2025 for details.)
- Identify which of these procedures select more applicants with advanced degrees.
- Calculate the number of successful degree holders for examples with (i) a given priority order and (ii) a random priority order (See Figure 8).

Beyond these specifics, the important qualitative message from class is that although the law governing H1B visas has not changed since 2005, this law did not precisely specify certain procedural details. Different implementations of this law can have major, but often overlooked, consequences for the final allocation. Administrations might take advantage of this ambiguity by choosing implementations that are most aligned with their policy objectives. An example of this is a procedural change made in 2019, which had the (intended) effect of increasing the number of

successful advanced degree holders by several thousand, with a corresponding decrease for those without advanced degrees.

**3.2. Multiple Reserve Categories**

In many applications, including those discussed in Section 4, seats are reserved for people from several overlapping categories, such as women, ethnic minorities, people with disabilities, people with low incomes, and so on. We have already seen that even with only one targeted population, multiple algorithms could be used to implement a given reservation policy. With multiple categories, this complexity increases.

Several papers have introduced general models of reserves to handle these cases. I will adopt the terminology and definitions of Arnosti et al. (2024). There is a set of applicants  $\mathcal{A}$  and a set of positions  $\mathcal{P}$ . For each position  $p \in \mathcal{P}$ , only certain applicants are eligible. These eligibility requirements can be summarized by a bipartite graph  $G = (\mathcal{A}, \mathcal{P}, \mathcal{E})$ , with vertices representing applicants and positions, and an edge from applicant  $a$  to position  $p$  if and only if  $a$  is eligible for  $p$ . The “positions” are for accounting purposes only: applicants care only about whether they were assigned to a position or not (rather than which position they were assigned).

**Definition 4.** A matching  $M$  is a subset of the edges of  $G$  such that each individual and position is incident to at most one selected edge. A selection of individuals  $S \subseteq \mathcal{A}$  is feasible if there is a matching of  $G$  that matches every applicant in  $S$ .

The H1B setting discussed above is a special case of this model, where there are two types of positions: standard and reserved. All applicants are eligible for the former, but only applicants with advanced degrees from U.S. institutions are eligible for the latter. The definition from Pathak et al. (2025) of what it means to “comply with the statute” includes three properties.

Figure 8. Reserve Practice

**Group Work:**  
 We have 65,000 unreserved visas, and 20,000 reserved visas.  
 How many advanced degree holders receive visas in each scenario?

		Earliest to Apply			Latest to Apply	
		First 65k	Next 20k	Rest of List	Policy	# Degree Selected
a)	Degree	24,600	8,700	25,000	Exemptions First	
	Nondegree	40,400	11,300	36,300	Over and Above	
b)	Degree	14,200	3,100	25,000	Exemptions First	
	Nondegree	50,800	16,900	36,300	Over and Above	

**Notes. Objectives.** Students will get practice with analyzing over-and-above and exemptions-first policies, even in settings where the complete priority list is not provided. They will see the large difference between these methods. They will note that the exemptions-first policy may not give an advantage to reserve-eligible applicants if they are already adequately represented.

**Solution.** Scenario a, when using exemptions first, all of the first 85,000 applicants receive visas, resulting in 33,300 degree holders being selected. When using over-and-above, 44,600 degree holders are selected (those in the first 65,000 and an additional 20,000). In Scenario b, when using exemptions first, all of the first 65,000 applicants receive visas, as well as an additional 5,800 degree holders, for a total of 20,000 degree holders. When using over-and-above, 34,200 degree holders are selected (those in the first 65,000 and an additional 20,000).

Two of these (“nonwastefulness” and “accommodating the reservation policy”) are jointly equivalent to requiring the selection of a maximal matching of  $G$ .

Requiring maximality in  $G$  is a fairly weak guarantee. In general, many algorithms may achieve this, even while producing selections that are priority dominated (Definition 3). For example, consider a setting with two visas (one standard and one reserved), and two applicants, one of whom is reserve eligible. Matching the reserve-eligible candidate to the standard visa and rejecting the second candidate is maximal in  $G$ , but only selects one applicant, even though it is possible to select both.

A natural goal is to find a feasible selection (Definition 4) that maximizes selected applicants (Definition 2). This requires finding a maximum matching in the graph  $G$ . Is there a computationally efficient way to do this? The following result says that not only can we find a maximum matching in polynomial time, but that there is no trade-off between maximizing selected applicants and selecting high-priority applicants: there is a single selection that optimizes both objectives simultaneously.

**Theorem 4** (Arnosti et al. 2024, Propositions 2 and 3). *For any compatibility graph  $G$  and any priority order  $\pi$ , there exists a feasible selection that priority dominates all other feasible selections. This selection can be found in polynomial time using (for example) the Hungarian algorithm.*

Closely related results appear in other papers (see, e.g., Sonmez and Yenmez 2022, proposition 2). In the special case of H1B visa allocation discussed in Section 3.1, the priority-dominant feasible selection can be found by applying the exemptions-first algorithm. In general, more sophisticated algorithms are required. For those teaching a course on optimization, this application motivates the study of maximum matching algorithms for bipartite graphs, such as the Hungarian algorithm, the Edmonds–Karp algorithm, and others. These algorithms can be used to find the priority-dominant selection by adding weights to the edges. One way of doing this is to give edges that involve higher-priority individuals a greater weight, and then find a maximum weight matching. (Interestingly, because of the matroid structure discussed below, the exact choice of weighting is inconsequential.)

It is instructive to compare the statements of Theorems 2 and 4. The conclusions are similar: there is a feasible selection that priority dominates all others. However, for reserves, this conclusion always holds. For quotas, this holds only if quota categories are nested. One might ask, is there an underlying structure that explains these results? The answer is yes.

This is an opportunity to introduce the concept of a matroid (Wikipedia 2025). When feasible selections are defined as in Definition 4, the set of applicants and

feasible selections form a matroid. This is also true for upper quotas when categories are nested. When there are lower quotas, the collection of feasible selections is not downward closed and is therefore not a matroid, but it is a generalized matroid, as defined by Yokoi (2017). Conversely, it is easy to see by example that when categories are not nested, the feasible selections do not form a matroid. Although I do not explore matroid theory in depth in my course, a course more focused on optimization could cover this topic more extensively.

**3.2.1. References.** Pathak et al. (2025) provide more information about the history of H1B visa allocation. In 2009, overwhelming demand forced USCIS to move to a lottery-based system. The government chose to conduct selection using sequential lotteries (one for all applicants, and one for advanced degree holders). Arnosti (2019) analyzes the effect of a 2019 change which reversed the order of these lotteries.

The difference between exemptions-first (or minimum guarantee) and over-and-above processing arises in other applications. For an example from school assignment in Boston, see Dur et al. (2018). In that case, parents and administrators appear to have not understood the importance of this distinction for at least a decade. This confusion has also been replicated in laboratory settings: Pathak et al. (2023) present evidence that many people understand the importance of varying the number of reserved visas, but do not understand the importance of which algorithm is chosen to implement these reserves.

In some applications, reserves are not absolute requirements: if a reserved position would go vacant because there is no eligible applicant to fill it, then it may be permissible to fill this position with another applicant. There are various ways to define these “soft reserves” precisely; see Sonmez and Yenmez (2019, 2022), Aygün and Turhan (2022), and Arnosti et al. (2024) for details.

## 4. Applications: Affordable Housing, Elections, Civil Service

Quotas and reserves arise in many other contexts. My class discusses several of these applications in detail. This section gives a brief overview of these topics and provides additional references for those interested in learning more.

### 4.1. Affordable Housing in New York

**4.1.1. Overview.** Until recently, New York’s 421a law provided tax credits to developers who agreed to provide a certain fraction of the units in their building at affordable rates. These units are listed on New York City’s (NYC’s) Housing Connect website. After the application window closes, the developer is in charge

of screening applicants to determine their eligibility and, if eligible, offering them a unit. This process is complicated by the following policies:

- **Processing order.** The city places applicants in a random order, which plays the role of priority: developers must follow this order when allocating apartments.
- **Unit-specific eligibility.** Each apartment can only go to households of an appropriate size and income, as determined by the layout (studio, one bedroom, two bedrooms, etc.) and affordability target (specified as a percentage of area median income).
- **Preferences (quotas).** At least 50% of apartments must go to community district residents. At least 7% must go to applicants with disabilities, and at least 5% to city employees. An ongoing lawsuit alleges that the community preference policy violates the Fair Housing Act.

Figure 9 illustrates these policies for a single building.

These policies are implemented using the following algorithm.

**Algorithm 3** (Developer Screening)

Consider applicants in four stages.

1. In lottery order, consider applicants with disabilities until 7% of apartments are occupied.
2. In lottery order, consider community residents until these applicants occupy 50% of apartments.
3. In lottery order, consider city employees until these applicants occupy 5% of apartments.
4. In lottery order, consider remaining applicants until no apartments remain.

**Notes:**

- When a household is considered, it may select any available apartment for which it is eligible.
- Households may count toward multiple categories. For example, if city employees occupy at least 5% of affordable apartments after Step 2, then none are selected in Step 3.

**4.1.2. Teaching This Material.** When I teach this class, I note that these policies essentially combine quotas and reserves. The city’s “preferences” function as quotas: if an applicant is selected because of community preference, and this applicant also happens to be a city employee, then they count toward the city employee

minimum as well. The eligibility requirements function as reserves: each unit can be thought of as a position, and only a subset of applicants are eligible for each unit (based on household size and income). However, unlike in the model presented in Section 3, applicants care about which “position” (unit) they are assigned.

The main conclusion that I highlight when teaching this material is that the developer screening algorithm causes the community preference policy to affect low-income applicants much more than middle-income applicants. In particular, low-income applicants who are not community residents have very low chances of success. The reason is that most applicants are low income, and thus most low-income units are claimed in Step 2 of the developer screening algorithm. This leaves few low-income units available for applicants who do not qualify for city preferences.

This point can be illustrated using the simple example in Figure 10, in which there are no disability or city employee preferences, all units have the same layout, and units are targeted to two disjoint income ranges. Analyzing success probabilities without community preference is straightforward. To analyze success probabilities with community preference, I do a demonstration in which 15 students receive ID numbers from 1–15. I generate a random lottery draw and walk through the selection procedure. After doing this several times, students see that community residents often claim both low-income units. I then run a large number of simulations to generate the selection probabilities displayed in the table in Figure 10.

This example also reinforces one key message from the H1B visa allocation class: policies that specify the number of units reserved for different groups can be implemented using many different algorithms. The choice of algorithm is important but often overlooked. In this example, using the top-down algorithm instead of the developer screening algorithm (while keeping the number of units reserved for each group the same) significantly changes success probabilities.

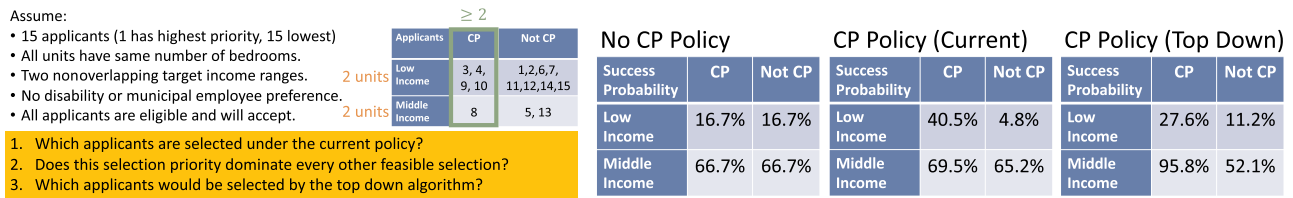
If you do not teach this content as a stand-alone class, it can still provide inspiration for homework or exam problems. Here are some potential questions:

**Figure 9.** NYC Affordable Housing Policies



*Notes.* Of the 268 units available at One East Harlem Residences, some must go to applicants from the community, with disabilities, or who are city employees (left). In addition, units are classified into types based on their layout and income target (middle). Complex rules determine the range of household sizes and incomes that are eligible for each unit type (right).

**Figure 10.** Analyzing the Consequences of Community Preference



**Notes. Objectives.** Students will see by example that the developer screening algorithm may produce a selection that is priority dominated, that the community preference (CP) policy disproportionately impacts low-income families, and that other algorithms would produce very different outcomes.

**Solution.** The developer screening algorithm selects applicants {3, 4, 5, 8}. This is priority dominated by the selection {1, 3, 5, 8}, which is the outcome of the top-down algorithm. Because priority is determined randomly, we can calculate the probability that an applicant with certain characteristics will be selected (see tables on right). For both income groups, the community preference policy gives an advantage to community applicants, as expected. However, the size of this advantage depends heavily on an applicant’s income. The developer screening algorithm starts by taking community residents. Because most applicants are low income, this leaves few units available for low-income applicants from outside the community. As a result, the current implementation of the community preference policy affects middle-income applicants much less than low-income ones. The top-down algorithm would result in very different selection probabilities.

- Based on the description of the NYC developer screening algorithm, do city preferences operate as quotas or reserves? Explain your answer.
- Based on the description of the NYC developer screening algorithm, do income and household size requirements operate as quotas or reserves? Explain your answer.
- Does the outcome of the NYC developer screening algorithm always priority dominate every other feasible selection? Explain why it does, or give a counterexample.
- Is the outcome of the NYC developer screening algorithm ever priority dominated by another feasible selection? Explain why not, or give an example demonstrating that it can be.

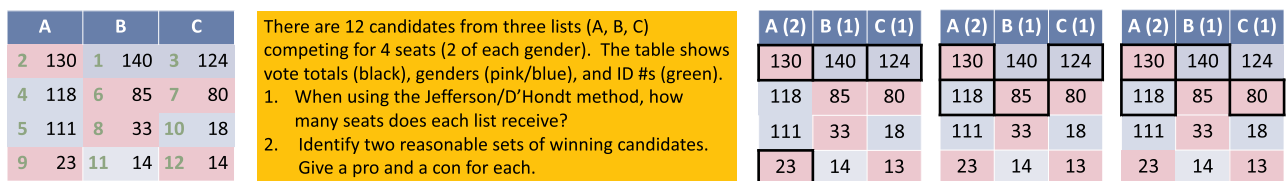
**4.1.3. References.** Arnosti (2022) presents a video analysis of the effects of the community preference policy. Beveridge (2019) conducts an analysis of the community preference policy on behalf of the plaintiffs, and Siskin (2019) provides a rebuttal on behalf of the city. Lecture slides are available on my website, [nickarnosti.com/teaching/](http://nickarnosti.com/teaching/).

## 4.2. Chile’s Constitutional Assembly

**4.2.1. Overview.** In 2021, Chile elected 155 delegates to draft a new constitution. Each district elected a pre-determined number of delegates based on its population, and a separate election was held for 17 of the 155 seats, which were reserved for members of indigenous groups. Outcomes were determined separately for each district. The delegates from each district had to be politically representative of the votes cast in that district (informally, a political coalition that received 1/3 of the votes in a district should receive approximately 1/3 of the delegates). The delegates from each district also had to include an equal number of men and women (these numbers could differ by one if the total number of delegates from the district was odd).

Given all of these constraints, it is not clear how to determine winners from vote totals. To illustrate, consider the stylized example in Figure 11, which has 12 candidates from three political coalitions (“lists”) A, B, and C. The first step in the Chilean algorithm is to determine the number of seats won by each list. This is done by summing the votes received by candidates

**Figure 11.** Chilean Constitutional Assembly Practice



**Notes. Objectives.** Give students practice with the Jefferson D’Hondt apportionment method and get them to think about different algorithms for resolving overlapping gender/list constraints.

**Solution.** The Chilean algorithm starts by ignoring gender constraints and hoping they are satisfied. In this example, taking the top two candidates from list A and the top candidates from lists B and C would result in three male delegates and only one female. There are three natural ways to modify this selection to ensure gender balance. The Chilean algorithm replaces the candidate from the overrepresented gender who has the fewest votes (resulting in the selection of Candidates 1, 2, 3, and 9). Alternatives include selecting the underrepresented gender candidate with the most votes (second proposal), or the selection that maximizes the total votes of elected delegates (third proposal).

from each list, and then applying the Jefferson D'Hondt method. This method works as follows. Let  $v_i$  be the total number of votes earned by list  $i$ , and let  $s$  be the total number of seats. Then list  $i$  wins  $\lfloor v_i/k \rfloor$  seats, where  $k$  is chosen to satisfy the equation  $\sum_i \lfloor v_i/k \rfloor = s$ .<sup>4</sup>

In the example from Figure 11, list  $A$  earns two seats, and lists  $B$  and  $C$  earn one seat each. However, selecting the top two candidates from  $A$  and the top candidates from  $B$  and  $C$  would result in the election of three men and one woman, violating the gender parity constraint. Clearly, one of the men must be replaced by a woman from the same list, but which one?

When presenting this example in class, I usually ask students who they think should be selected, triggering lively debate. The Chilean algorithm identifies the tentatively winning candidate from the majority gender who has the fewest votes and replaces this candidate with a candidate from the same list of the opposite gender.<sup>5</sup>

**4.2.2. Teaching This Material.** After applying the Jefferson D'Hondt method, we have a selection problem with quotas on the number of candidates from each list (with lower and upper quotas both equal to the number of seats won by the list) and gender (with lower and upper quotas of  $\lfloor s/2 \rfloor$  and  $\lceil s/2 \rceil$ , respectively, where  $s$  is the total number of seats). Candidates can naturally be ranked by the number of votes received.

This application offers a nice illustration of nested constraints and priority dominance. Without the gender constraints, categories would be nested, implying the existence of a feasible selection that priority dominates all others. After introducing gender quotas, categories are no longer nested, so it may be that no feasible selection priority dominates all others. However, the outcome of the simplified Chilean algorithm described in this paper is never priority dominated by another feasible selection.

The salience of priority dominance is illustrated by the results in Chilean District 3. Absent the gender constraint, the district would have elected three women and one man. The three women belong to three lists:  $XP$ ,  $ZN$ , and  $ZZ$ . Before defining the Chilean algorithm, I ask students which of the three women they think *should* be replaced. Students debate whether to swap the woman from list  $ZN$  or list  $XP$ : nobody advocates for swapping the woman from list  $ZZ$ . It turns out that swapping  $ZZ$  produces a priority dominated selection. Swapping  $ZN$  and  $XP$  both produce selections that are not priority dominated.

The Chilean election can also be used to launch an exploration of political apportionment. This topic arises, for example, when dividing 435 house seats among 50 states based on their populations. There is a rich literature on different apportionment methods. I often give

census data to my students and ask how many house seats each state would receive if we used the Jefferson D'Hondt method. The choice of methods is consequential for several states. For example, under the Jefferson D'Hondt method, Minnesota would receive only seven congressional seats instead of the current eight.

I check students' understanding of the Chilean election algorithm using questions such as the following:

- Given list, gender, and vote totals for each candidate, who won the election?
- Say that a set of candidates is "feasible" if it satisfies gender and list constraints. Given list, gender, and vote totals for each candidate, find every feasible selection that is not priority dominated by another feasible selection.
- Suppose that after calculating the number of representatives elected from each list, you determine that taking the top vote-getters from each list happens to satisfy gender parity. Does it follow that this selection priority dominates all other feasible selections? Justify your answer.

**4.2.3. References.** Cembrano et al. (2022) present more background on Chile's Constitutional Convention, along with some axiomatic characterizations of different selection rules. Cembrano et al. (2021) compare outcomes from the algorithm used in Chile to outcomes from alternative approaches.

I have cleaned data with all information required to reproduce the results of the 2021 election (candidate name, list, party, district, gender, and vote total) for all 1,278 candidates and made this data available through the following Google Sheet: <https://docs.google.com/spreadsheets/d/1vMa7RjvdQ5CuT-u01REDmeW7msJ9s6-IAOLK6vonBJc/edit?usp=sharing>.

Gender quotas for political office are actually quite common worldwide; see the Gender Quotas Database (International IDEA 2023) for more information.

Quotas also arise in other political contexts. For example, citizen's assemblies often seek to create panels that are representative of the entire population along a variety of dimensions such as age, gender, political affiliation, and more. Because these categories are not nested, finding even one feasible panel from a pool of volunteers is a nontrivial task. Flanigan et al. (2021) propose an algorithm to identify a distribution over panels that equalizes individual selection probabilities to the extent possible.

### 4.3. Civil Service and University Positions in India

**4.3.1. Overview.** In India, government civil service positions are awarded based on standardized exam scores. However, the constitution specifies a set of affirmative action policies that reserve positions for members of groups that have faced historical discrimination:

“Scheduled Tribes” (ST), “Scheduled Castes” (SC), and “Other Backward Classes” (OBC). More recently, seats have also begun to be reserved for women, people with disabilities, and other disadvantaged members of society. These latter reservations are referred to as “horizontal” reservations and targeted based on intersectional identity. That is, if there is a reservation for women, this is achieved by reserving some ST positions for women, some SC positions for women, some OBC positions for women, and some open positions for women.

The proper implementation of these policies has been contentious. Some outcomes have clearly run counter to the intent of the policies: in some cases, women from a scheduled caste were denied positions that were instead offered to women from more privileged castes who had *lower* exam scores. This and other concerns have resulted in many lawsuits that have reached district courts and the supreme court of India.

**4.3.2. Teaching This Material.** One interesting feature of this application is that there are two types of reservations, which align with the two algorithms for implementing H1B reserves discussed in Section 3.1. Reservations for scheduled tribes, scheduled castes, and other backward classes are referred to as “vertical” and are intended to go to applicants whose exam scores would otherwise not earn them a position. Other reservations (such as those for women) are referred to as horizontal, and any selected applicant eligible for these positions should count toward them (even if their exam score would earn a spot without this privilege). In other words, vertical reservations are intended to act in the spirit of an over-and-above reserve, and horizontal reservations should be implemented using the exemptions-first algorithm. I like to introduce this application after the class on H1B visa allocation to illustrate that both algorithms from that context have applications in other domains.

When there are multiple types of horizontal reservations, the eligible populations for these reservations are defined nonintersectionally. For example, within each vertical reservation there may be seats reserved for women and seats reserved for people with disabilities, but there are no seats reserved specifically for women with disabilities. As a result, with multiple horizontal reservations, reserve-eligible populations will not be nested. The allocation within each vertical reservation becomes equivalent to the general reserves problem discussed in Section 3.2: maximum weight-matching algorithms can be used to identify the priority-dominant feasible selection.

**4.3.3. References.** For more institutional detail, and a discussion of algorithms that implement horizontal

and vertical reservations as intended, see Sonmez and Yenmez (2022). This paper also has information on how to handle overlapping horizontal reservations, although a reader interested in only this issue may prefer to read an earlier working paper by the same authors (Sonmez and Yenmez 2019), or the work of Arnosti et al. (2024).

Recently, India passed a law implementing vertical reservations for members of “Economically Weaker Sections ([https://www.education.gov.in/sites/upload\\_files/mhrd/files/OM\\_EWS\\_Reservation.pdf](https://www.education.gov.in/sites/upload_files/mhrd/files/OM_EWS_Reservation.pdf)).” This is the first vertical reservation that is not caste based. Analyses of this new policy are given by Aygun et al. (2022) and Sonmez and Unver (2024).

A similar system is used for allocating seats at the prestigious India Institute of Technology campuses. For more details about the current algorithm, see Baswana et al. (2019). University positions reserved for OBC candidates can be “dereserved” if not enough candidates apply for these positions. Aygün and Turhan (2022) point out potential flaws with the current dereservation system and present an alternative approach.

#### 4.4. University Admissions in Brazil

**Overview.** In Brazil, university seats are reserved for students who graduate from public high schools. Among public high school students, some seats are reserved for students from low-income families, students belonging to ethnic minorities, and other groups. The current implementation sets separate quotas for each possible intersectional identity. So, for example, some seats are set aside for public high school students who are low income but not an ethnic minority, others for public high school students who are an ethnic minority but not low income, others for public high school students who are neither low income nor an ethnic minority, and still others for public high school students who claim both of these statuses.

This has the advantage of simplifying the selection algorithm: within each intersectional identity, the highest-achieving students are selected. However, it can also cause outcomes that feel unfair. For example, the cutoff score for low-income minorities might be above the cutoff score for low-income students who are not minorities. In this case, students with scores between these cutoffs would be harmed by revealing their minority status. Aygün and Bó (2021) provide suggestive evidence that these types of scenarios occur regularly.

**Teaching This Material.** There are several points that I make when teaching this content. First, this is an example where a desire for algorithmic simplicity arguably creates an unfair outcome. Second, this illustrates how intersectional identities can be used to

create nested (in this case, disjoint) eligibility categories, even if the nonintersectional identities are overlapping (as in the case of low-income students and minority students). For example, if the original goal was to reserve two seats for minority students and two seats for low-income students, requiring one low-income minority, one low-income nonminority, and one minority who is not low income ensures that the original goal will be satisfied (so long as there is a set of applicants who satisfy these new constraints). Third, the creation of nested categories typically involves adding constraints to the “original” problem. Thus, even if there is an outcome that priority dominates all others that satisfy the new constraints, it may not priority dominate other outcomes that satisfy the original constraints. Fourth, although any of the solutions proposed by Aygün and Bó (2021) are fair in the sense that they do not penalize applicants who belong to multiple targeted groups, none of these solutions is certain to find the unique priority-dominant feasible selection. This selection can be identified by the maximum matching algorithms referenced in Theorem 4.

**References.** Aygün and Bó (2021) provide more information about Brazilian laws and their implementation, and propose several alternatives that respect the spirit of affirmative action.

## 5. Conclusion

This paper provides materials to teach core optimization topics—such as computational complexity, integer programming, greedy algorithms, and matroid theory—while demonstrating their application to pressing societal issues, including visa allocation, elections, university admissions, and affordable housing.

Translating between real-world settings and optimization theory presents unique challenges. One challenge lies in translating ambiguous laws or policies into formal algorithms. The H1B visa lottery and New York’s community preference policy illustrate that seemingly minor differences in algorithmic implementation can lead to significant differences in outcomes. Policymakers and participants often recognize undesirable outcomes, but struggle to articulate precise objectives. Examples from this paper illustrate how abstract concepts like fairness and representation can be given precise mathematical definitions, thereby providing criteria for choosing among algorithms.

I hope these case studies equip educators with concrete, engaging examples that make optimization theory accessible, while inspiring students to apply their technical expertise to complex societal problems.

## Endnotes

<sup>1</sup> Regional quotas are determined by a formula specified in the 1990 Immigration and Nationality Act, which can be accessed at <https://www.justice.gov/sites/default/files/eoir/legacy/2009/03/04/IMMACT1990.pdf>. More information on the Diversity Immigrant Visa Program, including a map showing the regions, is available at [https://en.wikipedia.org/wiki/Diversity\\_Immigrant\\_Visa](https://en.wikipedia.org/wiki/Diversity_Immigrant_Visa). Regional quotas depend on recent immigration patterns and country population. As far as I know, the government does not post annual quota numbers, but they can be roughly inferred by observing the region of origin for immigrants issued visas through the program, which is available at <https://travel.state.gov/content/travel/en/us-visas/immigrate/diversity-visa-program-entry/diversity-visa-program-statistics.html>.

<sup>2</sup> See, for example, proposition 4 in Arnosti et al. (2024). When there are only upper quotas other than a single minimum quota on the total number of selected applicants, the problem of finding a feasible selection reduces to the independent set problem. When there are only lower quotas other than a single maximum quota on the total number of selected applicants, the problem of finding a feasible selection reduces to the set cover problem.

<sup>3</sup> See <https://www.dol.gov/agencies/whd/immigration/h1b>.

<sup>4</sup> Typically, multiple values of  $k$  satisfy this equation, but all lead to the same allocation of seats. Very rarely, there is no solution to this equation (i.e., if there is a single seat and two lists get exactly the same number of votes). In these cases, some tiebreaker must be decided upon.

<sup>5</sup> The actual algorithm used in Chile is slightly more complex, because of the presence of two levels of political representation: parties and lists (which consist of coalitions of parties). Thus, in reality, the Jefferson D’Hondt method is applied in two stages: first to determine the number of seats won by each list, and then to allocate these seats to parties within the list. If swapping candidates is necessary, the algorithm first tries to swap candidates from the same party, but if that is not possible, swaps of candidates of different parties from the same list are permitted.

## References

- Arnosti N (2019) The devilish details of H1B visa lotteries. Accessed January 7, 2025, <https://nickarnosti.com/blog/h1bvisas/>.
- Arnosti N (2022) Do “community preference” policies violate the fair housing act? (Part 3). Accessed January 7, 2025, <https://nickarnosti.com/blog/communitypreference-part3/>.
- Arnosti N, Bonet C, Sethuraman J (2024) Explainable affirmative action. *Proc. 25th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 310.
- Aygün O, Bó I (2021) College admission with multidimensional privileges: The Brazilian affirmative action case. *Amer. Econom. J. Microeconom.* 13(3):1–28.
- Aygün O, Turhan B (2022) How to de-reserve reserves: Admissions to technical colleges in India. *Management Sci.* 69(10): 6147–6164.
- Aygun O, Turhan B, Yenmez MB (2022) Challenges of executing EWS reservation efficiently. *Ideas for India* (March 12), <https://www.ideasforindia.in/topics/governance/challenges-of-executing-ews-reservation-efficiently.html>.
- Baswana S, Chakrabarti PP, Chandran S, Kanoria Y, Patange U (2019) Centralized admissions for engineering colleges in India. *INFORMS J. Appl. Analytics* 49(5):307–396.
- Beveridge A (2019) Expert report of professor Andrew A Beveridge. Accessed January 7, 2025, <https://www.antibiaslaw.com/sites/default/files/BevMain.pdf>.
- Budish E, Che YK, Kojima F, Milgrom P (2013) Designing random allocation mechanisms: Theory and applications. *Amer. Econom. Rev.* 103(2):585–623.
- Cembrano J, Correa J, Verdugo V (2022) Multidimensional political apportionment. *Proc. Natl. Acad. Sci. USA* 119(15):1–9.

- Cembrano J, Correa J, Diaz G, Verdugo V (2021) Proportional apportionment: A case study from the Chilean Constitutional Convention. *Proc. ACM Conf. Equity Access Algorithms, Mechanisms, Optim.* (Association for Computing Machinery, New York), 1–9.
- Dur UM, Kominers SD, Pathak PA, Sönmez T (2018) Reserve design: Unintended consequences and the demise of Boston's walk zones. *J. Political Econom.* 126(6):2457–2479.
- Flanigan B, Gözl P, Gupta A, Hennig B, Procaccia AD (2021) Fair algorithms for selecting citizens' assemblies. *Nature* 596(7873): 548–552.
- Gale D, Shapley LS (1962) College admissions and the stability of marriage. *Amer. Math. Monthly* 69(1):9–15.
- Gonczarowski YA, Nisan N, Kovalio L, Romm A (2019) Matching for the Israeli "mechinot" gap-year programs: Handling rich diversity requirements. *Proc. ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 321.
- International IDEA (2023) Gender quotas database. Accessed January 7, 2025, <https://www.idea.int/data-tools/data/gender-quotas-database>.
- Pathak PA, Rees-Jones A, Sonmez T (2023) Reversing reserves. *Management Sci.* 69(11):6417–7150.
- Pathak PA, Rees-Jones A, Sonmez T (2025) Immigration lottery design: Engineered and coincidental consequences of H1B reforms. *Rev. Econom. Statist.* 107(1):1–13.
- Siskin BR (2019) Expert report of Bernard R. Siskin, PhD. Accessed January 7, 2025, <https://int.nyt.com/data/documenthelper/1412-siskin-nyc-housing-report/70a77899cc711249d6d3/optimized/full.pdf>.
- Sonmez T, Unver U (2024) Informed neutrality in minimalist market design: A case study on a constitutional crisis in India. Working paper, Economics Department, Boston.
- Sonmez T, Yenmez B (2019) Affirmative action with overlapping reserves. Working paper, Economics Department, Boston.
- Sonmez T, Yenmez B (2022) Affirmative action in India via vertical, horizontal, and overlapping reservations. *Econometrica* 90(3):1143–1176.
- U.S. Department of State (2023) Instructions for the 2024 Diversity Immigrant Visa Program. [https://travel.state.gov/content/dam/visas/Diversity-Visa/DV-Instructions-Translations/DV-2024-Instructions-Translations/DV-2024-Instructions\\_508%20Accessible%207.5.2024.pdf](https://travel.state.gov/content/dam/visas/Diversity-Visa/DV-Instructions-Translations/DV-2024-Instructions-Translations/DV-2024-Instructions_508%20Accessible%207.5.2024.pdf).
- Wikipedia. Matroid. Accessed January 7, 2025, <https://en.wikipedia.org/wiki/Matroid>.
- Yokoi Y (2017) A generalized polymatroid approach to stable matchings with lower quotas. *Math. Oper. Res.* 42(1):238–255.