Group Fairness and Multi-Criteria Optimization in School Assignment

Santhini K. A. ✉
Indian Institute of Technology, Madras, India
GPM Government College Manjeshwar, India

Kamesh Munagala ✉
Duke University, Durham, NC, USA

Meghana Nasre ✉
Indian Institute of Technology, Madras, India

Govind S. Sankar ✉
Duke University, Durham, NC, USA

Abstract

We consider the problem of assigning students to schools when students have different utilities for schools and schools have limited capacities. The students belong to demographic groups, and fairness over these groups is captured either by concave objectives, or additional constraints on the utility of the groups. We present approximation algorithms for this assignment problem with group fairness via convex program rounding. These algorithms achieve various trade-offs between capacity violation and running time. We also show that our techniques easily extend to the setting where there are arbitrary constraints on the feasible assignment, capturing multi-criteria optimization. We present simulation results that demonstrate that the rounding methods are practical even on large problem instances, with the empirical capacity violation being much better than the theoretical bounds.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases School Assignment, Approximation Algorithms, Group Fairness

Digital Object Identifier 10.4230/LIPIcs.FORC.2025.20

Related Version *Full Version*: `https://arxiv.org/abs/2403.15623`

Supplementary Material *Software*: `https://github.com/govindssankar/School-Assignment`

Funding *Kamesh Munagala*: Supported by NSF grants CCF-2113798 and IIS-2402823.
Govind S. Sankar: Supported by NSF grants CCF-2113798 and IIS-2402823.

1 Introduction

Several societal decision-making problems manifest as assignment or matching. This includes the well-known school assignment or school redistricting problems, variants of which are implemented in several cities, including New York [3, 4], Boston [26], and San Francisco [5]. Typically, students express preferences over schools, and schools have *priorities* over different types of students and a fixed capacity to accept students. This assignment can then be modeled as a matching problem over students and school seats with one or two-sided preferences.

A standard solution approach is to find a stable matching [2, 3] but often, legislative or policy objectives require the problem to be augmented with additional features such as quotas and demographic requirements on the student body selected [1, 13, 32].

While many mechanisms for school choice attempt to reconcile these policy requirements with desirable traits like strategy-proofness or stability, we turn our focus towards a different consideration. Instead of designing mechanisms to achieve these traits, we examine schools

6th Symposium on Foundations of Responsible Computing (FORC 2025).
Editor: Mark Bun; Article No. 20; pp. 20:1–20:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

purely as resources to be allocated fairly among the students. We consider a viewpoint from the perspective of the student demographics, such as location, race, income, or parental education level, and seek a matching that is fair to these demographic groups. We adopt a model of *cardinal* preferences for students, where there is a numerical value to the utility that a student receives from being assigned to a school. This can capture outcomes such as average travel distance or assignment to higher-ranked schools. In such settings, the students are not strategic. Furthermore, concerns about stability may not be relevant when schools do not have preferences, or when demographic fairness is the primary goal. For example, as Abdulkadiroğlu and Sönmez [1] note,

> "During the redesign of the admissions process, BPS [Boston Public Schools] and the public considered the option of violating priorities at regular schools to promote student welfare. Likewise, the New York City high school system involves some schools at which respecting priorities emerges as a major policy goal and some other schools where priority violations may not be a cause of concern."

Analogous to previous work on cardinal preferences [5, 6, 27], we guarantee the existence of assignments that are fair to various demographic groups at the cost of adding a small number of extra seats to schools. Our model is general and allows for a variety of fairness objectives. We are required to assign students to schools subject to: (1) matching every student, and (2) being fair on the utilities to a pre-defined set of g (potentially overlapping) demographic groups of students while (3) respecting the capacities of schools as much as possible. As mentioned before, these groups can capture attributes like race, location, parental income, etc. Each student has cardinal utilities over schools, and the group fairness could either be captured by an objective defined over the total utility obtained by each group, or as a set of constraints capturing the same. We note that though we have presented school assignment as a canonical application, the assignment problem we consider is very general and has many applications such as job assignment or course assignment.

By now viewing the demographic groups as players in a fair division problem, where the schools are the resource to be allocated fairly, we can optimize for objectives such as minimum welfare or proportionality. Our main result (Theorem 7) is an algorithm to find an *exactly* proportional allocation, by adding $O(g^2)$ extra seats to the schools, where g is the number of demographic groups. Since g is small in practice, this is a mild violation.

Our model is similar to the one recently introduced in [23]. However, we significantly extend their results at the cost of a slightly larger capacity violation. Firstly, in their model, the demographic groups are required to form a partition of the students. We have no such restriction, and prove our results under the assumption that demographic groups can overlap arbitrarily - which is the case in practice when considering unrelated features such as race, sex, nationality, etc. Second, they assume that all students value schools in the same way. That is, there is a single, global utility function. Such an assumption is also not based in practice, where the student preferences may be globally correlated but exhibit large individual variations. Finally, they achieve only *approximate* proportionality, and cannot even guarantee that the allocation is Pareto-optimal on utilities. In contrast, our algorithms guarantee not only *exact* proportionality, but a host of other fairness measures such as maximin fairness and Nash welfare. Compared to their $O(g \log g)$ violation in total capacities, we violate capacities by a marginally larger amount ($O(g^2)$) but solve a much larger class of problems. To the best of our knowledge, our work is the first to achieve not just exact proportionality, but any concave fairness objective, in this setting while violating the school capacities by a small amount.

Lastly, we remark that our model can capture school-side preferences, albeit not to the extent of stability. Suppose that a school values each student across h different axes such as academics, sports, co-curriculars, etc. Then we can write h constraints for each school, to ensure that the student body assigned to the school is valued at at least (or at most) a certain threshold in each axes. In general, our framework can model such multi-objective optimization by incurring additive violations in the capacities as well as the constraints. In the school example, this would entail violating the capacities by an additive $O(hm)$, where m is the number of schools, and a similar factor in the constraints. Since $m \ll n$, the number of students, in most practical settings, this is still acceptable.

1.1 Model

Formally, there is a set S of n students divided into g possibly overlapping groups $S_1, S_2, \ldots, S_g$. There is a set T of m schools, and school $j \in T$ has capacity C_j. There is also a bipartite graph $G = (S \cup T, E)$, where $(i,j) \in E$ is an edge if for $i \in S, j \in T$, it is possible to assign student i to school j. Let $\vec{y}$ denote an assignment, where $y_e \in \{0,1\}$ denotes whether for edge $e = (i,j) \in E$, student $i \in S$ is assigned to school $j \in T$. This assignment is *feasible* if it satisfies the capacity constraint of each school, and each student is assigned to some school[1]. Let u_{ij} be the (non-negative) utility derived by student i, if assigned to school j.

Given an assignment $\vec{y} \in \mathcal{P}$, we define the utility of each group S_k as $U_k(\vec{y}) = \sum_{i \in S_k} \sum_{(i,j) \in E} u_{ij} y_{(i,j)}$. Let $\vec{U} = \langle U_1, U_2, \ldots, U_g \rangle$. The goal is to find a feasible assignment $\vec{y}$ that maximizes some fairness function on the utilities perceived by the g groups. Let $f(\cdot)$ be a non-decreasing concave function. Then, a general goal is to maximize

$$h(\vec{U}) = \sum_{k=1}^{g} f\left(U_k(\vec{y})\right).$$

In practice, several functions f can be reasonable. For instance, the celebrated *Nash welfare* objective sets $f = \log$, and the optimal solution $\vec{U^*}$ satisfies the following relation: For any other feasible utility vector $\vec{U'}$, we have the relation:

$$\frac{1}{g} \sum_{k=1}^{g} \frac{U'_k}{U^*_k} \leq 1. \tag{1}$$

The utilities $\vec{U^*}$ in this allocation are proportional for each of the groups, that is, $U^*_k \geq \frac{U'_k}{g}$, meaning each group gets at least $1/g$ of the utility it would have obtained had it been the only group in the system and the social welfare was maximized. This notion of proportionality is the objective considered in [23]. Note that Nash welfare is a much more general objective, capturing proportionality for various subsets of groups, and the approach in [23] does not extend to Nash welfare.

Other examples of utility functions that we can model include the **max-min** fairness objective, which maximizes

$$\min_{k=1}^{g} U_k$$

[1] Note that this is without loss of generality, since we can always add a dummy school with infinite capacity to ensure that every student can be matched. However, this naturally affects the fairness objective. For example, the proportional share of demographic groups can increase from this process.

and tries to make the least happy group as happy as possible. This can be modeled with simple linear constraints without needing the use of a convex function. Any objective over the utilities that has **Constant Elasticity of Substitution** (CES) with certain parameters also falls in this model. Such functions can be modeled as maximizing

$$\sum_{k=1}^{g} a_k (U_k)^r.$$

for non-negative real numbers a_k. When $r \leq 1$, this function is concave and can be maximized in our framework. Note that the $r > 1$ case models an inherently *unfair* allocation, since it is always better to allocate more utility to the best-off group.

Finally, we can also capture group fairness through arbitrary covering or packing constraints where we are explicitly given utility requirements for each group. The objective is to find an assignment where each constraint is satisfied. This can capture general *multi-criteria optimization* for assignment problems in a non-fairness context. Formally, given g arbitrary constraints, we obtain a solution that can violate the capacities by a small function of g, and satisfies the constraints up to an additive function of g and $|q_{\max}|$, the largest magnitude coefficient in the constraint matrix. We discuss this further below.

1.2 Our Results

Our main contribution is a set of polynomial-time approximation algorithms for this problem for arbitrary fairness objectives. In Section 2, we present two approximation algorithms based on rounding a natural convex programming relaxation, which yield somewhat different guarantees, as summarized in the theorem below.

▶ **Theorem 1.** *Given any monotone, concave fairness function f, let $\vec{U}^*$ be the utilities in the optimal solution. Then, there exist algorithms to compute an assignment $\vec{y}$ that satisfies relaxed school capacities $\vec{C}'$ and yields utilities $\vec{U}'$ with $U'_k \geq U^*_k$ for all groups $k \in \{1, 2, \ldots, g\}$, with one of the following guarantees:*

1. *A polynomial[2] running time and satisfies $C'_j \leq C_j + 1 + \delta_j$, where $\sum_j \delta_j \leq 2g$.*
2. *A $n^{O(g)}$ running time and satisfies $C'_j \leq C_j + \delta_j$ with $\sum_j \delta_j = O(g^2)$.*

Note that the latter algorithm is slower but yields better violation of capacities if $g \ll |T|$. For practical school choice scenarios [3], the number of students significantly outweighs the number of available schools, while the number of groups g is typically small, even constant. The above violations are, therefore, quite mild. We substantiate this via our empirical results, which we discuss later.

We also show that the school assignment problem mentioned above is NP-Hard for the max-min fairness objective, even when the number of schools or the number of groups is only two. The latter result extends to the proportionality objective. This motivates the need to relax the capacity constraints (as in Theorem 1) if our goal is to achieve a polynomial time algorithm.

▶ **Theorem 2.** *Suppose the number of groups g is part of the input, and the objective is to decide if the minimum utility received by any group is at least one. Then, the school assignment problem is* NP-Complete *even when there are only two schools.*

[2] Throughout the paper, we use this to mean polynomial in n, m, and g.

▶ **Theorem 3.** *Suppose the objective is to decide if an exactly proportional allocation exists. Then, the school assignment problem is weakly* NP-Complete*, even when the number of groups $g = 2$.*

We defer the proofs to Section A. We also note that to achieve the proportionality objective, it is known that a capacity violation of $\frac{g}{2}$ is required in the worst-case [23].

Packing/Covering Constraints and Multi-criteria Optimization

In Section 3, we present an extension of our framework to handle assignments with more general constraints. As an application, suppose the utility of a student for a school is multi-dimensional, capturing aspects like academic excellence, or location, or diversity of student body. The goal is to achieve at least a specified total utility value in each dimension. Such *multi-objective* optimization [16, 25, 28] can be modeled by covering or packing constraints, and we present a result similar to Theorem 1 for this general setting. Formally, we solve a linear relaxation of the following integer program, where we have a setting as in Section 1.1 but instead of the fairness objective and utilities, we have a matrix $Q \in \mathbb{R}^{n \times r}$ defining r linear constraints that we are required to satisfy. In other words, we wish to solve the following integer program:

$$\text{(IP)} \quad \begin{aligned} \sum_j y_{ij} &= 1 && \forall \text{ students } i && (2)\\ \sum_i y_{ij} &\le C_j && \forall \text{ schools } j && (3)\\ \sum_{i,j} q^{\ell}_{ij} y_{ij} &\ge Q_\ell && \forall \ell \in \{1, 2, \ldots, r\} && (4)\\ y_{ij} &\in \{0, 1\} && \forall\ i, j && (5) \end{aligned}$$

We show that Theorem 1 generalizes to this setting at the cost of incurring an additional additive loss proportional to $|q_{\max}| := \max_{i,j,\ell} \left|q^{\ell}_{ij}\right|$ and r, the number of rows in Q.

▶ **Theorem 4.** *For arbitrary constraints, when the linear programming relaxation of (IP) has a feasible solution, there are algorithms that output an integer solution*

- *In polynomial time, such that the constraints Equation (4) are preserved up to an additive $r \cdot |q_{\max}|$; and if each school is given one unit extra capacity, the total violation in Equation (3) over this is $2r$.*
- *In $n^{O(r)}$ time, such that the constraints Equation (4) are preserved up to an additive $O(r^2 \cdot |q_{\max}|)$; and the total violation in Equation (3) is $O(r^2)$.*

As a specific application, in Section 3.4 we study the assignment with ranks problem first considered in [18]. Here, each student ordinally ranks the schools with possible ties. An input signature $\vec{\rho}$ of length r, the goal is to find an assignment where the number of students who are assigned their first k choices (for all $k \le r$) is at least $\sum_{j=1}^{k} \rho_j$. In comparison to the algorithm in [18] that runs in time $n^{O(r^2)}$ and uses multivariate polynomial interpolation, our algorithms present substantial improvements in both runtime and ease of implementation at the cost of a small capacity violation, when students are given a small choice r of ranks. We show that this problem has an additional "monotonicity" in the constraints, that enable us to avoid the additive violation mentioned above.

▶ **Theorem 5.** *Given a feasible fractional solution to a matching with ranking instance with input signature ρ, there is an algorithm to output a matching with signature $\sigma \succ \rho$ that satisfies relaxed school capacities $\vec{C'}$ with one of the following guarantees:*

1. *A $\mathsf{poly}(n, r)$ running time and satisfies $C'_j \leq C_j + 1 + \delta_j$, and $\sum_j \delta_j \leq 2r$.*
2. *A $n^{O(r)}$ running time and satisfies $C'_j \leq C_j + \delta_j$, and $\sum_j \delta_j = O(r^2)$.*

Benchmark and Simulation

In Section 4, we present a benchmark for the group fairness objective, where we use an ILP to find the optimum violation in capacity needed to achieve the utilities generated by the convex program. Theorem 1 yields a theoretical upper bound on the capacity violation. However, we show that both the ILP benchmark as well as our rounding algorithms yield substantially better violations on realistic instances, hence showcasing the practicality of our approach. We also present empirical results for the aforementioned matching with ranks problem in Section C.

1.3 Techniques and Related Work

Our algorithm uses LP rounding and borrows ideas from the seminal Generalized Assignment Problem (GAP) rounding technique of Lenstra, Shmoys, and Tardos [19, 29]. Their iterative rounding procedure involves the observation that the number of fractional variables in a vertex solution to a linear programming relaxation is bounded. We build on this idea and apply it to a linear program written on paths and cycles instead of assignments, enabling us to combine it with a theorem of Stromquist and Woodall [31]. This approach was recently used in a similar model by the authors in [23], which they called "cake frosting". This theorem is a consequence of the celebrated ham-sandwich theorem [30], and is, hence, non-constructive. Using this technique, they achieve approximate proportionality while violating the total capacity by $O(g \log g)$, where g is the number of groups.

In contrast, we use convex programming relaxation to handle *arbitrary* fairness objectives such as proportional fairness, Pareto-optimality, and maximin fairness, vastly generalizing the space of objectives. Our method only loses $O(g^2)$ on the total capacity, while preserving utilities from the fractional relaxation. For instance, our method would achieve *exact* proportionality, and by Equation (1), it even achieves a generalization of this concept to subsets of groups. Further, we show empirically that the use of convex programming keeps the cake frosting instance very small and hence tractable.

At a high level, our main technical contribution is to show how the cake frosting method can be applied to certain types of fractional solutions, in particular, a vertex solution constructed via GAP rounding of the convex programming relaxation. We hence showcase the full power of the technique in [31]. In addition, as discussed above, our techniques extend smoothly to handle arbitrary covering or packing constraints on the allocations, which is motivated by multi-objective optimization and rank optimization.

We note that the idea of using cake frosting to round fractional solutions has appeared before for *packing problems* in Grandoni *et al.* [17], where the authors develop a PTAS for matchings in general graphs with $O(1)$ budget constraints on the set of chosen edges. At a high level, all these approaches – the ones in [17, 23] and our work – apply cake frosting to decompose paths and cycles to approximately preserve constraints, but differ in the details of how the paths and cycles are constructed from the integer or fractional solutions. For instance, in contrast to [23], which defines the frosting function based on schools, we define it based on students, hence avoiding an additive violation on the utility. Further, since [17] consider packing problems, their reduction to cake frosting is entirely different in the technical details.

Matching with Violations

Unlike many resource allocation problems, school assignments have flexibility in the capacities assigned to schools. Additional seats can be added with appropriate investments, or minor adjustments to class structures. Governments have also shown a willingness to add seats, particularly in situations where students would have gone unassigned [8]. Along these lines, several papers have considered such assignment problems with small capacity violations. These papers mostly fall into two categories - those that try to directly optimize the capacity violations in some form while achieving a set goal like stability or perfectness [9, 11, 15, 18] and those that optimize some other objective like fairness with provably small capacity violations [17, 21, 23]. Our model falls in the latter category – we wish to find an assignment that satisfies some notion of fairness while violating capacities by as little as possible.

Group Fairness

The school assignment problem that we study was first considered recently in Procaccia, Robinson, and Tucker-Foltz [23]. The only objective considered in this work is *proportionality* – in the assignment, each of the g demographic groups is required to achieve at least $1/g$ fraction of the utility it could have achieved had it been the only group in the system. Various such notions of group fairness have been studied in many contexts such as clustering [7, 14], knapsack [22], and matchings [12, 24]. The objective function in *Socially Fair k-clustering* [14], where the average clustering cost across each demographic group has to be minimized, is particularly similar to ours.

2 Approximation Algorithm: Proof of Theorem 1

In this section, we prove Theorem 1. We begin with a convex programming relaxation to the problem and then present two rounding schemes that yield the two guarantees in the theorem.

2.1 Convex Program Relaxation

Recall that there is a set S of n students divided into g possibly overlapping groups sets $S_1, S_2, \ldots, S_g$. There is a set T of schools, where school j has capacity C_j. Finally, there is a bipartite graph $G = (S \cup T, E)$ between the students and schools that represents possible assignments. An assignment of students to schools is a feasible solution $\vec{y} \in \mathcal{Q}$, where $\mathcal{Q}$ is the polytope defined by the following constraints:

$$\begin{array}{rcll} \sum_{j \in T} y_{ij} & = & 1 & \forall i \in S \\ \sum_{i \in S} y_{ij} & \leq & C_j & \forall j \in T \\ y_{ij} & \in & \{0,1\} & \forall (i,j) \in E \end{array}$$

Let $\mathcal{P}$ denote the linear relaxation of $\mathcal{Q}$, where the last constraint is replaced by $0 \leq y_{ij} \leq 1$. The first step is to write the following convex programming relaxation:

$$\text{Maximize} \sum_{k=1}^{g} f(U_k)$$

$$\begin{array}{rcll} \sum_{i \in S_k} \sum_{j \in T} u_{ij} y_{ij} & \geq & U_k & \forall \text{ groups } k \\ \vec{y} & \in & \mathcal{P} & \\ U_k & \geq & 0 & \forall \text{ groups } k \end{array}$$

This can be solved in polynomial time. Let the optimal solution to the convex program yield utility vector $\vec{U}^*$. We now need to round the following set of constraints so that the $\{y_{ij}\}$ values are integer. Denote this formulation as (LP1).

$$\vec{y} \in \mathcal{P}, \qquad \forall \text{ groups } k, \textstyle\sum_{i \in S_k} \sum_{j \in T} u_{ij} y_{ij} \quad \geq \quad U_k^* \quad .$$

We now present two rounding algorithms that yield the corresponding guarantees in Theorem 1.

2.2 Generalized Assignment Rounding

The first rounding algorithm is similar to rounding for generalized assignment (GAP) [19, 29] and is presented in Algorithm 1. We remark that the iterative procedure from Step 1-6 is not necessary; the same can be achieved with a single LP solution. We present it this way for ease of exposition. We show that it achieves the following guarantee, yielding the first part of Theorem 1.

▶ **Theorem 6.** *Algorithm 1 runs in polynomial time and finds an integer solution $\vec{y} \in \mathcal{P}$ that satisfies relaxed school capacities $\vec{C}'$ and yields utilities $\vec{U}'$, where*

1. *$U'_k \geq U^*_k$ for all groups k, and*
2. *$C'_j \leq C_j + 1 + \delta_j$, where $\sum_j \delta_j \leq 2g$.*

Algorithm 1 GAP Rounding.

```
1: repeat
2:     Obtain a vertex solution $\vec{y}$ to (LP1).
3:     for all $y_{ij} = b \in \{0, 1\}$ do
4:         Fix $y_{ij} = b$ and remove this variable from (LP1), and update the constraints.
5:     end for
6: until (LP1) is not modified.
7: For each remaining student $i$ (these have a degree more than 1), assign this student to
   $\text{argmax}_j\{u_{ij} | y_{ij} > 0\}$.
```

Proof of Theorem 6. We denote the number of incident edges with $y_{ij} > 0$ as the "degree" of a vertex. First, note that since each student i is assigned to $\text{argmax}_j\{u_{ij} | y_{ij} > 0\}$, the utility in the integer solution is at least that in (LP1).

We now bound the capacity violation. We note that Step 4 cannot violate any capacities since $\vec{y}$ was feasible for (LP1). It remains to argue that Step 7 does not incur too many capacity violations. Let E be the set of remaining edges with $y_{ij} \in (0, 1)$. Since $\vec{y}$ was an extreme point solution to (LP1), $|E|$ constraints of (LP1) must be tight.

At the beginning of Step 7, let T' be the set of remaining schools and among these, let $\hat{T}$ be those whose capacity constraints are tight. Let S' be the set of remaining students. Each student has a tight constraint associated with it. Suppose g' of the g constraints corresponding to the groups are tight. Since we have a vertex solution,

$$2|E| = 2(|S'| + |\hat{T}| + g') \tag{6}$$

From the Handshaking lemma, we also have $\sum_{v \in S' \cup T'} \deg(v) = 2|E|$. Combining this with Equation (6), we have,

$$\sum_{v \in S' \cup T'} \deg(v) - 2|S'| - 2|\hat{T}| = 2g' \implies \sum_{v \in T' \setminus \hat{T}} \deg(v) + \sum_{v \in S' \cup \hat{T}} (\deg(v) - 2) = 2g' \leq 2g$$

We know that each school in $\hat{T}$ has a degree of at least 2, since the capacity is an integer, and all assignment variables are strict fractions. Similarly, each student in S' has a degree at least 2. Therefore, every term in the above summation is non-negative. Let each student or school $v \in \hat{T} \cup S'$ have degree $2 + \delta_v$, while schools $v \in T' \setminus \hat{T}$ have degree δ_v. We will refer to these δ_v terms as the excess degrees. Then, the above implies

$$\sum_{v \in T' \cup S'} \delta_v \leq 2g. \tag{7}$$

To bound the capacity violation in Algorithm 1, we observe the following properties: First, if a school in $\hat{T}$ had degree 2, then it must have had a capacity of at least 1, and in the worst case, both students with edges to it will match to it. This leads to a violation of 1 in this school's capacity. Next, for any other school $v \in \hat{T}$, it again has capacity at least 1 and has $2 + \delta_v$ students applying to it. In the worst case, this leads to a capacity violation of at most $1 + \delta_v$. Finally, for schools $v \in T' \setminus \hat{T}$, since the degree is δ_v, this leads to a violation of at most δ_v.

In total, this leads to a capacity violation of one per school and the excess degrees $\sum_v \delta_v$ lead to an additional $2g$ violation overall. This completes the proof. ◀

2.3 Improved Capacity Violation via Cake Frosting

While the previous section provides a polynomial-time solution, we can improve the capacity violation bound, albeit at the cost of increased runtime. We achieve this by replacing the last step in Algorithm 1 with a more sophisticated 'cake frosting' technique, building on the work of [23]. We show the following theorem, corresponding to the second part of Theorem 1.

▶ **Theorem 7.** *There is a $n^{O(g)}$ time algorithm that computes an integer assignment $\vec{y} \in \mathcal{P}$ that satisfies relaxed school capacities $\vec{C'}$ and yields utilities $\vec{U'}$, such that:*

1. *$U'_k \geq U^*_k$ for all groups k; and*
2. *$C'_j \leq C_j + \delta_j$ with $\sum_j \delta_j = O(g^2)$.*

Paths and Cycles

Let G be the graph at the beginning of Step 7 in Algorithm 1. Recall that the maximum degree in G was 2, except for some vertices with excess degrees in Equation (7). We will process the graph into a graph of maximum degree 2, with some additional properties, in Algorithm 2.

At the end of the process, let $G(V, E)$ be the resulting graph on fractional edges. Any vertex has a degree of at most two, and hence we get a graph with the following structure: Every connected component is a path or a cycle; every student has degree exactly two, and is an internal node of a path or cycle; every school $j \in T'$ has capacity one and degree at most two; and finally, any school $j \in T' \setminus \hat{T}$ has degree one and capacity one, and is, therefore, a leaf of a path.

▶ **Lemma 8.** *Algorithm 2 violates the total capacity by an additional $4g$.*

Proof. For $j \in S_1$, let degree$(j) = 2 + \delta_j > 2$, implying $\delta_j \geq 1$. We increase the capacity by $1 + \delta_j \leq 2\delta_j$. For $j \in S_2$, the new capacity is the same as the original capacity. For $j \in T' \setminus \hat{T}$, suppose degree$(j) = \delta_j$, then we increase the capacity by δ_j. By Equation (7), the total increase is at most $4g$. ◀

Algorithm 2 Graph Modification.

```
for each student $i$ with degree strictly more than 2 do
    Add a capacity of one to $j^* = \text{argmax}_j\{u_{ij} | y_{ij} > 0\}$.
    Fix $y_{ij^*} = 1$ and remove this student.
end for
$S_1 = \{j | j \in \hat{T}, \text{degree}(j) > 2\}$.
$S_2 = \{j | j \in \hat{T}, \text{degree}(j) = 2, C_j = 2\}$.
for each school $j \in S_1 \cup S_2 \cup (T' \setminus \hat{T})$ do
    $d = \text{degree}(j)$.
    Create $d$ copies of $j$, each with capacity one.
    Assign (add an edge from) each $i$ with $y_{ij} > 0$ to a distinct copy of $j$.
                                                  ▷ Each new school has degree one.
end for
If a school has degree one, reduce its capacity to one.
```

In the graph G, suppose $e = (i, j)$; then we denote $x_e = y_{ij}$. This graph is a collection of paths and cycles. For non-leaf school or student v, the above conditions imply $x_{e_1} + x_{e_2} = 1$ if the two edges incident on v are e_1 and e_2. This follows because a degree-two school must belong to $\hat{T}$ and corresponds to a tight constraint, and any student is associated with a tight constraint. This implies the following claim:

▷ Claim 9. For every component (path or cycle) C of G, there is some $\alpha \in (0, 1)$ such that every even edge e in the component has $x_e = \alpha$ and every odd edge has $x_e = 1 - \alpha$.

Bounding the Number of Fractional Components

We view this fractional solution as follows. For component C, set $z_C = \alpha$ if $x_e = \alpha$ for every even edge. Let $u_C^{even}(i)$ be the utility that group i gets in the assignment that selects all even edges (and no odd edges) of component C. Let $u_C^{odd}(\ell)$ be the utility that group ℓ gets in the assignment that selects all odd edges of component C. We modify (LP1) to the following, where $\hat{U}_\ell^*$ is the modified utility after removing the integral variables.

$$\begin{aligned} &\forall \text{ groups } \ell, && \sum_C z_C \cdot u_C^{even}(\ell) + (1 - z_C) \cdot u_C^{odd}(\ell) \geq \hat{U}_\ell^* \\ &\forall \text{ components } C, && z_C \in [0, 1]. \end{aligned}$$

Let s denote the number of variables. In any extreme point solution, at least $s - g$ of the constraints $z_C \in [0, 1]$ are tight, which means that at most g of the z_C variables can be fractional, in $(0, 1)$. For all integral z_C, we select the even matching if $z_C = 1$ or the odd matching if $z_C = 0$. Remove these variables and rewrite the above LP just on the fractional variables.

Reduction to Cake Frosting

For the g components with fractional z_C, we need to find an integral solution that approximately preserves the utilities. This would be achieved if we could "interpolate" z_C fraction from the odd matching to the even matching. We can view this as a cake-frosting problem as in [23], where the g groups are the players. First, we convert each cycle into a path as follows: Pick some student i on this cycle, assign i to $\text{argmax}_j\{u_{ij} | y_{ij} > 0\}$, and delete this

Algorithm 3 Cake Frosting Rounding.

```
for every student $i$ do
    if $[\frac{i-1}{r}, \frac{i}{r}) \subseteq X$ then
        Choose the edge from the even matching for student $i$, and include $i$ in set $T_1$.
    else if $[\frac{i-1}{r}, \frac{i}{r}) \subseteq [0,1] \setminus X$ then
        Choose the edge from the odd matching for student $i$ and include $i$ in set $T_2$.
    else
        Assign $i$ to $\text{argmax}_{j'}\{u_{ij'}, y_{ij'} > 0\}$.
    end if
end for
```

student. This step increases the capacity of at most g schools by one and reduces each cycle to a path that begins and ends at a school. We now present a generalization of the "cake frosting" lemma first presented in [31] and used in [17, 23].

▶ **Lemma 10** (Cake Frosting Lemma). *Given g piecewise constant functions $f_\ell, \ell = 1, 2, \ldots, g$ with domain $[0,1]$, and given any $\alpha \in (0,1)$, there is a "perfect frosting" $X \subseteq [0,1]$ written as a union of at most $2g-1$ intervals such that for all ℓ:*

$$\int_X f_\ell(x)dx = \alpha \cdot \int_0^1 f_\ell(x)dx.$$

We now show how to apply the above lemma similarly to [23, 17]. Fix a path C. Let $z_C = \alpha$. Let there be r students in C, indexed from 1 to r. We divide the interval $[0,1]$ into r parts where $[\frac{i-1}{r}, \frac{i}{r})$ belongs to the i^{th} student.[3] Define for every group ℓ,

$$u_{even}(\ell, i) = \begin{cases} u_{ij} & \text{if the even matching assigns student } i \\ & \text{to school } j \text{ and student } i \text{ is in group } \ell \\ 0 & \text{Otherwise} \end{cases}$$

$$u_{odd}(\ell, i) = \begin{cases} u_{ij} & \text{if the odd matching assigns student } i \\ & \text{to school } j \text{ and student } i \text{ is in group } \ell \\ 0 & \text{Otherwise} \end{cases}$$

For $x \in [\frac{i-1}{r}, \frac{i}{r})$, define $f_\ell(x) = r(u_{even}(\ell, i) - u_{odd}(\ell, i))$.

Rounding Procedure

For path C, we now apply the cake frosting lemma to the function f as defined above, with $\alpha = z_C$ to find the perfect frosting X that is a union of at most $2g-1$ intervals. Given X, we construct the assignment as in Algorithm 3. The final algorithm applies this procedure separately to each of the g fractional paths. Note that the α value depends on the path.

Analysis

We first bound the utility of each group ℓ in path C. Define $T_3 := [r] \setminus (T_1 \cup T_2)$, i.e. the set of students in C not in T_1 or T_2. The utility of group ℓ in the solution is

[3] This is in contrast to the method in [23], which defines intervals based on schools.

$$
\begin{aligned}
&\sum_{i\in T_1} u_{even}(\ell,i) + \sum_{i\in T_2} u_{odd}(\ell,i) + \sum_{i\in T_3} \max(u_{even}(\ell,i), u_{odd}(\ell,i))\\
=&\sum_{i\in T_1}(u_{even}(\ell,j) - u_{odd}(\ell,j)) + \sum_{i\in[r]} u_{odd}(\ell,j) + \sum_{i\in T_3} \max(u_{even}(\ell,i), u_{odd}(\ell,i)) - u_{odd}(\ell,i)\\
\geq&\sum_{i\in T_1}(u_{even}(\ell,j) - u_{odd}(\ell,j)) + \sum_{i\in[r]} u_{odd}(\ell,j) + \sum_{i\in T_3} \left|\left[\frac{i-1}{r},\frac{i}{r}\right)\cap X\right|(u_{even}(\ell,i) - u_{odd}(\ell,i))\\
\\
=&\frac{1}{r}\sum_{i\in T_1}\int_{x\in[\frac{i-1}{r},\frac{i}{r})} f_\ell(x) + u_C^{odd}(\ell) + \frac{1}{r}\sum_{i\in T_3}\int_{x\in[\frac{i-1}{r},\frac{i}{r})\cap X} f_\ell(x)\\
=&\frac{1}{r}\int_{x\in X} f_\ell(x) + u_C^{odd}(\ell) = \frac{\alpha}{r}\cdot\int_{x\in[0,1]} f_\ell(x) + u_C^{odd}(\ell)\\
=&\alpha\cdot(u_C^{even}(\ell) - u_C^{odd}(\ell)) + u_C^{odd}(\ell) = \alpha\cdot u_C^{even}(\ell) + (1-\alpha)\cdot u_C^{odd}(\ell).
\end{aligned}
$$

The first equality follows by adding and subtracting $\sum_{i\in T_1\cup T_3} u_{odd}(\ell,i)$. The second line and the only inequality follows from the observation that $\left|\left[\frac{i-1}{r},\frac{i}{r}\right)\cap X\right| \leq 1$ and $\max(u_{even}(\ell,i), u_{odd}(\ell,i)) - u_{odd}(\ell,i) \geq 0$. The third line follows from the definition of f, the fourth line follows from the the structure of X and T_1, T_3, and the fifth follows from the cake frosting lemma. The above chain of inequalities shows that for each group ℓ, the integer solution has utility at least that of the fractional solution.

To bound the total capacity violation, note that Algorithm 3 violates the capacity by one at every interval boundary. By the Cake Frosting lemma, this is an additional violation of $O(g)$ per path, and hence $O(g^2)$ overall. This completes the proof of Theorem 7, and hence Theorem 1.

3 Generalization to Arbitrary Constraints

We now consider a more general setting. As before, we are given a set T of schools, where school j has capacity C_j, a set S of students, and a bipartite graph $G = (S\cup T, E)$ between the students and schools. The objective is to find an integral assignment $\vec{y}$ of all the students that satisfies an additional set of r covering or packing constraints (possibly with negative coefficients[4]). Define LP to be the linear relaxation of (IP) in Section 1.2 obtained by relaxing Equation (5) to $y_{ij}\in[0,1]$. Unlike the previous section, a given y_{ij} variable can appear in the constraints arbitrarily. We now show that both Theorems 6 and 7 generalize to this setting, completing the proof of Theorem 4.

3.1 Generalizing Theorem 6

Our algorithm runs in the following steps, which build on Algorithm 1.

1. Solve the linear programming relaxation LP, fix and remove the integral variables, and find a vertex solution. Let E be the set of fractional variables, and S' be the remaining students.
2. Rewrite LP on the variables E and without the capacity constraints Equation (3).
3. Keep eliminating integer variables, stopping at a vertex solution where all variables are fractional. Let E' be the remaining variables and S'' be the remaining students.
4. Set an arbitrary $y_{ij} > 0$ to 1 for each $i\in S''$.

[4] Note that the only place we require non-negativity in the coefficients is in solving the convex program.

▶ **Theorem 11.** *For arbitrary covering or packing constraints, when the linear programming relaxation has a feasible solution, there is a polynomial time algorithm that outputs an integer matching and that achieves the following guarantee:*

- *The constraints Equation (4) are preserved up to an additive $r \cdot q_{\max}$; and*
- *If each school is given one unit extra capacity, the total violation in Equation (3) over this is $2r$.*

Proof. First, the proof of Theorem 6 shows that regardless of how the students in S' are assigned, if each school is given one extra unit of capacity, then the total violation in capacity is at most $2r$.

Therefore, we can focus on assigning the students so that the constraints Equation (4) are not violated significantly. In Step (2), since any student in S'' has degree at least 2, we have $|E'| \geq 2|S''|$. Further, any extreme point in Step (3) has exactly $|E'|$ tight constraints. Since the number of potential tight constraints is at most $|S''| + r$, we obtain $|S''| \leq r$. Therefore Step (4) violates each constraint by an additive $r \cdot q_{\max}$, completing the proof. ◀

3.2 Generalizing Theorem 7

We next generalize Theorem 7. We first apply Algorithm 1 to the linear programming relaxation, stopping before Step 7. We then follow the procedure in Section 2.3 and sequentially apply Algorithms 2 and 3 to the fractional solution. To set up the cake frosting game to apply Algorithm 3, we view each of the r constraints in Equation (4) as a player of the cake frosting instance. Define $u_{even}(\ell, i) := q^{\ell}_{ij}$ where (i, j) is the even matching edge adjacent to j and define $u_{odd}(\ell, i)$ similarly. The only steps that are different are the assignment steps – Step 3 in Algorithm 2 and Step 7 in Algorithm 3. Here, we perform an arbitrary assignment of the students to the schools. We present all the details in Algorithm 4 for completeness.

▶ **Theorem 12.** *When the linear programming relaxation has a feasible solution, if r is the number of constraints Equation (4), there is a $n^{O(r)}$ time algorithm that outputs an integer matching and achieves the following guarantee:*

- *The constraints Equation (4) are preserved up to an additive $O(r^2 \cdot q_{\max})$; and*
- *The total violation in Equation (3) is $O(r^2)$.*

Proof. We first argue about the violation in Equation (4). The only steps that affect the constraints are the assignment steps – Steps 9 and 27 in Algorithm 4. In Step 9, the number of students assigned is $O(r)$ from Equation (7), while that in Step 27 is $O(r^2)$. If these students are arbitrarily assigned, each assignment loses an additive $q_{\max}$ in the constraint. Therefore, the overall additive loss is $O(r^2 \cdot q_{\max})$. Note that the bound on the capacity violation follows from the proof of Theorem 7 and holds even when these students are arbitrarily assigned. ◀

Using Theorem 4.12 in [17], we can improve Theorem 12 to the following corollary. We do this by guessing $8r^2/\epsilon$ chosen edges with highest utility for each group and subsequently applying Algorithms 2 and 3. Omitting the standard details yields the following corollary.

▶ **Corollary 13.** *Suppose the linear programming relaxation has a feasible solution and let r be the number of constraints Equation (4). Then, for any constant $\epsilon > 0$, there is a $n^{O(r^3/\epsilon)}$ time algorithm that outputs an integer matching and that achieves the following guarantee:*

- *The constraints Equation (4) are preserved up to a multiplicative factor of $(1 - \epsilon)$; and*
- *The total violation in Equation (3) is $O(r^2)$.*

3.3 Better Bounds for Monotonic Constraints

We next show that if the constraints Q have an additional monotonicity structure, then we can generalize Theorem 6 without the additive loss in the constraints. We say that Q satisfies *monotonicity* if for each student i, there is an ordering $\succeq_i$ of the schools $j_1 \succeq_i j_2 \succeq_i \ldots \succeq_i j_m$ such that for all $\ell \in \{1, 2, \ldots, r\}$ and $k \in \{1, 2, \ldots, m-1\}$, we have $q^{\ell}_{ij_k} \geq q^{\ell}_{ij_{k+1}}$.

▶ **Theorem 14.** *If the constraints Q are monotone and the linear programming relaxation has a feasible solution, there is a polynomial-time algorithm that outputs an integer matching and achieves the following guarantee:*

- *The constraints Equation (4) are preserved; and*
- *If each school is given one unit extra capacity, the total violation in Equation (3) over this is $2r$.*

▶ **Theorem 15.** *If the constraints Q are monotone, when the linear programming relaxation has a feasible solution, there is a $n^{O(r)}$ time algorithm that outputs an integer matching and that achieves the following guarantee:*

- *The constraints Equation (4) are preserved; and*
- *The total violation in Equation (3) over this is $O(r^2)$.*

Proof of Theorems 14 and 15. We proceed as in Theorems 6 and 7. At each point where the algorithm assigns a student to $j^* = \arg\max_{j \in X} u_{ij}$ for some set X, we simply assign it to $\min_k\{j_k \mid j_k \in X\}$. That is, assign it to the most preferred school (according to $\succeq_i$). This also preserves the r constraints because of monotonicity. ◀

3.4 Application: Weak Dominance of Ranks

As a special case, we consider the setting in [18]. Here, every student ranks the schools it has an edge to, and this ranking may have ties. Let r be the largest rank any student has, which can be much smaller than the number of schools. Given a matching, the rank of edge (p, q) is the rank of school q in student p's ranking. A matching M has *signature* $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_r)$ if it has σ_t rank t edges for every $t \in [r]$. We say that signature σ weakly dominates[5] signature ρ, or $\sigma \succ \rho$ if

$$\forall\ t \in [r], \sum_{t'=1}^{t} \sigma_{t'} \geq \sum_{t'=1}^{t} \rho_{t'}. \tag{8}$$

Given an input signature ρ, the goal is to find a matching whose signature weakly dominates ρ. We term this the matching with ranking problem. We have the following theorems, which directly follow from the observation that the constraints satisfy the monotonicity assumption. At each step where, for some set X, we assign i to $\arg\max_{j \in X} u_{ij}$, we instead assign it to its most preferred school from X. This preserves the signature of any fractional assignment. Our approach yields faster $n^{O(r)}$ time deterministic algorithms at the cost of small violations in capacities, whereas the algorithm in [18] is randomized and takes $n^{O(r^2)}$ time. This concludes the proof of Theorem 5.

[5] The authors of [18] use "cumulatively better than".

4 ILP Benchmark and Simulation Study

The goal of our simulation is to show the practicality of the convex programming framework in Section 2.1 as well as our rounding methods for addressing group fairness constraints in assignments.

ILP Benchmark

First, note that our framework yields a benchmark for capacity violation for concave group fairness objectives. We first solve the convex program in Section 2.1 to obtain the utility vector $\langle U_1^*, U_2^*, \ldots, U_g^* \rangle$. Subsequently, we can write an ILP to satisfy all utilities and violates total capacity the least as:

$$\text{Minimize } \sum_{j \in T} \delta_j$$

$$\begin{array}{rcll} \sum_{i \in S_k} \sum_{j \in T} u_{ij} y_{ij} & \geq & U_k^* & \forall k \\ \sum_{j \in T} y_{ij} & = & 1 & \forall i \in S \\ \sum_{i \in S} y_{ij} & \leq & C_j + \delta_j & \forall j \in T \\ y_{ij} & \in & \{0, 1\} & \forall i \in S, j \in T \\ \delta_j & \geq & 0 & \forall j \in T \end{array}$$

Theorem 1 says that the optimal value to this ILP is at most $\min\big(O(g^2), m + 2g\big)$. In our experiments, we compare the ILP benchmark for capacity violation with that of the rounding methods in Algorithm 1 and Algorithm 3. We show that the capacity violation for both the ILP and the rounding methods is much smaller than the theoretical bounds in Theorem 1, showing that group fairness functions have efficient near-optimal algorithms in the wild. We now present our empirical results for the general school assignment problem. We present our experiments for the rank dominance problem in Section C.

4.1 Empirical Results for School Assignment

Simulation Setup

We generate $r = 100$ random instances with $n = 1000$ students, $m = 10$ schools with equal capacity $C = 100$, and $g = 7$ groups. The instances are generated as follows. For every school j and student i, an edge is added independently with probability $p = \frac{3}{m}$. Afterwards, edges are added from students with degree zero to a school chosen uniformly at random so that the minimum degree is 1. Every school j has a "popularity measure" $\alpha_j \sim \texttt{Uniform}[0, 1]$. We set $u_{ij} := \hat{u}_{ij} \alpha_j$ where $\hat{u}_{ij} \sim \texttt{Uniform}[0, 1]$. This makes the utility of a school for different students correlated. The capacities C_j are set to minimize $\sum_j C_j$ so that all students can be feasibly assigned. This is found by solving an LP. Each group k has a parameter $\beta_k \sim \texttt{Uniform}[0, 1]$. Each student belongs to group k with probability β_k independently of other students and its other group identities. We set the fairness objective to be Nash Welfare, corresponding to $f = \log$, which by Equation (1) achieves proportionality and its generalization to subsets of groups.

Empirical Results

We first solve the convex program in Section 2.1 to find the utility vector $\vec{U_k^*}$. We then consider the following three approaches to find an integer assignment with small capacity violations while preserving the utilities $\vec{U_k^*}$.

Table 1 Results of experimental evaluation.

Procedure	Average violation	Range of violations
Optimal	0.66	[0,1]
GAP Rounding (Algorithm 1)	2.3	[0,6]
Cake Frosting (Algorithm 3)	1.24	[0,6]

- To find the integer assignment with minimum violation of capacities, $\sum_j \delta_j$, we solve the ILP described above.
- We solve the LP in Section 2.1 and round via Algorithm 1.
- We solve the LP in Section 2.1 and round via Algorithm 3.

The capacity violations are reported in Table 1. For these instances, Theorem 1 implies an integer assignment violating capacities by at most $m + 2g = 24$. For all approaches above, the capacity violation is much lower than the theoretical bound, both on average and per instance, with our rounding schemes finding solutions very close to the ILP benchmark. Further, both the ILP benchmark and Algorithm 3 run within a minute on a laptop on instances of this size. This is likely because most of the instances are already close to being integral. All the LP solutions had at most 30 fractional variables, with an average of 21.73. This shows the practicality of the convex programming relaxation.

5 Conclusion

We have presented a theoretically sound yet practical framework for handling group fairness and multi-objective optimization in capacitated assignment problems. Several open questions arise from our work. An immediate open question is to improve our theoretical bound on the capacity violation. We believe that a $O(g)$ violation should be possible in Theorem 1. More broadly, our framework uses cardinal utilities and it would be interesting to incorporate group fairness into ordinal preferences, as in stable matchings. An even more basic question is to consider random allocations with ordinal preferences [10, 20], and define group fairness for lotteries over allocations. Finally, it would be interesting to incorporate group fairness into other optimization problems with rounding-based approximation algorithms, for instance, scheduling and routing problems.

References

1 Atila Abdulkadiroglu. Generalized matching for school choice. *Unpublished paper, Duke University.[1311, 1312]*, 2011.

2 Atila Abdulkadiroglu and Aram Grigoryan. School choice. In Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani, editors, *Online and Matching-Based Market Design*, pages 180–200. Cambridge University Press, 2023.

3 Atila Abdulkadiroğlu and Tayfun Sönmez. School choice: A mechanism design approach. *American economic review*, 93(3):729–747, 2003.

4 Atila Abdulkadiroğlu, Nikhil Agarwal, and Parag A. Pathak. The welfare effects of coordinated assignment: Evidence from the new york city high school match. *American Economic Review*, 107(12):3635–89, December 2017. doi:10.1257/aer.20151425.

5 Maxwell Allman, Itai Ashlagi, Irene Lo, Juliette Love, Katherine Mentzer, Lulabel Ruiz-Setz, and Henry O'Connell. Designing school choice for diversity in the san francisco unified school district. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, pages 290–291, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3490486.3538271.

6 Itai Ashlagi and Peng Shi. Optimal allocation without money: An engineering approach. *Management Science*, 62(4):1078–1097, 2016. doi:10.1287/mnsc.2015.2162.

7 Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

8 Federico Bobbio, Margarida Carvalho, Andrea Lodi, Ignacio Rios, and Alfredo Torrico. Capacity planning in stable matching: An application to school choice. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC '23, page 295, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3580507.3597771.

9 Federico Bobbio, Margarida Carvalho, Andrea Lodi, and Alfredo Torrico. Capacity variation in the many-to-one stable matching, 2022. doi:10.48550/arXiv.2205.01302.

10 Anna Bogomolnaia and Hervé Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001. doi:10.1006/jeth.2000.2710.

11 Jiehua Chen and Gergely Csáji. Optimal capacity modification for many-to-one matching problems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pages 2880–2882, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. doi:10.5555/3545946.3599110.

12 Seyed Esmaeili, Sharmila Duppala, Davidson Cheng, Vedant Nanda, Aravind Srinivasan, and John P Dickerson. Rawlsian fairness in online bipartite matching: Two-sided, group, and individual. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5624–5632, 2023. doi:10.1609/AAAI.V37I5.25698.

13 Daniel Fragiadakis, Atsushi Iwasaki, Peter Troyan, Suguru Ueda, and Makoto Yokoo. Strategyproof matching with minimum quotas. *ACM Trans. Econ. Comput.*, 4(1), January 2016. doi:10.1145/2841226.

14 Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 438–448, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3442188.3445906.

15 Salil Gokhale, Samarth Singla, Shivika Narang, and Rohit Vaish. Capacity modification in the stable matching problem. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, pages 697–705, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. doi:10.5555/3635637.3662922.

16 Laurent Gourvès, Jérôme Monnot, Fanny Pascual, and Daniel Vanderpooten. Bi-objective matchings with the triangle inequality. *Theoretical Computer Science*, 670:1–10, 2017. doi:10.1016/j.tcs.2017.01.012.

17 Fabrizio Grandoni, R. Ravi, Mohit Singh, and Rico Zenklusen. New approaches to multi-objective optimization. *Math. Program.*, 146(1-2):525–554, 2014. doi:10.1007/S10107-013-0703-7.

18 Santhini K. A., Govind S. Sankar, and Meghana Nasre. Optimal matchings with one-sided preferences: Fixed and cost-based quotas. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pages 696–704, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. doi:10.5555/3535850.3535929.

19 Jan Karel Lenstra, David B. Shmoys, and Eva Tardos. Approximation algorithms for scheduling unrelated parallel machines. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 217–224, 1987. doi:10.1109/SFCS.1987.8.

20 Herve Moulin and Anna Bogomolnaia. A simple random assignment problem with a unique solution. *Economic Theory*, 19:623–636, April 2002. doi:10.1007/s001990100168.

21 Thanh Nguyen and Rakesh Vohra. Near-feasible stable matchings with couples. *American Economic Review*, 108(11):3154–3169, 2018.

22 Deval Patel, Arindam Khan, and Anand Louis. Group fairness for knapsack problems. *arXiv preprint arXiv:2006.07832*, 2020. arXiv:2006.07832.

23 Ariel D. Procaccia, Isaac Robinson, and Jamie Tucker-Foltz. School redistricting: Wiping unfairness off the map. In *Proc. ACM-SIAM SODA*, 2024.

24 Govind S Sankar, Anand Louis, Meghana Nasre, and Prajakta Nimbhorkar. Matchings with group fairness constraints: Online and offline algorithms. *arXiv preprint arXiv:2105.09522*, 2021. arXiv:2105.09522.

25 Paolo Serafini. Some considerations about computational complexity for multi objective combinatorial problems. In Johannes Jahn and Werner Krabs, editors, *Recent Advances and Historical Development of Vector Optimization*, pages 222–232, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

26 Peng Shi. Guiding school-choice reform through novel applications of operations research. *Interfaces*, 45(2):117–132, 2015. doi:10.1287/INTE.2014.0781.

27 Peng Shi. Optimal priority-based allocation mechanisms. *Management Science*, 68(1):171–188, 2022. doi:10.1287/mnsc.2020.3925.

28 Natsumi Shimada, Natsuki Yamazaki, and Yuichi Takano. Multi-objective optimization models for many-to-one matching problems. *Journal of Information Processing*, 28:406–412, 2020. doi:10.2197/IPSJJIP.28.406.

29 David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993. doi:10.1007/BF01585178.

30 A. H. Stone and J. W. Tukey. Generalized “sandwich” theorems. *Duke Mathematical Journal*, 9(2):356–359, 1942. doi:10.1215/S0012-7094-42-00925-6.

31 Walter Stromquist and D.R Woodall. Sets on which several measures agree. *Journal of Mathematical Analysis and Applications*, 108(1):241–248, 1985. doi:10.1016/0022-247X(85)90021-6.

32 Tayfun Sönmez and M. Bumin Yenmez. Affirmative action in india via vertical, horizontal, and overlapping reservations. *Econometrica*, 90(3):1143–1176, 2022. doi:10.3982/ECTA17788.

A Proofs of NP-Hardness results

Proof of Theorem 2. We reduce from Set Cover with a collection $\mathcal{C}$ of sets and a universe U of elements. Suppose the goal is to decide if a set cover instance has k sets that cover U. Then each element becomes a group and each set a student. A student belongs to a group if the corresponding set covers the corresponding element. There are two schools, s_1 and s_2. The former school has capacity k and the latter has capacity ∞. Each student has utility 1 for s_1 and 0 for s_2. Then the goal of matching k students to s_1 to give each group utility at least one is exactly the same as finding a set cover of size k, completing the proof. ◀

Proof of Theorem 3. We reduce from Partition. Given a set of numbers $x_1, .., x_n$, the goal is to decide if there is a subset of sum exactly $X/2$ where $X = \sum_i x_i$. For every number x_i, create two students p_i and q_i, and one school S_i of capacity 1. There is also a dummy school S_0 of capacity n. The students p_i and q_i have edges only to S_i and S_0, where p_i and q_i have utility x_i for S_i and 0 for S_0. All the p_i students belong to group 1 and all the q_i students belong to group 2. We want to find a matching that gives utility $X/2$ to both groups, which is the proportional share.

Suppose there is a subset T of the numbers that sums to exactly $X/2$. Then for every $x_i \in T$, we assign p_i to S_i and q_i to S_0. For every $x_i \notin T$, we assign q_i to S_i and p_i to S_0. Both groups get utility $X/2$ each. The reverse direction is similar, completing the proof. ◀

B Algorithm for Theorem 12

Algorithm 4 Algorithm for Theorem 12.

```
repeat
    Obtain a vertex solution $\vec{y}$ to (LP1).
    for all $y_{ij} = b \in \{0, 1\}$ do
        Fix $y_{ij} = b$ and remove this variable from (LP1), updating the constraints as needed.
    end for
until (LP1) is not modified.
for each student $i$ with degree strictly more than 2 do
    Add a capacity of one to an arbitrary school $j$ with $y_{ij} > 0$.
    Fix $y_{ij} = 1$ and remove this student.
    Decrease the capacity of $j$ correspondingly.
end for
$S_1 = \{j | j \in \hat{T}, \text{degree}(j) > 2\}$.
$S_2 = \{j | j \in \hat{T}, \text{degree}(j) = 2, C_j = 2\}$.
for each school $j \in S_1 \cup S_2 \cup (T' \setminus \hat{T})$ do
    $d = \text{degree}(j)$.
    Create $d$ copies of $j$, each with capacity one.
    Assign (add an edge from) each $i$ with $y_{ij} > 0$ to a distinct copy of $j$.
end for
If a school has degree one, reduce its capacity to one.
Set up the Cake Frosting instance as described in the text. Let $X$ be a perfect frosting.
for every student $i$ do
    if $[\frac{i-1}{r}, \frac{i}{r}) \subseteq X$ then
        Choose the edge from the even matching for student $i$.
    else if $[\frac{i-1}{r}, \frac{i}{r}) \subseteq [0, 1] \setminus X$ then
        Choose the edge from the odd matching for student $i$.
    else
        Assign $i$ to some $j$ with $y_{ij'} > 0$.
    end if
end for
```

C Empirical Results for Weak Dominance of Ranks

Simulation Setup

We generate 100 random instances with $n = 1000$ students, $m = 10$ schools with maximum rank $r = 8$. The capacities C_j are set so to minimize $\sum_j C_j$ so that all students can be feasibly assigned. This is found by solving an LP. These instances are generated as in section 4. That is, for every school j and student i, an edge is added independently with probability $p = \frac{3}{m}$. For every student, we select a random permutation of the schools in its neighborhood to obtain a ranking of the schools for that student.

In the weak dominance of ranks setting, we generate the input signature $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_r)$ as follows: For $t \in [r]$, let M_t denote a maximum matching on edges of rank up to t in the generated instance. We set σ_1 to be a random number between $0.9|M_1|$ and $|M_1|$. For, $i = 2, \ldots, r$, to set σ_i, we select a random number between 0 and $|M_i| - \sum_{t'=1}^{i-1} \sigma_{t'}$.

Empirical Results

To decide whether there exists a feasible solution for the instance with the given signature, we solve the linear relaxation of the IP defined in Section 1.2. Out of the 100 instances, 79 of them admit a feasible solution, of which the solution is fractional in 26 instances.

Next, for the instances where the LP gives a fractional solution, we obtain an integral solution using the algorithm of Theorem 14. This yields capacity violations of at most 4, with an average violation of 0.53. Subsequently, we use the algorithm of Theorem 15 to obtain an integral solution. This yields capacity violation of at most 2 with an average violation of 0.07. As in the previous experiment, we observe that the violation values are much better than what the theoretical bounds predict.