Allocating with Priorities and Quotas: Algorithms, Complexity, and Dynamics

SIDDHARTHA BANERJEE, Cornell University, USA
MATTHEW EICHHORN, Cornell University, USA
DAVID KEMPE, University of Southern California, USA

In many applications such as rationing medical care and supplies, university admissions, and the assignment of public housing, the decision of who receives an allocation can be justified by various normative criteria (ethical, financial, legal, etc.). Such settings have motivated the following *priority-respecting allocation problem*: several categories, each with a quota of interchangeable items, wish to allocate the items among a set of agents. Each category has a list of eligible agents and a priority ordering over these agents; agents may be eligible in multiple categories. The goal is to select a *valid* allocation: one that respects quotas, eligibility, and priorities and ensures Pareto efficiency.

We provide a complete algorithmic characterization of all valid allocations, exhibiting a bijection between sets of agents who can be allocated and maximum-weight matchings under carefully chosen rank-based weights. While prior work provides a polynomial-time algorithm to locate a valid allocation, our characterization admits a simpler algorithm that enables two wide-reaching extensions:
1. Selecting valid allocations that satisfy additional criteria: Via three examples — inclusion/exclusion of some chosen agent; agent-side Pareto efficiency vs. welfare maximization; and fairness from the perspective of allocated vs. unallocated agents — we show that finding priority-respecting allocations subject to some secondary constraint straddles a complexity knife-edge; in each example, one problem variant can be solved efficiently, while a closely related variant is NP-hard.
2. Efficiency-envy tradeoffs in dynamic allocation: In settings where allocations must be made to T agents arriving sequentially via some stochastic process, we show that while insisting on zero priority violations leads to an $\Omega(T)$ loss in efficiency, one can design allocation policies ensuring that the *sum* of the efficiency loss and priority violations in hindsight is $O(1)$ (under mild regularity conditions on the arrival process).

CCS Concepts: • **Theory of computation → Algorithmic game theory**; *Solution concepts in game theory*.

Additional Key Words and Phrases: Pareto Efficiency, Priorities, Quotas, Online Allocation, Public Goods

ACM Reference Format:
Siddhartha Banerjee, Matthew Eichhorn, and David Kempe. 2023. Allocating with Priorities and Quotas: Algorithms, Complexity, and Dynamics. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC '23), July 9–12, 2023, London, United Kingdom.* ACM, New York, NY, USA, 32 pages. https://doi.org/10.1145/3580507.3597733

1 INTRODUCTION

A core socio-economic question is how to ration scarce resources without money. While not new, this question has been forcefully reintroduced into public consciousness by COVID-19 [Andrews et al., 2021, Binkley and Kemp, 2020, Emanuel et al., 2020, Pathak et al., 2021, White and Lo, 2020].

EC '23, July 9–12, 2023, London, United Kingdom

ACM ISBN 979-8-4007-0104-7/23/07…$15.00
https://doi.org/10.1145/3580507.3597733

Defining "good" allocations is far from straightforward, as legal, financial, and ethical considerations can lead to nuanced, often clashing, requirements. For example, consider the following:

Academic Fellowships: Donors often define qualification requirements for named scholarships to promote students with certain demographics/backgrounds/skills.
Medical Care: The COVAX program set standards for the equitable distribution of vaccines in developing countries, prioritizing vaccination of groups such as healthcare workers, the elderly, and individuals with comorbidities [COVAX].
Primary School Enrollment: In Boston, half of a school's seats are reserved for students in the neighborhood, and priority is given to siblings [Abdulkadiroğlu et al., 2005]. Chicago requires that schools allocate roughly 25% of seats to each of four socio-economic tiers [Benabbou et al., 2019]. Chile's School Inclusion Law defines which factors can/cannot be used to prioritize students, and has quotas for students with economic hardships [Correa et al., 2021].
Public Housing: Singapore's 1989 Ethnic Integration Policy places quotas on the number of public housing units that may be allocated to each of three major ethnic groups [Benabbou et al., 2018].

The above settings broadly share the following features: a resource (scholarships, vaccines, school seats, housing) must be rationed among agents, whose number typically exceeds the available resource budget. The budget is split into several *categories*, each of which has a *quota* the category is responsible for distributing — this is sometimes due to physical constraints (different schools/housing projects), and at other times to implement some social norm (fellowship funds reserved for local/international/under-represented students; vaccine quotas for countries/states/target populations). Each category has rules to determine which agents are *eligible* for allocation. Each agent wants up to a single unit of the resource, but is indifferent as to which category allocates that unit[1]. Agents may be eligible in multiple categories, so categories must coordinate to maximize allocations. Finally, categories often define rankings (or *priorities*) over eligible agents, which are intended to help choose (and justify) which eligible agents get allocated. These rankings are often idiosyncratic, so there may be no natural way to compare agents across categories.

To understand how the above features (quotas, eligibility, priorities) restrict allocations, we build on the framework introduced by Pathak et al. [2021], which has led to a line of work aiming to understand its properties [Aziz and Brandl, 2021, Biró and Gudmundsson, 2021, Delacrétaz, 2021]. We briefly summarize the framework below; see Section 2 for a formal model.

The Priority-Respecting Allocation Problem
- q resource units, split into quotas q_c for categories $c \in C$, must be rationed to agents $\mathcal{A}$.
- Each category c has a set $\mathcal{E}_c \subseteq \mathcal{A}$ of *eligible* agents, and a priority order $\succeq_c$ over $\mathcal{E}_c$.
- Agent a is allocated $x_{a,c}$ from each category c, with $\sum_{c \in C} x_{a,c} \leq 1$. (*unit demand*)
- Category c can allocate only to agents in $\mathcal{E}_c$. (*eligibility-respecting*)
- Category c can allocate up to its quota, i.e., $\sum_{a \in \mathcal{A}} x_{a,c} \leq q_c$. (*quota-respecting*)
- Category c can allocate to agent a only once all higher priority agents are allocated i.e., $x_{ac} > 0 \implies \sum_{c' \in C} x_{a',c'} = 1$ for all agents $a' \succ_c a$. (*priority-respecting*)

The above problem tries to formalize what policymakers desire when using quotas, eligibility rules, and priorities, by providing a test for determining whether an allocation "respects" these requirements or not. The first two conditions impose that a category should only allocate from its quota, and only to eligible agents — these are standard and easily implemented. The third condition

[1]This may not hold in all settings — for example, families do have preferences between schools and housing units. We return to this issue in Section 4.2. Nevertheless, it is true up to first order that agents prefer being allocated to staying unallocated.

interprets priorities as a requirement that a category never allocate to an agent if a higher-priority agent has not been satisfied. This requirement is trickier to implement (and verify) since the higher-priority agents may receive allocations from any category. Nevertheless, the axioms are easy to satisfy — for example, each category can sequentially pick the highest-priority unallocated agent(s) in its eligibility list (more generally, via *serial dictatorship*; see Section 3.2).

One issue with the above requirements, however, is that they do not consider the "efficiency" of an allocation. For example, allocating to no one satisfies all the requirements. More problematic are settings with partial eligibility, where even if each category allocates maximally (i.e., until it exhausts its quota or eligibility list), the allocation may still end up wasting resources (for example, see Fig. 1). A natural additional requirement, therefore, is for allocations to be *Pareto efficient* — whereby there is no way for agents to exchange allocations such that at least one agent ends up gaining while no one is worse off. Ensuring Pareto efficiency in addition to the above requirements, however, seems challenging, and prior work [Abdulkadiroğlu and Grigoryan, 2021, Delacrétaz, 2021, Pathak et al., 2021] states this as an open question. More recently, Aziz and Brandl [2021] provide a scheme for finding a particular maximum-size (and hence Pareto efficient) allocation by solving $|\mathcal{A}|$ bipartite matching problems — this result, however, does not give any insight into general Pareto efficient valid allocations, and/or how one can select from among such allocations to satisfy some secondary objective. Figs. 1 and 2 give some intuition into the challenge of finding priority-respecting allocation — in particular, Fig. 2 shows that unlike maximum matchings, the set of priority-respecting allocations is not necessarily convex.

1.1 Our Contributions

Our work aims to characterize the set of *valid allocations*: those which respect eligibility, quotas, and priorities, and are Pareto efficient (see Section 2). Our main result is paraphrased as follows:

> *A set of agents can be allocated via a valid allocation if (Theorem 1) and only if (Theorem 2) they are allocated under the maximum matching for a weighted matching instance with edge weights picked from a certain valid set.*

The set of valid weights (Definition 2) is based on perturbing[2] the unweighted matching objective such that the perturbations are consistent with the priorities, and the total perturbation is small (at most $1/2$). As an immediate consequence, we get that *every* valid allocation allocates the same number of units, which moreover equals the size of an optimal matching *without* priority requirements. We also show that although the set of valid allocations is non-convex, every fractional valid allocation is realized as a convex combination of integral valid allocations (Proposition 2).

More importantly, our transformation of the problem of locating valid allocations into cardinal welfare maximization enables two wide-reaching and practical extensions. First, in Section 4, we consider how to select valid allocations satisfying additional criteria through three case studies:

(1) **Valid allocations with agent inclusion/exclusion**: In Section 4.1, given an agent $a \in \mathcal{A}$, we ask if one can find a valid allocation $\mathbf{x}$ that *excludes* a (i.e., $\sum_{c \in C} x_{a,c} = 0$), or *includes* a (i.e., $\sum_{c \in C} x_{a,c} > 0$). We show that while the former problem can be efficiently solved, the latter is NP-hard (Proposition 4). These results give a glimpse into the strange algorithmic landscape of valid allocations; note that both of these problems can be efficiently handled for maximum matchings and stable matchings (via the LP characterization of Vande Vate [1989]).

(2) **Incorporating agent preferences**: In Section 4.2, we augment the basic priority-respecting allocation problem by incorporating agent preferences for categories. We show how to

[2]Importantly, the perturbations can be *local* (i.e., the edge weight between agent a and category c only depends on a's position in c's priority order); this is surprising, as we show that no such result is possible for the set of *stable* matchings (Proposition 12 in Appendix C).

efficiently find allocations that respect eligibility, quotas and priorities, and are also Pareto efficient under the agents' preference orders (Theorem 3). On the other hand, we show that the problem of selecting a valid allocation that maximizes practically any aggregate function of agents' utilities is NP-hard (Theorem 4).

(3) **Inner/outer allocation thresholds**: In Section 4.3, we consider the problem of selecting valid allocations that optimize some score based on the *inner allocation threshold* (the lowest-priority agent *allocated* in each category) or, alternately, the *outer allocation threshold* (the highest-priority agent in each category who remains *unallocated*). Understanding these thresholds is important for auditing the "fairness" of an allocation. We show that optimizing over inner thresholds can be done efficiently (Propositions 5 and 6), while optimizing over outer thresholds is NP-hard (Propositions 7 and 8).

Each of these cases demonstrates that selecting valid allocations straddles the line of computational efficiency; one possible variant of each case study admits an efficient algorithm (based on our main approach in Section 3), while a closely related variant is computationally hard.

Online Priority-Respecting Allocations with Dynamic Arrivals: Finally, we consider an online variant of the priority-respecting allocation problem. Ours appears to be the first work that develops algorithms and performance guarantees for online settings, even though several of the original motivations for the priority-respecting allocation model were intrinsically tied to online allocation (of vaccines/medical supplies) [Pathak et al., 2021]. In the setting we consider, agents belong to one of a small number of observable types. Agents arrive one at a time over T rounds via some (known) underlying stochastic process, and the principal, after observing each agent's type, must immediately and irrevocably decide to either allocate this agent a unit from some category or leave the agent unallocated. We demonstrate that completely forbidding priority violations leads to $\Omega(T)$ regret with all but exponentially small probability. However, by incorporating priority violations into the objective, our LP formulation of the problem enables the development of a Bayes selector online algorithm for which the sum of the expected efficiency loss and priority violations in hindsight is $O(1)$ (i.e., independent of the number of arriving agents and the resource budgets, but depending polynomially on the number of types and categories).

In the interest of space, a discussion of related work can be found in Appendix A. In addition, many of our formal proofs have been relegated to Appendix B. Finally, a more complete discussion of the connection of our problem to Scarf's Lemma and stable matching can be found in Appendix C.

2 MODEL

Resources, Categories, Quotas and Agents: A set $\mathcal{A}$ of *agents* compete for q units of a resource. The units are distributed to a set C of *categories*, through which they are allocated. Each category $c \in C$ is given an integer *quota* of q_c units to allocate, such that $q = \sum_c q_c$.

Each agent is unit-demand, i.e., can consume at most one unit of the resource. For the initial part, we assume that agents are indifferent as to which category provides their allocation; in Section 4.2, we discuss how to incorporate agent utilities in this setting.

Eligibility and Priorities: Each category partitions $\mathcal{A}$ into a set of *eligible* and *ineligible* agents. The eligible agents are further partitioned into *priority* tiers.

Formally, each category $c \in C$ has an associated *eligible set* of agents $\mathcal{E}_c \subseteq \mathcal{A}$, and a total preorder $\succeq_c$ over $\mathcal{E}_c$. Given any two agents $a, a' \in \mathcal{A}$, $a \succeq_c a'$ denotes that a has weakly higher priority than a' in c. We write $a \succ_c a'$ when $a \succeq_c a'$ and $a' \not\succeq_c a$, so a has (strictly) higher priority in c. Given any agent a and any category c, we define the *rank* of a in c, denoted by $r_c(a)$, to be the length ℓ of the longest chain $a_1 \succ_c a_2 \succ_c \cdots \succ_c a_\ell = a$ with each $a_i \in \mathcal{A}$. Note that $1 \le r_c(a) \le |\mathcal{A}|$. We visualize instances using charts in the style of Fig. 1.

Allocation 1

α (1)	β (1)	γ (1)
$\boxed{c}$	$\boxed{a}$	b , c
	b	a , $\boxed{d}$

Allocation 2

α (1)	β (1)	γ (1)
c	$\boxed{a}$	b , $\boxed{c}$
	b	a , d

Allocation 3

α (1)	β (1)	γ (1)
$\boxed{c}$	$\boxed{a}$	$\boxed{b}$, c
	b	a , d

Allocation 4

α (1)	β (1)	γ (1)
$\boxed{c}$	a	b , c
	$\boxed{b}$	$\boxed{a}$, d

Fig. 1. *Four (integer) allocations in an instance with $C = \{\alpha, \beta, \gamma\}$ with quotas $(1, 1, 1)$, and $\mathcal{A} = \{a, b, c, d\}$. Each category c lists its eligible agents $\mathcal{E}_c$, where the agents listed in the i'th row have rank $r_c(\cdot) = i$.*
— Allocation 1 violates **[PR]***: d is allocated in category γ, but b, who has higher priority, remains unallocated.*
— Allocation 2 violates **[PE]***, as it is Pareto dominated by Allocation 3; note, however, that it is non-wasteful.*
— Allocations 3 and 4 are both valid (Definition 1) and allocate to the same set of agents.
— Allocation 4 violates **[CS]** *(Section 3.2), as β and γ can swap to allocate to higher-priority agents.*

Desiderata for Valid Allocations: Our goal is to find allocations that respect eligibility, quotas, and priorities. Formally, a (fractional) allocation is a function $\mathbf{x} : \mathcal{A} \times C \to [0, 1]$, with $\sum_c x_{a,c} \leq 1$ for each $a \in \mathcal{A}$ (since agents are unit-demand). If all $x_{a,c} \in \{0, 1\}$ (i.e., the matching is integral), then $\mathbf{x}$ coincides with an allocation map $\varphi : \mathcal{A} \to C \cup \{\perp\}$ assigning each agent to either a category (through which they are allocated) or the outside option $\perp$ (if they remain unallocated).

Moreover, the allocation must satisfy the three desiderata given below. These were proposed for integral allocations by Pathak et al. [2021]; we state the generalization for fractional matchings due to Delacrétaz [2021] since they naturally specialize to the integral case.

[QR] Quota Respecting: No category allocates more units than its quota.

$$\sum_{a \in \mathcal{A}} x_{a,c} \leq q_c \quad \text{for all } c \in C.$$

[ER] Eligibility Respecting: Agents are allocated only through eligible categories.

$$x_{a,c} = 0 \quad \text{for all } c \in C, a \notin \mathcal{E}_c.$$

[PR] Priority Respecting: An agent receives any allocation through category c only after all higher-priority agents in c are *fully* allocated.

$$x_{a',c} > 0 \ \wedge \ a \succ_c a' \implies \sum_{c' \in C} x_{a,c'} = 1 \quad \text{for all } a, a' \in \mathcal{A}, c \in C.$$

While the above desiderata determine which allocations are invalid due to violating the prescribed properties, they still admit many allocations that are undesirable. In particular, setting all $x_{a,c} = 0$ satisfies the preceding desiderata. A natural additional property is that any chosen allocation be *Pareto efficient*, which we formalize as follows.

[PE] Pareto Efficient: An allocation $\mathbf{x}$ satisfying **[QR]**, **[ER]**, and **[PR]** is Pareto efficient if there is no other allocation $\mathbf{y}$ satisfying these desiderata in which one agent gets a strictly greater allocation and no one receives a smaller allocation. Formally, for every $\mathbf{y}$ satisfying **[QR]**,**[ER]**, and **[PR]**,

$$\text{there is an } a \in \mathcal{A} \ : \sum_{c \in C} y_{a,c} > \sum_{c \in C} x_{a,c} \implies \text{there is an } a' \in \mathcal{A} \ : \sum_{c \in C} y_{a',c} < \sum_{c \in C} x_{a',c}.$$

As a special case of this definition, an *integral* allocation is Pareto efficient if and only if there is no feasible way to allocate to a strict superset of agents. Although quite natural, Pareto efficiency has

not been directly addressed in previous work. Pathak et al. [2021] and Delacrétaz [2021] consider a weaker *non-wastefulness* property that stipulates that in any category with unallocated quota, all eligible agents must be fully allocated. It is easy to construct settings that admit non-wasteful but Pareto inefficient allocations; for example, see the Allocation 2 in Fig. 1. Aziz and Brandl [2021] strengthen non-wastefulness to a *maximality* property: any selected valid allocation must maximize the total number of allocated units. While maximality clearly implies Pareto efficiency, we show in Proposition 1 that the two properties are in fact equivalent in this setting. We find Pareto efficiency to be more natural, both in this desideratum and when extending to settings with agent preferences. Together, our four desiderata provide a notion of a *valid* allocation:

Definition 1 (Valid Allocation). *An allocation is* valid *if it satisfies* **[QR]**, **[ER]**, **[PR]**, *and* **[PE]**.

As an illustration, Fig. 1 depicts possible allocations for the same instance: while the first violates **[PR]** and the second violates **[PE]**, the third and fourth are both valid allocations that, moreover, allocate to the same set of agents $\{a, b, c\}$. Note also that $\{a, b, c\}$ is the only set of agents who can be allocated via a valid allocation; there is no valid allocation in which agent d gets allocated. (We revisit this idea in Section 4.1.)

3 AN ALGORITHMIC CHARACTERIZATION OF VALID ALLOCATIONS

The primary concern of our work is to develop efficient algorithms that, given an instance with quotas, eligibility lists, and priorities, select a valid allocation satisfying some additional properties. To this end, we require an algorithmic way to characterize the set of valid allocations. Note that it is straightforward to find an allocation that satisfies **[QR]**, **[ER]**, and **[PR]** (and is also non-wasteful [Delacrétaz, 2021, Pathak et al., 2021]) — for example, via a round-robin policy where each category sequentially picks their top remaining agent. However, as Allocation 2 in Fig. 1 demonstrates, this may not ensure **[PE]**. On the other hand, any maximum-cardinality matching satisfies **[QR]** and **[PE]**. The challenge is to achieve all four desiderata simultaneously.

Our main result shows that there is in fact a *bijection between the set of agents selected in valid allocations, and the set of maximum weight matchings under certain valid weights.* In Section 3.1, we show that in any instance, a valid allocation can be found using a simple weighted bipartite matching LP. Subsequently, in Section 3.2, we explore some consequences of this LP formulation, including a complete characterization of all valid allocations and a discussion of their geometry; we also show that such a characterization is impossible for stable matchings.

3.1 Finding Valid Allocations via Weighted Matchings

As the basis of our formulation, we start with the following LP, which we denote by (P_0).

$$
\begin{aligned}
(P_0) \qquad & \max && V(\mathbf{x}) && \\
& \text{subject to} && \sum_{a \in \mathcal{A}} x_{a,c} \le q_c && \text{for all } \; c \in C \\
& && \sum_{c \in C} x_{a,c} \le 1 && \text{for all } \; a \in \mathcal{A} \\
& && x_{a,c} = 0 && \text{for all } \; a \in \mathcal{A}, c \in C \text{ with } a \notin \mathcal{E}_c \\
& && x_{a,c} \ge 0 && \text{for all } \; a \in \mathcal{A}, c \in C.
\end{aligned}
$$

The decision variables $\mathbf{x} = (x_{a,c})_{a \in \mathcal{A}, c \in C}$ represent the amount allocated to agent a through category c. The three sets of constraints enforce **[QR]**, the unit demand of agents, and **[ER]**, respectively. The objective, $V(\mathbf{x}) := \sum_{a \in \mathcal{A}} \sum_{c \in C} x_{a,c}$ is the total allocation of the agents; maximizing

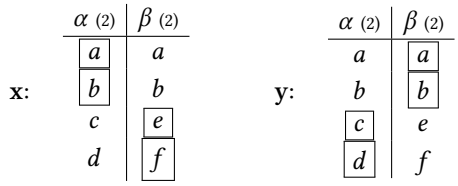

Fig. 2. *Consider the two integral allocations* $\mathbf{x}$ *and* $\mathbf{y}$ *depicted for the above allocation instance (taken from [Delacrétaz, 2021]). Both* $\mathbf{x}$ *and* $\mathbf{y}$ *are valid; however, the fractional allocation* $\mathbf{z} = \frac{1}{2}(\mathbf{x}+\mathbf{y})$ *does not respect priorities: in particular category* α *gives agent* d *an allocation* $z_{d,\alpha} = \frac{1}{2} > 0$*, but agent* $c \succeq_\alpha d$ *is not fully allocated.*

$V(\mathbf{x})$ ensures Pareto efficiency. Note that the constraints of (P_0) encode a bipartite b-matching polytope. (P_0), however, does not incorporate the category's priorities, so its solutions may not satisfy **[PR]**. Adding constraints that enforce respect for priorities appears non-trivial. In particular, the set of valid allocations is not even closed under convex combinations, as demonstrated in Fig. 2.

The critical observation is that one can perturb the coefficient of each $x_{a,c}$ in the objective to $1 - \delta_{a,c}$ in such a way as to ensure that *any optimal solution* to the perturbed LP satisfies all of the desiderata. To do so, we introduce the notion of a *valid perturbation*.

Definition 2 (Valid Perturbation). *A perturbation profile* $(\delta_{a,c})$ *is* valid *if it satisfies the following three properties:*

Positivity: $\delta_{a,c} > 0$ *for all* $a \in \mathcal{A}, c \in C$.

Small Effect: $\sum_{a\in\mathcal{A}} \sum_{c\in C} \delta_{a,c} \le \frac{1}{2}$.

Consistency: $a \succeq_c a'$ *if and only if* $\delta_{a',c} \ge \delta_{a,c}$.

Now, consider the modified objective

$$V_\delta(\mathbf{x}) := \sum_{a\in\mathcal{A}} \sum_{c\in C} x_{a,c} \cdot \left(1 - \delta_{a,c}\right) = V(\mathbf{x}) - \sum_{a\in\mathcal{A}} \sum_{c\in C} \delta_{a,c} \cdot x_{a,c}.$$

Let (P_δ) be the LP with the same constraint polytope as (P_0), but with objective $V_\delta(\mathbf{x})$. The following theorem shows that the solutions to any such perturbed LP give allocations satisfying all of our desiderata.

Theorem 1. *Let* δ *be any valid perturbation profile, and let* $\mathbf{x}^*$ *be a solution to* (P_δ)*. Then,* $\mathbf{x}^*$ *is a valid allocation (i.e., it satisfies* **[QR]**, **[ER]**, **[PR]**, *and* **[PE]***).*

Proof. The constraints immediately ensure that any feasible solution of (P_δ) satisfies **[QR]** and **[ER]**. To establish **[PR]**, let $\mathbf{x}$ be a feasible solution, a, a' be agents and c a category such that $a' \succ_c a$, $x_{a,c} = \varepsilon_1 > 0$ and $\sum_{c'} x_{a',c'} = 1 - \varepsilon_2 < 1$. Then, we can decrease $x_{a,c}$ and increase $x_{a',c}$ by $\min(\varepsilon_1, \varepsilon_2)$ without violating any constraints. Since δ is consistent, we have $\delta_{a,c} < \delta_{a',c}$, so the reassignment strictly increases the objective value. Thus, such an $\mathbf{x}$ is not optimal, and $\mathbf{x}^*$, being optimal, satisfies **[PR]**.

It remains to establish **[PE]**. Note that for any optimal solution $\hat{\mathbf{x}}$ to (P_0), we have

$$V(\mathbf{x}^*) \ge V_\delta(\mathbf{x}^*) \ge V_\delta(\hat{\mathbf{x}}) = V(\hat{\mathbf{x}}) - \sum_{a\in\mathcal{A}} \sum_{c\in C} \hat{x}_{a,c}\delta_{a,c} \ge V(\hat{\mathbf{x}}) - \sum_{a\in\mathcal{A}} \sum_{c\in C} \delta_{a,c} \ge V(\hat{\mathbf{x}}) - \tfrac{1}{2}.$$

Here, the first inequality follows since each $x^*_{a,c}, \delta_{a,c} \ge 0$. The second inequality follows since $\mathbf{x}^*$ is an optimal solution to (P_δ). The third inequality follows because the unit demand constraints ensure that each $\hat{x}_{a,c} \le 1$. Finally, the fourth inequality follows since δ has small effect.

Additionally, $\hat{\mathbf{x}}$ maximizes V among all feasible solutions to (P_0), which include $\mathbf{x}^*$. Therefore, $V(\hat{\mathbf{x}}) \geq V(\mathbf{x}^*)$. Combining both inequalities, we find that

$$V(\hat{\mathbf{x}}) \geq V(\mathbf{x}^*) \geq V(\hat{\mathbf{x}}) - \tfrac{1}{2}. \tag{1}$$

Observe that the constraint matrix of (P_0) is *totally unimodular*, as it encodes a b-matching polytope. Consequently, since all of the quotas q_c are integral, every corner point of the constraint polytope is integral. As the sum of entries $(x_{a,c})$, $V(\mathbf{x})$ is integral at corner points, and therefore at all maximizers $\mathbf{x}$ of V. In particular, since $\hat{\mathbf{x}}$ maximizes V, $V(\hat{\mathbf{x}})$ is integral. If $\mathbf{x}^*$ is a corner point, then $V(\mathbf{x}^*)$ is also integral. However, integral solutions satisfying the bounds in Eq. (1) require $V(\hat{\mathbf{x}}) = V(\mathbf{x}^*)$.

If $\mathbf{x}^*$ is not a corner point, then we write $\mathbf{x}^* = \sum_i \lambda_i \mathbf{x}^{(i)}$ as a convex combination of corner points $\mathbf{x}^{(i)}$. Because $\mathbf{x}^*$ maximizes V_δ, each of the $\mathbf{x}^{(i)}$ must also maximize V_δ. By the argument from the previous paragraph, $V(\hat{\mathbf{x}}) = V(\mathbf{x}^{(i)})$ for all i. But then, the convex combination $\mathbf{x}^*$ must also have $V(\mathbf{x}^*) = V(\hat{\mathbf{x}})$. Thus, each maximizer $\mathbf{x}^*$ of V_δ (whether or not it is a corner point) is also a maximizer of V, and hence satisfies **[PE]**. □

A surprising consequence of this result is that in any priority-respecting allocation problem, Pareto efficiency comes "for free" — the total allocation size under a valid integral allocation remains the same, irrespective of the priority orderings! The following proposition asserts that this is in fact true more generally for *any* valid allocation (integral or fractional).

Proposition 1. *Let V^* be the size of the allocation returned by (P_0) (i.e., satisfying* **[QR]**, **[ER]**, *and* **[PE]**). *Then, given any priority orders $(\succeq_c)_{c\in C}$, any valid allocation $\mathbf{x}$ has $V(\mathbf{x}) = V^*$.*

In particular, this proposition establishes that all Pareto efficient allocations are maximal. The proof of this result follows from a contrapositive argument, wherein we show that given an allocation $\mathbf{x}$ satisfying **[QR]**, **[ER]**, and **[PR]** with size less than V^*, we can construct a Pareto dominating solution. We defer the proof to Appendix B.1.

The fact that there is at least one valid allocation with size V^* was established by Aziz and Brandl [2021] based on the properties of their Reverse Rejection algorithm. Theorem 1 gives a simple way to see why this holds, and Proposition 1 shows it to be true for all valid allocations. Moreover, Theorem 1 provides a much more efficient algorithm for selecting a valid allocation: compared to Reverse Rejection, which requires one to solve $|\mathcal{A}|$ separate b-matching problems, our approach only requires solving a single weighted b-matching problem, which can be efficiently solved, for instance using the Hungarian algorithm [Ramshaw and Tarjan, 2012].

Corollary 1. *A valid allocation can be found in $O(|C|\,|\mathcal{A}|\,q + q^2 \log q)$ time.*

3.2 The Subtle Geometry of Priority-Respecting Allocations

To conclude this section, we discuss three issues related to our algorithm for locating valid allocations. First, we introduce an additional property that allows us to completely characterize the set of valid integer allocations. Next, we consider the geometry of the set of valid allocations. Finally, we consider whether an analogous LP perturbation can be used for finding stable matchings.

Characterizing all Valid Integral Allocations

By Theorem 1, we know that solving (P_δ) with any valid δ locates a valid integral allocation. A follow-up question is whether *all* valid integral allocations are solutions of (P_δ) for some choice of δ. This turns out not to be the case: for example, consider Allocations 3 and 4 in Fig. 1. While both are valid, and the former can be realized as a solution to a perturbed LP, for the latter allocation, under any valid perturbation, swapping from $x_{a,\gamma} = x_{b,\beta} = 1$ to $x_{b,\gamma} = x_{a,\beta} = 1$ (as in Allocation 3 in Fig. 1) leads to an increase in V_δ.

Fortunately, the problem illustrated by this instance is the only obstacle to realizability, as we show below. To formalize this, we introduce an additional property that we call category stability.

[CS] Category Stability: No group of categories can organize an agreeable trade through which at least one category transfers allocation to a higher-priority agent. Formally, there do not exist $j \geq 2$ categories $c_0, c_1, \ldots, c_j = c_0$ and agents $a_0, a_1, \ldots, a_j = a_0$ such that for all $0 \leq i < j$, $x_{a_i,c_i} > 0$ and $a_{i+1} \succeq_{c_i} a_i$, and at least one of the priority relations above is strict.

Note that category stability is not an added restriction on *agents* selected via valid allocations — in particular, given a valid allocation $\mathbf{x}$ that violates **[CS]**, we can modify it to get another valid allocation $\mathbf{y}$ that satisfies **[CS]** and *allocates to each agent to the same extent* (i.e., $\sum_c x_{a,c} = \sum_c y_{a,c}$ for all a). In other words, **[CS]** only discriminates among valid allocations which are equivalent in terms of the set of allocated agents. We are ready to state our main equivalence theorem.

Theorem 2. *Let* $\mathbf{x}$ *be a valid integral allocation. Then* $\mathbf{x}$ *is a solution to* (P_δ) *for some valid* δ if and only if $\mathbf{x}$ *satisfies* **[CS]**.

The forward implication (i.e., $\mathbf{x}$ is a solution to $(P_\delta) \implies \mathbf{x}$ satisfies **[CS]**) follows from a straightforward contradiction argument. For the reverse direction, we introduce the set of *serial dictatorship allocations*, which can be realized by iteratively allowing a category to allocate to its most preferred remaining agent. We argue that every valid and **[CS]** allocation can be realized through serial dictatorship, and that the serial dictatorship ordering can be used to construct a suitable valid perturbation. See Appendix B.2 for the complete proof.

Fractional Valid Allocations

Our perturbed LP procedure gives a way to locate all valid and category stable *integral* allocations, as these are corner points of our b-matching constraint polytope; however, this does not imply anything about the set of valid *fractional* allocations. Fig. 2 demonstrates that this set need not be convex; here, the convex combination of two integral valid allocations is not valid. Thus, there could exist valid fractional allocations outside the convex hull of valid integral allocations. The following proposition rules out this possibility.

Proposition 2. *Suppose that* $\mathbf{x}$ *is a valid fractional allocation. Then, we can represent* $\mathbf{x}$ *as a convex combination of valid integer allocations.*

We outline a procedure that iteratively constructs this convex combination in Appendix B.2. The result, however, provides some insight into the geometry of the set of valid allocations — it consists of a union of convex sets, each with integer corner points. The valid allocations in the example from Fig. 2 form two non-coplanar triangles with a common edge. It is an interesting open direction to further characterize the sets of valid allocations that may arise from priority-respecting allocation instances. For example, are these sets necessarily connected?

LP Perturbations and Stable Matching

As noted in Section 1, a closely related problem to priority-respecting allocation is stable matching. Both seek bipartite matchings that conform to a set of preferential constraints; moreover, the existence of solutions in both problems is implied by Scarf's Lemma (see Appendix C). Central to this discussion is the observation that unlike for priority-respecting allocation, one cannot realize stable matchings as the solutions of a perturbed b-matching polytope under a particular class of perturbations. This shows that while priority-respecting allocations and stable matchings appear syntactically similar, they have very different algorithmic properties.

4 THE COMPLEXITY OF SELECTING VALID ALLOCATIONS

In this section, we consider three possible extensions of the basic problem of selecting valid allocations. In each extension, we consider the problem of selecting from among valid allocations subject to a particular class of external objectives. Surprisingly, in each case, we show that the valid-allocation selection problem straddles the line of computational efficiency; while one given selection rule admits an efficient algorithm, a closely related selection rule is computationally hard.

4.1 Including/Excluding Agents from Valid Allocations

To formalize our first set of valid-allocation selection criteria, we first define two types of agents

Definition 3 (Unanimous/Serviceable Agents). *Given an instance $\mathcal{I} = \big(\mathcal{A}, C, (q_c), (\mathcal{E}_c), (\succeq_c)\big)$ and an agent $a \in \mathcal{A}$,*

— *Agent a is* unanimous *in $\mathcal{I}$ if it is fully allocated under* every *valid allocation* $\mathbf{x}$ *(i.e., $\sum_c x_{a,c} = 1$).*
— *Agent a is* serviceable *in $\mathcal{I}$ if there is some valid allocation* $\mathbf{x}$ *in which a is allocated (i.e., $\sum_c x_{a,c} > 0$).*

As an example, consider the instance in Fig. 1; here, one can check that agents $\{a, b, c\}$ are unanimous (and therefore serviceable), while agent d is *not* serviceable. Note that though we define serviceability in terms of non-zero allocation, as a consequence of Proposition 2, we have that any agent who can be partially allocated via a fractional valid allocation can also be fully allocated via an integral valid allocation. Hence, we can equivalently define an agent to be serviceable if it is allocated in some integral allocation.

We now show that while there is a polynomial-time algorithm to determine whether an agent is unanimous, determining whether an agent is serviceable is NP-hard. For the first claim, we establish an equivalent characterization of unanimous agents in terms of a *restricted* allocation instance.

Definition 4 (Restriction). *Given an allocation instance $\mathcal{I} = \big(\mathcal{A}, C, (q_c), (\mathcal{E}_c), (\succeq_c)\big)$ and an agent $a \in \mathcal{A}$, the* a-restriction *of c, denoted by $\mathcal{I}_{\setminus a}$, is another allocation instance with the same $\mathcal{A}$, C, and (q_c). Its eligible sets $(\mathcal{E}'_c)$ are given by*

$$\mathcal{E}'_c = \mathcal{E}_c \setminus \Big(\{a\} \cup \{a' \in \mathcal{A} \;:\; a \succ_c a'\}\Big),$$

and its priorities $(\succeq'_c)$ are the induced relations of $(\succeq_c)$ on $(\mathcal{E}'_c)$.

Given a subset $A \subseteq \mathcal{A}$, the A-restriction *of $\mathcal{I}$, denoted by $\mathcal{I}_{\setminus A}$, is defined similarly, where the eligible sets are the intersections of the eligible sets of the a-restrictions for all $a \in A$.*

Intuitively, one can think of the a-restriction as cutting each of the priority lists at agent a. Alternatively, one can view the a-restriction as the instance that would result if we committed to never allocating to a (and therefore, due to the priority constraints, never allocating from a category c to anyone ranked below a in c).

Proposition 3. *Let V^* be the value of (P_0) on instance $\mathcal{I}$. For a given agent $a \in \mathcal{A}$, let $V^*_{\setminus a}$ be the value of (P_0) on $\mathcal{I}_{\setminus a}$. Then, a is unanimous if and only if $V^* > V^*_{\setminus a}$.*

The proof of this proposition is provided in Appendix B.3. As immediate corollaries, we can derive two sufficient conditions for an agent to be unanimous.

Corollary 2. *Let V^* be the value of (P_0). Then an agent a is unanimous if either*
— *the union of all eligible agents in the a-restriction has cardinality less than V^*, or*
— *there is some category c with $a \in \mathcal{E}_c$ such that the a-restriction of c has size less than q_c.*

In other words, an agent must be allocated if they are either in the top q_c agents in any category c or alternatively if the total (over categories) number of higher-ranked agents is less than the total number of items available. On the other hand, the problem of deciding whether an agent is *serviceable* or not turns out to be NP-hard.

Proposition 4. *Given an instance $\mathcal{I}$, deciding whether an agent $a \in \mathcal{A}$ is serviceable is NP-hard.*

We show this via a reduction from the Exact Cover By 3-Sets problem or X3C [Karp, 1972]. The details of the reduction are provided in Appendix B.3.

In this context, Saban and Sethuraman [2015] study the complexity of computing selection probabilities under *random serial dictatorship* (where agents are ordered uniformly at random, and then pick their favorite remaining items in turn), and show that while one can efficiently identify items which have probability 1 of being selected by some agent, it is NP-hard to identify items which have selection probability 0. When specialized to this context, our results in this section recover and generalize this characterization.

4.2 Incorporating Agent Utilities in Selecting Valid Allocations

As a second extension, we consider how we can augment our basic model to incorporate agents' utilities for allocations from various categories. We relax our assumption of agent indifference, and equip each agent $a \in \mathcal{A}$ with a utility function $u_a : C \to (0, 1]$ that expresses the value that they derive from an allocation in each category. Thus, given an allocation $\mathbf{x}$, the *realized utility* of agent $a \in A$ is given by

$$u_a(\mathbf{x}) = \sum_{c \in C} u_a(c) \cdot x_{a,c}.$$

In this setting, the natural objective is no longer only to allocate to as many agents as possible, but rather to use the realized utility of the agents to select valid allocations. There are two potential ways to do so: First, we can select valid allocations that are Pareto efficient with respect to agent utility. Alternatively, we can attempt to optimize some aggregate *welfare* function of the agents' realized utilities. We next show that while the first goal admits an efficient algorithm, the second goal is NP-hard for most natural utility aggregation functions.

First, we consider locating an allocation that is Pareto efficient with respect to agent utilities. To this end, in this section, we denote our usual notion of *category-side* Pareto efficiency (i.e., the **[PE]** property defined in Section 2) by **[C-PE]**, and formalize Pareto efficiency from the viewpoint of agents as follows.

> **[A-PE] Agent-side Pareto Efficient:** An allocation $\mathbf{x}$ is *Pareto efficient* with respect to agent utilities if there is no allocation $\mathbf{y}$ satisfying **[QR]**, **[ER]**, and **[PR]** such that:
> — Each agent receives at least as much utility through $\mathbf{y}$: $u_a(\mathbf{y}) \geq u_a(\mathbf{x})$.
> — At least one agent receives strictly higher utility through $\mathbf{y}$ than through $\mathbf{x}$.

Intuitively, an allocation is agent-side Pareto efficient if there is no incentive for the agents to attempt to trade their allocations (from the different categories); any trade would violate one of the other constraints (**[ER]** or **[PR]**), decrease some involved agent's utility, or leave all utilities unchanged. We now argue that we can select an agent-side Pareto efficient allocation via the following two-stage algorithm (Algorithm 1). At a high level, the first stage of the algorithm determines which agents will be included in the final allocation, while the second stage maximizes the utility realized by these agents.

Algorithm 1 Valid Allocation Selection with Agent-Side Pareto Efficiency

1: **Input:** Allocation instance $\mathcal{I}$ and agent utilities $(u_a)_{a\in\mathcal{A}}$
2: Solve (P_δ) for any valid δ to pick an integral valid allocation $\mathbf{x}$
3: $A \leftarrow \left\{a \in \mathcal{A} : \sum_c x_{a,c} = 0\right\}$
4: $\left(C, \mathcal{A}, (q_c), (\mathcal{E}'_c), (\succeq'_c)\right) \leftarrow \mathcal{I}_{\setminus A}$
5: Define $U = \max_{a,c}\{u_{a,c}\}$ and $\delta'_{a,c} = \frac{U-u_a(c)}{2|\mathcal{A}|\,|C|}$ (Note: higher $u_a(c) \implies$ smaller $\delta'_{a,c}$)
6: Solve $(P_{\delta'})$ for instance $\mathcal{I}_{\setminus A}$ to locate an integral allocation $\mathbf{y}$
7: **Return:** allocation corresponding to $\mathbf{y}$

Theorem 3. *Algorithm 1 computes an allocation satisfying* **[QR]**, **[ER]**, **[PR]**, *and* **[A-PE]**.

Proof. The constraints of $(P_{\delta'})$ ensure that $\mathbf{y}$ satisfies **[QR]** and **[ER]** in $\mathcal{I}_{\setminus A}$. Moreover, since the restriction operation leaves quotas unchanged and reduces the set of eligible agents, this allocation also satisfies these desiderata in $\mathcal{I}$.

Next, note that $\mathbf{y}$ is a **[QR]**, **[ER]**, **[PR]**, and maximal allocation in $\mathcal{I}_{\setminus A}$. By construction,

$$\sum_{c\in C}\sum_{a\in\mathcal{A}} \delta'_{a,c}\cdot x_{a,c} \le \sum_{c\in C}\sum_{a\in\mathcal{E}'_c} \delta'_{a,c} \le \sum_{c\in C}\sum_{a\in\mathcal{E}'_c} \tfrac{1}{2|\mathcal{A}|\,|C|} \le \tfrac{1}{2}.$$

Therefore, the objective values satisfy

$$V(\mathbf{x}) \ge V(\mathbf{y}) \ge V_{\delta'}(\mathbf{y}) \ge V_{\delta'}(\mathbf{x}) = V(\mathbf{x}) - \sum_{c\in C}\sum_{a\in\mathcal{A}} \delta'_{a,c}\cdot x_{a,c} \ge V(\mathbf{x}) - \tfrac{1}{2}.$$

Using the fact that $\mathbf{x}$ and $\mathbf{y}$ are integral allocations, we find that $V(\mathbf{x}) = V(\mathbf{y})$. Thus, $\mathbf{y}$ allocates to all agents in the restricted instance, so it satisfies **[PR]**; the definition of the restriction $\mathcal{I}_{\setminus A}$ ensures that no unallocated agent in A has priority over an agent allocated in $\mathbf{y}$.

It remains to argue that $\mathbf{y}$ is agent-side Pareto efficient. The perturbations $\delta'_{a,c}$ are monotone decreasing in the utilities $u_a(c)$. Therefore, $\mathbf{y}$ maximizes the total utility of the allocated agents within $\mathcal{I}_{\setminus A}$. Therefore, any alternate allocation in which one agent realizes a higher utility must also include an agent who realizes a lower utility, so the allocation given by $\mathbf{y}$ satisfies **[A-PE]**. □

Note that Algorithm 1 does not admit an analogous realizability result to Theorem 2. As argued above, the computation of $\mathbf{x}$ in the first stage ensures that the final allocation is maximal. However, not every utility Pareto efficient allocation is maximal, as demonstrated by the example in Fig. 3.

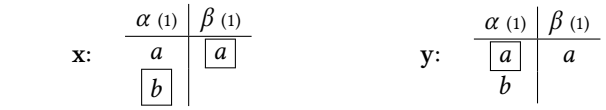

Fig. 3. *Consider the above instance with $u_{a,\alpha} = 1$, $u_{a,\beta} = u_{b,\alpha} = \frac{1}{3}$. The maximal (so* **[C-PE]***) allocation shown on the left is* **[A-PE]***, and will be output by Algorithm 1. However, the allocation shown on the right is also* **[A-PE]***, and moreover utility-maximizing. However, as this allocation is not* **[C-PE]***, it cannot be realized by Algorithm 1.*

Next, we turn our attention to the hardness of maximizing aggregate functions of the agents' realized utilities. Our main result in this setting is captured by the following theorem:

Theorem 4. *Let $(F_n : [0,1]^n \to \mathbb{R})_{n=1}^{\infty}$ be a family of aggregation functions that are all continuous and strictly increasing in each of their arguments. Then, the following problem is NP-hard: Given an*

allocation instance $\mathcal{I}$ with $\mathcal{A} = \{a_1, \ldots, a_n\}$, select a valid allocation maximizing the aggregate agent utility under F_n, i.e., find

$$\mathbf{x}^* \in \operatorname*{argmax}_{\mathbf{x} \text{ valid}} \left\{ F_n\Big(u_{a_1}(\mathbf{x}) \ , \ \ldots \ , \ u_{a_n}(\mathbf{x})\Big)\right\}.$$

Proof. We restrict attention to utilities of the following form:

$$u_{a_1}(c) = 1 \quad \text{for all } c \in C$$
$$u_{a_j}(c) = u \quad \text{for all } c \in C \qquad \text{for all } j \geq 2$$

for some $u \in (0, 1]$; in other words, all agents are indifferent regarding which category they are allocated through, and there is (weakly) higher utility for allocating agent a_1. We write $t_{\mathbf{x}} \in [0, 1]^{|\mathcal{A}|}$ for the vector of total agent allocations with $(t_{\mathbf{x}})_j = \sum_c x_{a_j,c}$. We then define

$$f : (0, 1] \times [0, 1]^n \to \mathbb{R} \qquad f\,(u, t_{\mathbf{x}}) = F_n\left((t_{\mathbf{x}})_1, u \cdot (t_{\mathbf{x}})_2, \ldots, u \cdot (t_{\mathbf{x}})_n\right)$$

as the aggregate agent utility for a given parameter u and agent allocation $\mathbf{x}$. Inheriting properties of F_n, f is continuous and strictly increasing in u, and strictly increasing in each $(t_{\mathbf{x}})_j$ when $u > 0$.

Now, suppose that there is some valid allocation $\mathbf{x}$ with $(t_{\mathbf{x}})_1 = \tau > 0$ (i.e., $\mathbf{x}$ allocates to a_1) and another valid allocation $\mathbf{y}$ with $(t_{\mathbf{y}})_1 = 0$ (i.e., $\mathbf{y}$ does not allocate to a_1). Let $\mathbf{e}_1$ denote the first standard basis vector and $\mathbf{1}$ denote the all ones vector, both in $\mathbb{R}^{|\mathcal{A}|}$. Since $f(0, \tau \cdot \mathbf{e}_1) > f(0, \mathbf{0}) = f(0, \mathbf{1} - \mathbf{e}_1)$ and f is continuous in its first argument, then we can choose some sufficiently small $\varepsilon > 0$ such that

$$f(\varepsilon, \tau \cdot \mathbf{e}_1) > f(\varepsilon, \mathbf{1} - \mathbf{e}_1).$$

Since f is strictly increasing in each agent's total allocation, the two allocations $\mathbf{x}, \mathbf{y}$ have aggregate utilities

$$f(\varepsilon, t_{\mathbf{y}}) \leq f(\varepsilon, \mathbf{1} - \mathbf{e}_1) < f(\varepsilon, \tau \cdot \mathbf{e}_1) \leq f(\varepsilon, t_{\mathbf{x}}).$$

Therefore, we can reduce the problem of deciding whether a_1 is serviceable to determining whether the F_n-maximizing valid allocation $\mathbf{x}^*$ (under the utilities defined above) has value greater than $f(\varepsilon, \mathbf{1} - \mathbf{e}_1)$. From Proposition 4 we know that checking whether an agent is serviceable is NP-hard — hence, so is the problem of selecting a valid allocation that maximizes aggregate utility. □

Note that this theorem relies on the fact that agents have *cardinal* utilities for categories. We can use this theorem to conclude that many natural welfare optimization problems are computationally hard in the reserve allocation setting. We record two such results below.

Corollary 3. *Given an instance $\mathcal{I}$ equipped with a utility function u_a for each agent $a \in \mathcal{A}$, it is NP-hard to find a valid allocation $\mathbf{x}^*$ that maximizes total agent utility*

$$\sum_{a \in \mathcal{A}} \sum_{c \in C} u_a(c) \cdot x_{a,c}.$$

This corollary provides a stark contrast to our original setting (concerned with the number of allocated agents), where maximizing total allocation and ensuring Pareto efficiency were equivalent (see Proposition 1).

Corollary 4. *Given an instance $\mathcal{I}$ equipped with a utility function u_a for each agent $a \in \mathcal{A}$, it is NP-hard to find a valid allocation $\mathbf{x}^*$ that maximizes Nash social welfare*

$$\mathrm{NSW}(\mathbf{x}) := \left(\prod_{a \in \mathcal{A}} \sum_{c \in C} u_a(c) \cdot x_{a,c} \right)^{\frac{1}{|\mathcal{A}|}}.$$

4.3 Auditing Valid Allocations via Optimizing Cutoffs

Thus far, we have considered the quality of allocations only through the formal desiderata that we have introduced. While theoretically satisfying, such an approach fails to acknowledge their impact on agents affected by these algorithms in practice. How can we convince the recipients (or, more aptly, non-recipients) of medical care, school seats, or other resources that decisions have been made fairly? This is discussed in great detail by Pathak et al. [2021], who suggest that one way addressing this issue is via the notion of *auditability*: revealing extra information to agents to help satisfy them that their allocation is appropriate. In particular, a natural way to audit allocations is by revealing allocation *thresholds* (or *cutoff vectors* Pathak et al. [2021]) in each category. In this section, we study how to select valid allocations to optimize some metric related to these thresholds.

For notational ease, throughout this case study, we restrict our attention to integral allocations , realized as maps $\varphi : \mathcal{A} \to C \cup \{\perp\}$ (where $\varphi(a) = c$ if and only if $x_{a,c} = 1$, and $\varphi(a) = \perp$ corresponds to a being unallocated, i.e., $\sum_c x_{a,c} = 0$.). This is natural for defining cutoffs, and also is without loss of generality since our approach in Theorem 1 naturally returns integral allocations.

Definition 5 (Allocation Thresholds). *Thresholds $\theta : C \to \mathbb{N}$ corresponding to allocation φ satisfy:*

- *Every agent allocated in category $c \in C$ has rank equal to or less than c's threshold, i.e.*
$$\varphi(a) = c \quad \Longrightarrow \quad r_c(a) \le \theta(c) \qquad \text{for all } a \in \mathcal{A}.$$
- *Every unallocated agent has rank equal to or greater than the threshold in each eligible category*
$$\varphi(a) = \perp \text{ and } a \in \mathcal{E}_c \quad \Longrightarrow \quad r_c(a) \ge \theta(c) \qquad \text{for all } a \in \mathcal{A}.$$

There are two natural thresholds associated with any allocation φ (see Fig. 4 for a visualization):

- The **inner threshold** of φ, denoted by $\underline{\theta}$, has $\underline{\theta}(c) = \max\{r_c(a) : \varphi(a) = c\}$, the *maximum rank over all agents allocated* in each category.
- The **outer threshold** of φ, denoted by $\overline{\theta}$, has $\overline{\theta}(c) = \min\{r_c(a) : \varphi(a) = \perp, a \in \mathcal{E}_c\}$, the *minimum rank over all unallocated eligible agents* in each category. If all agents in category c are allocated, we set $\overline{\theta}(c)$ equal to one more than the maximum eligible rank in the category.

α (3)	β (2)	γ (2)
$\boxed{a_1}$	$\boxed{a_5}$	$\boxed{a_6}$
$\boxed{a_2}$	$\boxed{a_3}$, a_8	a_3
a_3	a_4	a_1
$\boxed{a_4}$	a_0 , a_9	$\boxed{a_8}$
a_9	a_1	a_7
a_8		a_0

Fig. 4. *In this allocation instance (where the boxed agents form a valid allocation), the inner threshold $\underline{\theta} = (4, 2, 4)$ corresponds to the rank of the highest red-shaded tier in each column; all allocated agents occur at that priority level or higher. The lowest red-shaded tier corresponds to the outer threshold $\overline{\theta} = (5, 4, 5)$; the three unallocated agents (a_0, a_7 and a_9) are at or below this level in each category. Any mapping from the categories to one of the red-shaded tiers gives a valid threshold function.*

Auditing Allocated Agents by Optimizing Inner Thresholds: One way to audit a valid allocation is by the quality of *allocated* agents. Allocations with large inner threshold are the "most" respectful of priorities in the sense that each category allocates only to agents in high priority tiers. There are two natural ways to quantify this: we can minimize the *sum* of ranks of allocated agents, or we can minimize the maximum rank of an allocated agent. Both of these objectives are handled by our approach by carefully choosing the valid perturbation δ (proofs provided in Appendix B.4.).

Proposition 5. *Given an instance $\mathcal{I}$, define perturbations $\delta_{a,c} = \frac{r_c(a)}{2|C|\,|\mathcal{A}|^2}$. Then any (integral) allocation $\mathbf{x}$ returned by (P_δ) is a valid allocation that minimizes the sum of allocated agents' ranks.*

Proposition 6. *Given an instance $\mathcal{I}$, define perturbations $\delta_{a,c} = \frac{1}{2|C|\,|\mathcal{A}|} \cdot \left(\frac{1}{|\mathcal{A}|+1}\right)^{|\mathcal{A}|-r_c(a)}$. Then any (integral) allocation $\mathbf{x}$ returned by (P_δ) is a valid allocation that minimizes the maximum rank over all allocated agents (i.e., maximum inner threshold over all categories).*

Auditing Unallocated Agents by Optimizing Outer Thresholds: Suppose instead that from the perspective of categories, what matters is that highly-ranked agents are allocated from *some* category. A natural way to audit this is via the *outer threshold*, which marks the rank of the first unallocated agent in a category; one may thus want to select valid allocations that have larger values for these outer thresholds. Again, there are two natural realizations of this objective: we can maximize the *minimum* outer threshold, or the *sum* of the outer thresholds over categories. Unlike the inner threshold, however, optimizing both of these objectives is NP-hard.

Proposition 7. *Given an instance $\mathcal{I}$, selecting a valid allocation φ^* that maximizes the minimum over all categories of the outer threshold is NP-hard.*

Proposition 8. *Given an instance $\mathcal{I}$, selecting a valid allocation φ^* that maximizes the sum over all categories of the outer threshold is NP-hard.*

The proofs for the above results are based on a reduction from X3C in a similar vein as the proof of Proposition 4; for details, refer to Appendix B.4. More surprisingly, there is a sense in which the second objective is strictly harder: suppose we *de-reserve* units from the categories by removing the quota constraints, and instead impose a single global constraint that the total number of allocations across all categories is at most q. Now, maximizing the first objective becomes trivial (one can iteratively assign to the highest-ranked unallocated agent over all categories), but the objective of maximizing the sum of outer thresholds remains hard.

Proposition 9. *Given an instance $\mathcal{I}$, selecting an allocation φ^* giving to at most q agents that maximizes the sum over all categories of the outer threshold is NP-hard.*

5 ONLINE PRIORITY-RESPECTING ALLOCATION

The second broad application we consider is allocating resources to agents who arrive *online*, while still respecting priority and quota considerations. Our results here again critically depend on the equivalence between valid matchings and perturbed maximum-weight matchings, demonstrating the importance of our characterization in Theorem 1.

5.1 Online Allocation with Priorities: Preliminaries

Our model is as follows: Agents arrive one at a time over T rounds $t = 1, \ldots, T$; we refer to the agent arriving in round t simply as *agent* t. Each arriving agent has an observable *type* $\theta[t] \in \Theta$; here, Θ is a discrete and typically small set. For example, each category could give each agent a *priority level* in $\{1, 2, \ldots, \ell, \text{ineligible}\}$ for some small ℓ; in this case, an agent's type is their vector of priority levels. Categories now have eligibility criteria and priorities over these types; that is, the eligible set is $\mathcal{E}_c \subseteq \Theta$, and the total pre-order $\preceq_c$ is defined over $\mathcal{E}_c$. By distinguishing between agents and their types, our model allows us to separate out two parameters: the number of types (which is typically small), and the number of agents (which may be large). Indeed, our main goal is to achieve online algorithms whose costs can be bounded in terms of the "small" parameters (number of types and number of categories), independent of the total number of agents T.

In each round t, the type $\theta[t]$ of the arriving agent is drawn randomly from some known probability distribution; for simplicity[3], we assume that $\theta[t] = \theta$ i.i.d. with probability p_θ. We use $\mathbf{p} = (p_\theta)_{\theta \in \Theta}$ for the vector of all these probabilities. Under this arrival model, the number of agents $(N_\theta)_{\theta \in \Theta}$ of each type on a given sample path follows a Multinomial$(T, \mathbf{p})$ distribution.

After observing the type $\theta[t]$ of agent t, the principal must irrevocably decide to either allocate a reserved unit from one of the categories to agent t or leave t unallocated forever. Given the online

[3]Under suitable technical assumptions, our results can be generalized to non-stationary arrival probabilities. However, the added notational overhead outweighs the mild added generalization.

nature of the problem and uncertainty due to randomness, it is impossible to satisfy all of the axioms we considered earlier; Pareto efficiency stands in obvious conflict with respecting priorities. To see this, notice that when an algorithm early on considers allocating to an agent of a particular type, there are two possible extreme scenarios that could occur with positive probability: if all subsequent agents have lower priority, then not allocating to the agent may result in a drastic loss in efficiency. Conversely, if all subsequent agents have higher priority, then allocating to the agent would deprive one of the future agents of an allocation, violating priorities. Thus, it is important to decide how to quantitatively trade off the violated axioms.

One natural option is to treat the priorities as a hard constraint, and maximize the expected number of allocations subject to this constraint. Doing so leads to a straightforward MDP; unfortunately, treating priorities as a hard constraint can lead to very poor performance.

Proposition 10. *Even with a single category and three priority levels (types), there exist instances in which any online allocation algorithm guaranteeing no priority violations must incur $\Omega(T)$ efficiency loss in hindsight, with all but at most exponentially small probability.*

While we defer the formal proof to Appendix B.5, the intuition behind Proposition 10 is simple. Consider a single category with quota $q = \frac{T}{2}$ and three eligible types: $a \succ b \succ c$ with $p_a = p_b = p_c = \frac{1}{3}$. By the law of large numbers, the optimal allocation in hindsight accepts roughly half of the arriving type-b agents (and no type-c agent). However, to deterministically guarantee no priority violations, an algorithm is forced to allocate only to type-a agents until it can be sure that there is room to accommodate *all* potential future type-a agents. However, at that point, the number of type-b agents who can be allocated is $\Omega(T)$ less than the optimal solution with hindsight.

In light of Proposition 10, it is necessary to relax the **[PR]** axiom to achieve meaningful guarantees. We therefore consider the tradeoff between the following two metrics:

> **Efficiency loss** Δ_e: The difference between the maximum cardinality of any allocation and the number of allocations made by the algorithm.
> **Priority loss** Δ_p: The number of unallocated agents with some type θ eligible in some category c that allocated one or more slots to lower-priority agents (i.e., with type $\theta' \prec_c \theta$).

We henceforth refer to unallocated agents contributing to the priority loss as *priority violations*. Note that both Δ_e and Δ_p are random variables, computed in hindsight on each sample path. Moreover, the optimal *offline* (i.e., hindsight) allocation simultaneously makes both losses 0. Our goal is to understand how online algorithms can trade off between these losses.

5.2 Efficiency-Priority Tradeoffs for Online Allocation

We now present our main result in this section: we design an online allocation policy that guarantees that *the sum of the efficiency loss and priority loss is independent of T and q* (i.e., of the number of agents/allocations). Formally, we have the following guarantee.

Theorem 5. *Let $p_{min} = \min_{\theta \in \Theta} p_\theta$. For any valid δ, the allocation returned by the* Online Priority-Respecting Allocation with Restrictions *Policy (Algorithm 2) satisfies*

$$\mathbb{E}[\Delta_e + \Delta_p] \leq \frac{|\Theta|^5 (|C|+1)^4}{p_{min}^4}.$$

The dependence of this bound on each of these three parameters (Θ, C, and $p_{\min}$) is unavoidable. Note that given any problem instance, we can duplicate each category with distinct types in each copy, leading to at least linear dependence on $|C|$ and $|\Theta|$. In addition, note that the problem of

selecting the top k elements of a random stream is a special case of our setting. For this problem, it is known that linear dependence on $\frac{1}{p_{\min}}$ is unavoidable (see Figure 1 in [Arlotto and Gurvich, 2019]). Getting the optimal dependence on $|\Theta|, p_{\min}$ and $|C|$ is left open for future work.

The central idea behind our algorithm is to solve the perturbed LP on the expected number of future arrivals and use the solution to select an action that is least likely to cause priority violations or efficiency loss. The guarantee follows by using the compensated coupling technique of Vera and Banerjee [2021] (see also Banerjee and Freund [2020]), which essentially allows us to leverage smoothness properties of linear programs to obtain sample-path regret bounds. Our characterization in Theorem 1 is essential for using this approach. We note also that since our objective (in particular, Δ_p) has a Lipschitz constant that grows with T, we cannot directly adopt existing uniform-regret results [Banerjee and Freund, 2020]; rather, we must carefully use *restrictions* (Definition 4) to control the Lipschitz constant and obtain our results.

Our algorithm uses as a subroutine the following **Interim LP relaxation** $P_\delta\Big(t, \mathbf{N}[t], \mathcal{E}[t], \mathbf{q}[t]\Big)$:

$$
\begin{aligned}
\max \quad & \sum_{c\in C}\sum_{\theta\in\Theta} x_{\theta,c}[t]\cdot(1-\delta_{\theta,c}) \\
\text{subject to} \quad & x_{\theta,\perp}[t] + \sum_{c\in C} x_{\theta,c}[t] = N_\theta[t] && \text{for all } \theta\in\Theta \\
& \sum_{\theta\in\Theta} x_{\theta,c}[t] \le q_c[t] && \text{for all } c\in C \\
& x_{\theta,c}[t] = 0 && \text{for all } c\in C, \theta\notin\mathcal{E}_c[t] \\
& x_{\theta,c}[t] \ge 0 && \text{for all } c\in C\cup\{\perp\}, \theta\in\Theta
\end{aligned}
$$

The interim LP can be viewed as a proxy solution to the perturbed LP (P_δ) in Theorem 1, *given past allocation decisions*. t indexes the current arrival, the parameters $\mathbf{N}[t] = (N_\theta[t])_{\theta\in\Theta}$ represent the number of *future* arrivals of each type $\theta\in\Theta$ over rounds $t,\dots,T$, the parameters $\mathcal{E}[t] = (\mathcal{E}_c[t])_{c\in C}$ represent the restricted eligibility sets (see Algorithm 2) at time t, and the parameters $\mathbf{q}[t] = (q_c[t])_{c\in C}$ represent the available quotas at the start of round t. The decision variables $\mathbf{x}[t] = (x_{\theta,c}[t])_{c\in C\cup\{\perp\},\theta\in\Theta}$ represent the number of agents of type θ who will be allocated in category c (or remain unallocated, for $c=\perp$) from among the arrivals $t,\dots,T$. The objective function, as before, accrues one unit for each allocated agent minus some chosen perturbation $\delta_{\theta,c}$. The first constraint accounts for future arrivals of type θ; the second ensures that the combination of past and future allocations does not exceed the reserved quota for any category; the third ensures that the solution respects eligibility. Note that the interim LP does not ensure respect for priorities, as it does not account for which agent types were allocated in the past. In fact, as shown in Proposition 4, it is NP-hard to compute whether there is a valid allocation that includes these agents. Given this LP family, we are ready to state our algorithm.

For each arriving agent t, the algorithm solves the LP using its current quotas $\mathbf{q}[t]$ and eligible sets $\mathcal{E}[t]$, the current arrival $\theta[t]$, and the expected number of future arrivals of each type. It allocates to agent t through a category (including the "no allocation" category $\perp$) maximizing the expected allocation under the optimal LP solution. When an agent is not allocated, the algorithm takes a restriction of the allocation instance to prevent future priority violations.

Proof of Theorem 5. To bound the expected loss of Algorithm 2, we use a variant of the compensated coupling argument of Vera and Banerjee [2021]. In each round t, we consider two random variables, with the randomness taken over the future arrivals $\theta[t+1],\dots,\theta[T]$.

Algorithm 2 Online Priority-Respecting Allocation with Restrictions

Input: Allocation Instance $\left(C, \Theta, \mathbf{q}, \mathcal{E}, (\preceq_c)_{c\in C}, \mathbf{p}\right)$, Online Arrivals $(\theta[t])_{t\in[T]}$
Output: Allocations $(y[t])_{t\in[T]}$, $y[t] \in C \cup \{\perp\}$

Select a valid perturbation δ; Initialize $\mathcal{E}[1] \leftarrow \mathcal{E}$, $\mathbf{q}[1] \leftarrow \mathbf{q}$
for each $t = 1, \ldots, T$ **do**
  $\mathbf{x}^*[t] \leftarrow$ solution to $P_\delta\Big(t, \left(\mathbb{1}(\theta = \theta[t]) + (T-t)\cdot p_\theta\right)_{\theta\in\Theta}, \mathcal{E}[t], \mathbf{q}[t]\Big)$
  $y[t] \leftarrow \underset{c\in C\cup\{\perp\}}{\operatorname{argmax}} \left(x^*_{\theta[t],c}[t]\right)$, $q_c[t+1] \leftarrow q_c[t] - \mathbb{1}(y[t] = c)$ for each $c \in C$
  if $y[t] = \perp$ **then**
    $\mathcal{E}_c[t+1] \leftarrow \mathcal{E}_c[t] \setminus \left(\{\theta[t]\} \cup \{\theta \in \Theta\colon \theta \prec_c \theta[t]\}\right)$ for each $c \in C$
  else
    $\mathcal{E}_c[t+1] \leftarrow \mathcal{E}_c[t]$ for each $c \in C$

— $\Delta_e[t]$ represents the *efficiency* loss due to the algorithm's decision at time t. Using our notation,

$$\Delta_e[t] = \underbrace{P_0\Big(t, \mathbf{N}[t], \mathcal{E}[t], \mathbf{q}[t]\Big)}_{\substack{\text{Optimal offline allocation given}\\ \text{decisions made before round } t}} - \Big(\mathbb{1}(y[t] \neq \perp) + \underbrace{P_0\Big(t+1, \mathbf{N}[t+1], \mathcal{E}[t+1], \mathbf{q}[t+1]\Big)}_{\substack{\text{Optimal offline allocation given}\\ \text{decisions made through round } t}}\Big).$$

— $\Delta_p[t]$ represents the *priority* loss due to the algorithm's decision at time t, i.e., the number of additional unallocated and envious agents that arise as a result of the allocation of $\theta[t]$.

We denote the value of the decision variables at an optimum of the *offline* LP at time t by $\mathbf{x}^*[t]$. We separately reason about these two sources of loss in two cases: when agent t is allocated through some category, vs. when t remains unallocated.

If agent t is allocated, then $y[t] = c$ for some $c \in C$. If $x^*_{\theta[t],c}[t] > 0$, then the optimal solution along this sample path allocates to an agent of type $\theta[t]$ in category c. Thus, the allocation to agent t has not deviated from this optimal allocation, so no loss needs to be compensated for. If $x^*_{\theta[t],c}[t] = 0$, meaning that the optimal solution does not allocate to any agents of type $\theta[t]$ from time t onwards, the algorithm's choice of allocation may reduce the efficiency by at most one. This is because the optimal solution can introduce at most one augmenting path into the bipartite allocation graph. In addition to the efficiency loss, the allocation to t may prevent some agents with higher priority in c from receiving an allocation, leading to priority violations. A crude upper bound on the increase in the number of priority violations is $T - t$, i.e., all remaining agents. Hence, we obtain the upper bound $\Delta_e[t] + \Delta_p[t] \leq \mathbb{1}(x^*_{\theta[t],c}[t] = 0)\cdot(1 + T - t)$.

Next, we consider the case in which agent $\theta[t]$ remains unallocated, so $y[t] = \perp$. Again, if $x^*_{\theta[t],\perp}[t] > 0$, i.e., the optimum solution also leaves at least one agent of type $\theta[t]$ unallocated, the failure to allocate to agent t does not cause any loss in efficiency or priority. Therefore, we assume that $x^*_{\theta[t],\perp}[t] = 0$. The non-allocation to $\theta[t]$ causes the algorithm to restrict the allocation instance: in the future, it will never be able to allocate to agents whose types have lower priority than $\theta[t]$. Even so, the efficiency loss can be safely upper-bounded by $T - t + 1$, i.e., all agents after and including agent $\theta[t]$. In addition to the efficiency loss, the failure to allocate to agent t may lead to a priority violation at the expense of t; however, this can be the only resulting priority violation. Thus, we obtain the upper bound $\Delta_e[t] + \Delta_p[t] \leq \mathbb{1}(x^*_{\theta[t],\perp}[t] = 0)\cdot(T - t + 1 + 1)$.

Combining the above, we get that the sum of the losses in round t can be upper-bounded as

$$\Delta_e[t] + \Delta_p[t] \le \mathbb{1}(x^*_{\theta[t],y[t]}[t] = 0) \cdot (T - t + 2),$$

and summing over all rounds, and taking expectations, we get

$$\mathbb{E}[\Delta_e + \Delta_p] \le \sum_{t=1}^{T} \mathbb{P}[x^*_{\theta[t],y[t]}[t] = 0] \cdot (T - t + 2). \tag{2}$$

Next, we establish a bound on the probability $\mathbb{P}[x^*_{\theta[t],y[t]}[t] = 0]$. Recall that in each round, $y[t]$ is selected as an argmax over $c \in C \cup \{\perp\}$ of $x_{\theta,c}[t]$. That is, $y[t]$ is a most frequent assignment of the future arriving agents of type $\theta[t]$ when the expected number of agents of each type arrive. In expectation, the number of arrivals of type $\theta[t]$ in rounds $t, \dots, T$ is $1 + (T - t) \cdot p_{\theta[t]}$. Thus, $x_{\theta[t],y[t]} \ge \frac{1+(T-t)p_{\theta[t]}}{|C|+1}$; this implies a lower bound on the infinity norm of the difference between the LP solution $\mathbf{x}[t]$ and the optimal offline solution $\mathbf{x}^*[t]$, i.e., $\left\|\mathbf{x}[t] - \mathbf{x}^*[t]\right\|_\infty \ge \frac{1+(T-t)\cdot p_{\theta[t]}}{|C|+1}$.

On the other hand, using the $(1, \infty)$-Lipschitz property of maximum-weight matchings with respect to budgets [Vera and Banerjee, 2021, Proposition 4], we have that

$$\left\|\mathbf{x}^*[t] - \mathbf{x}[t]\right\|_\infty \le \left\|\mathbf{N}[t+1] - (T - t - 1) \cdot \mathbf{p}\right\|_1.$$

Thus, the (bad) event $x^*_{y[t],\theta[t]}[t] = 0$ implies that $\left\|\mathbf{N}[t+1] - (T-t-1)\cdot\mathbf{p}\right\|_1 \ge \frac{1+(T-t)\cdot p_{\theta[t]}}{|C|+1}$, i.e., that the actual type counts differ a lot from their expectations. A large deviation of the $\|\cdot\|_1$-norm implies that at least one coordinate must differ by at least the average, so this event implies that the actual number of arrivals for at least one type differs from its expectation by at least an additive $\frac{1+(T-t)\cdot p_{\theta[t]}}{|\Theta|\cdot(|C|+1)}$. Because arrival counts for any type θ follow the distribution $N_\theta[t] \sim \text{Binomial}(T - t, p_\theta)$, the Hoeffding bound gives us that the probability of a large deviation for any one type θ is

$$\mathbb{P}\left[\left|N_\theta[t+1] - \mathbb{E}\left[N_\theta[t+1]\right]\right| \ge \frac{1+(T-t)\cdot p_\theta}{|\Theta|\cdot(|C|+1)}\right] \le 2 \cdot \exp\left(\frac{-2(T-t)\cdot p_\theta^2}{|\Theta|^2\cdot(|C|+1)^2}\right) \le 2e^{-\kappa(T-t)},$$

where $\kappa = \frac{2p_{\min}^2}{|\Theta|^2(|C|+1)^2} \le \frac{1}{2}$. Taking a union bound over the $|\Theta|$ types θ and substituting the resulting upper bound into (2), the expected loss is upper-bounded by

$$\begin{aligned}\mathbb{E}[\Delta_e + \Delta_p] &\le \sum_{t=1}^{T} (T - t + 2) \cdot 2|\Theta| \cdot e^{-\kappa\cdot(T-t)} \\ &\le 2|\Theta| \cdot \int_0^\infty (z + 2) \cdot e^{-\kappa z}\, dz \;=\; |\Theta| \cdot \tfrac{4\kappa+2}{\kappa^2} \overset{\kappa\le\frac{1}{2}}{\le} |\Theta| \cdot \tfrac{4}{\kappa^2} \;=\; \tfrac{|\Theta|^5(|C|+1)^4}{{p_{\min}}^4}.\end{aligned}$$

□

6 CONCLUSION

We studied allocation settings where units of some public resource are to be divided between multiple categories, each with a quota of items, and a priority ordering over eligible agents. The goal is to find a *valid allocation* — one which respects the quotas, eligibility, and priority requirements, while still being Pareto optimal. Our main result demonstrates a bijection between valid integral allocations and maximum-weight matchings under a set of *valid weights*. This approach allowed us to efficiently locate and select valid allocations, despite the set of valid allocations being non-convex. On the other hand, our hardness results demonstrate the strange geometry of this set, due to which optimizing over it remains challenging. We hope our work can help guide the use of priorities and quotas in a wide variety of settings. Extending our approach to models involving two-sided preferences and/or complementarities provide interesting avenues for future research.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support from AFOSR grant FA9550-23-1-0068, ARO MURI grant W911NF-19-1-0217, NSF grants ECCS-1847393 and CNS-195599, and the Simons Institute for the Theory of Computing. The authors also thank Oktay Günlük, Karola Mészáros, Rakesh Vohra, and the participants at the 2022 ACM Symposium on Foundations of Responsible Computing (FORC) for useful comments that helped shape this paper.

REFERENCES

A. Abdulkadiroğlu and A. Grigoryan. Priority-based assignment with reserves and quotas. *NBER Tech. Rep.*, 2021.

A. Abdulkadiroğlu, P. A. Pathak, A. E. Roth, and T. Sönmez. The Boston public school match. *American Economic Review*, 95 (2):368–371, 2005.

E. E. Andrews, K. B. Ayers, K. S. Brown, D. S. Dunn, and C. R. Pilarski. No body is expendable: Medical rationing and disability justice during the Covid-19 pandemic. *American Psychologist*, 76(3):451, 2021.

A. Arlotto and I. Gurvich. Uniformly bounded regret in the multisecretary problem. *Stochastic Systems*, 9(3):231–260, 2019.

H. Aziz and F. Brandl. Efficient, fair, and incentive-compatible healthcare rationing. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 103–104, 2021.

H. Aziz and Z. Sun. Multi-rank smart reserves. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 105–124, 2021.

S. Banerjee and D. Freund. Uniform loss algorithms for online stochastic decision-making with applications to bin packing. In *ACM SIGMETRICS'20*, 2020.

N. Benabbou, M. Chakraborty, X.-V. Ho, J. Sliwinski, and Y. Zick. Diversity constraints in public housing allocation. In *17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*, 2018.

N. Benabbou, M. Chakraborty, and Y. Zick. Fairness and diversity in public resource allocation problems. *Bulletin of the Technical Committee on Data Engineering*, 2019.

C. E. Binkley and D. S. Kemp. Ethical rationing of personal protective equipment to minimize moral residue during the Covid-19 pandemic. *Journal of the American College of Surgeons*, 230(6):1111–1113, 2020.

P. Biró and J. Gudmundsson. Complexity of finding Pareto-efficient allocations of highest welfare. *European Journal of Operational Research*, 291(2):614–628, 2021.

J. Correa, N. Epstein, R. Epstein, J. Escobar, I. Rios, N. Aramayo, B. Bahamondes, C. Bonet, M. Castillo, A. Cristi, et al. School choice in Chile. *Operations Research*, 2021.

COVAX. Covax explained. https://www.gavi.org/vaccineswork/covax-explained, 2020. Accessed: 2022-02-14.

D. Delacrétaz. Processing reserves simultaneously. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 345–346, 2021.

E. J. Emanuel, G. Persad, R. Upshur, B. Thome, M. Parker, A. Glickman, C. Zhang, C. Boyle, M. Smith, and J. P. Phillips. Fair allocation of scarce medical resources in the time of Covid-19. *New England Journal of Medicine*, 382(21):2049–2055, 2020.

A. Erdil and H. Ergin. Two-sided matching with indifferences. *Journal of Economic Theory*, 171:268–292, 2017.

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

R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Springer, 1972.

S. Kintali. Complexity of scarf's lemma and related problems. *arXiv preprint arXiv:0812.1601*, 2008.

T. Nguyen and R. Vohra. Complementarities and externalities. In *Online and Matching-Based Market Design*. Cambridge University Press, 2022.

P. A. Pathak, T. Sönmez, M. U. Ünver, and M. B. Yenmez. Fair allocation of vaccines, ventilators and antiviral treatments: leaving no ethical value behind in health care rationing. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 785–786, 2021.

L. Ramshaw and R. E. Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. *HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1*, 2012.

D. Saban and J. Sethuraman. The complexity of computing the random priority allocation matrix. *Mathematics of Operations Research*, 40(4):1005–1014, 2015.

H. E. Scarf. The core of an n person game. *Econometrica: Journal of the Econometric Society*, pages 50–69, 1967.

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

A. Vera and S. Banerjee. The bayesian prophet: A low-regret framework for online decision making. *Management Science*, 67(3):1368–1391, 2021.

D. B. White and B. Lo. A framework for rationing ventilators and critical care beds during the Covid-19 pandemic. *Journal of the American Medical Association*, 323(18):1773–1774, 2020.

A RELATED WORK

As mentioned, we build on the framework of Pathak et al. [2021], which has inspired several follow-up papers. Delacrétaz [2021] notes that since the axioms do not uniquely identify an allocation, different choices can induce biases; to allay this, he introduces a waterfilling-style *simultaneous allocation* procedure that leads to a unique (fractional) outcome. On the other hand, Aziz and Brandl [2021] introduce a procedure that results in a maximum-size allocation. Aziz and Sun [2021] describe how to incorporate diversity goals as an optimization objective in these settings. Finally, Abdulkadiroğlu and Grigoryan [2021] consider lower bounds on categories, and develop a choice rule that minimizes the number of priority violations in this setting.

A closely related problem to reserve allocation is fair division, where agents have preferences over (non-identical) items, and we seek a Pareto efficient division. The key distinction between these problems is that in fair division, agents' preferences determine the stability of an allocation, while in our setting, the justification for an allocation is dictated by category preferences, while its utility may depend on agent preferences. Nevertheless, the structures of desired allocations in both turn out to be quite similar. Our results provide some intuition as to why this is the case, as when viewed as an ordinal welfare maximization problem, it is clear that the two sides of the market are symmetric. Consequently, our techniques and results share commonalities with this literature. For example, our case study in Section 4.1 recovers results of Saban and Sethuraman [2015] on computing match probabilities under random serial dictatorship. On the other hand, our perturbation approach is foreshadowed by Biró and Gudmundsson [2021]'s, who propose using (pseudo)welfare maximization for computing Pareto efficient fair division solutions.

Finally, settings with two-sided preferences have a long history, stemming from Gale and Shapley's seminal work on the deferred acceptance (DA) algorithm [Gale and Shapley, 1962]. While a fairly robust algorithm, DA can fail to compute a Pareto efficient allocation in the case of indifferences, as pointed out by Erdil and Ergin [2017]. They describe an iterative procedure to Pareto improve an allocation while preserving its stability, illustrating that notions of stability and efficiency can be simultaneously realized. The flow-augmentation ideas in their improvement procedure share commonalities with our arguments in Section 3.

B DEFERRED PROOFS

B.1 Proofs from Section 3.1

Proof of Proposition 1. We will argue the contrapositive — i.e., any $\mathbf{x}$ that does not maximize $V(\mathbf{x})$ is not **[PE]**. Consider the flow network representation of the allocation problem shown in Fig. 5. The nodes on the left side correspond to the agents $a \in \mathcal{A}$, and the nodes on the right to categories $c \in C$. Edges are drawn between each eligible agent-category pair. Finally, given an allocation $\mathbf{x}$, for every category c that has an eligible agent $a \in \mathcal{E}_c$ who is not fully allocated in $\mathbf{x}$, we color all its eligible agents (i.e., all $a' \in \mathcal{E}_c$) red, whether or not they are fully allocated.

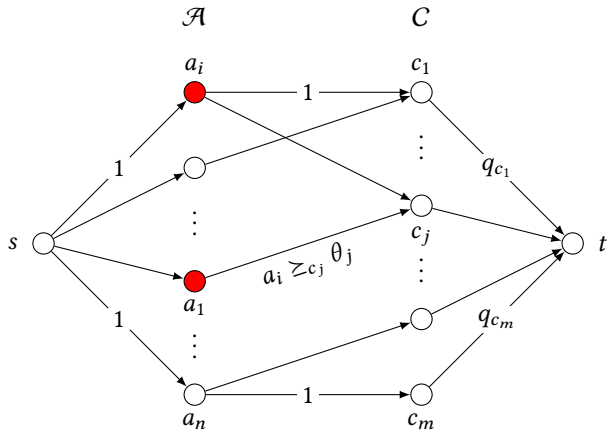

Fig. 5. *A flow network representation of an allocation instance. The source node s has a unit-capacity edge to each agent node. Each category node has an edge to the sink node t with capacity equal to that category's quota. There are unit-capacity edges from each agent node to the nodes of categories in which the agent is eligible.*

If $\mathbf{x}$ is not a maximal allocation, then there is an augmenting path $P = (s, a_1, c_1, \ldots, a_k, c_k, t)$ in this flow network. We record the following observations.

(1) a_1 is red: The in-weight of each agent node is its allocation. Augmenting along P will increase the in-weight of its first agent node, so this agent node must not have been fully allocated.
(2) c_k has not exhausted its quota: The out-weight of each category node is its allocated quota. Augmenting along P will increase the out-weight of its last category node, so this category must not have exhausted its quota.
(3) Given any red agent a, there is a path of the form $s \to a_0 \to c_0 \to a$ in the residual graph for $\mathbf{x}$, where a_0 is a highest-priority agent in c_0 that is not fully allocated: this follows from the definition of red agent nodes.

Let a_i be the *last* red node in P (there must be such a node by Observation 1), and consider the alternate augmenting path $P' = (s, a_0, c_0, a_i, c_i, \ldots, a_k, c_k, t)$ using the "shortcut" from Observation 3. Augmenting along P' will strictly increase the allocation to a_0 and conserves the allocations of $a_i, \ldots, a_k$. Let $\mathbf{y}$ be the allocation after this augmentation. By the construction of the flow network, $\mathbf{y}$ still satisfies **[ER]** and **[QR]**. Moreover, every agent $a \succeq_{c_0} a_0$ is fully allocated, and every agent $a_i, \ldots, a_k$ maintains its allocation in $\mathbf{y}$, so $\mathbf{y}$ also satisfies **[PR]**. Thus, $\mathbf{y}$ is a Pareto improvement to $\mathbf{x}$, meaning $\mathbf{x}$ did not satisfy **[PE]**. □

B.2 Proofs from Section 3.2

The main tool we use to show the reverse implication of Theorem 2 is an alternate characterization of the valid allocations that additionally satisfy **[CS]** as those realizable through *serial dictatorship*. Let Σ be the collection of all multi-set orderings of $\left\{c^{q_c}\right\}_{c \in C}$ (i.e., the set of all sequences of length q wherein each category $c \in C$ appears q_c times). We refer to Σ as the set of choice orders for our system. For a given choice order $\sigma \in \Sigma$, we define the *serial dictatorship allocation* $\mathbf{x}_\sigma$ to be the (integral) allocation obtained by cycling through categories in the order given by σ, and allocating to the highest-priority unallocated agent in the chosen category. This process is formalized in Algorithm 3[4].

Algorithm 3 Serial Dictatorship Allocation

Input: Choice order $\sigma \in \Sigma$

1: **for** each $\sigma_i = c$ in σ in order **do**
2: **if** c has remaining quota and an eligible unallocated agent **then**
3: c allocates to its highest-priority unallocated agent.

Serial dictatorship allocations x_σ generalize the sequential reserve allocations of Pathak et al. [2021]. It is straightforward to see that they (by definition) satisfy **[QR]**, **[ER]** and **[PR]**. Note, however, that $\mathbf{x}_\sigma$ may not be Pareto efficient (for example, consider allocation 2 in Fig. 1 — it can be realized as a serial dictatorship allocation $\mathbf{x}_\sigma$ with $\sigma = (\beta, \gamma, \alpha)$.) The following lemma fully characterizes the allocations obtained via serial dictatorship and generalizes a main result of Pathak et al. [2021].

Lemma 1. *For all $\sigma \in \Sigma$, the serial dictatorship allocation $\mathbf{x}_\sigma$ satisfies* **[QR]**, **[ER]**, **[PR]**, *and* **[CS]**. *Conversely, every valid integral allocation (i.e., obeying* **[QR]**, **[ER]**, **[PR]**, *and* **[PE]**) *that additionally satisfies* **[CS]** *corresponds to a serial dictatorship allocation $\mathbf{x}_\sigma$ under some choice order $\sigma \in \Sigma$.*

Proof. For the first claim, it is immediate from the definition of serial dictatorship that $\mathbf{x}_\sigma$ satisfies **[QR]** (since σ contains q_c copies of c), **[ER]** (since each category c only allocates to eligible agents), and **[PR]** (since a category always allocates to a highest-priority unallocated agent). To see that $\mathbf{x}$ is stable, for any subset S of allocated agents consider the first time that agent $a \in S$ is allocated by a category c. By definition, c selects a highest-priority unallocated agent, so $a \succeq_c s$ for all $s \in S$. Thus, S cannot form an unstable cycle.

To show the second claim (that every valid integral allocation satisfying **[CS]** can be generated via a serial dictatorship allocation), we perform an induction on q. The base case $q = 1$ is trivial: if c is the category with $q_c = 1$, then any valid allocation that also satisfies **[CS]** must give this unit to a highest-priority eligible agent in c, if one exists.

Suppose that the claim holds for all instances with $q = k - 1$, and consider an instance with quota $q = k$. We first show that in any valid and **[CS]** allocation $\mathbf{x}$ (with $V(\mathbf{x}) > 0$), a highest-priority agent in some category is allocated from that category. Suppose that this were not the case, and consider an agent a who is allocated from category c. By assumption, there is some highest-priority agent a' who is not allocated from c. If a' is unallocated, then $\mathbf{x}$ would violate **[PR]**. Hence, a' must be allocated in some other category c'. By assumption, a' does not have highest priority in c', meaning that the highest-priority agent a'' of c' is not allocated in c'. Continuing this reasoning,

[4]For ease of presentation, we ignore ties in Algorithm 3. This assumption corresponds to each category having a total ordering over eligible agents; in case there are multiple unallocated agents in the same highest-priority tier, we can use any fixed tie-breaking rule (alternately, any fixed extension of the total preorder $\succeq_c$).

we will (by finiteness) eventually revisit an agent and discover an unstable cycle, contradicting that $\mathbf{x}$ satisfies **[CS]**.

Now, let c^* be a category allocating to its highest-priority agent, and a^* the highest-priority agent in c^*. We can realize this allocation by having c^* be the first category in the ordering σ. What remains is an allocation problem for agents $\mathcal{A} \setminus \{a^*\}$ to categories C, where the quota of c^* has been reduced by 1. Let $\mathbf{y}$ be the restriction of $\mathbf{x}$ to this problem. It is immediate that $\mathbf{y}$ is a valid and **[CS]** allocation. By our inductive hypothesis, $\mathbf{y}$ can be realized as a serial dictatorship allocation $\mathbf{y}_{\sigma'}$ in this sub-problem. Then, $\mathbf{x}_{(c^*,\sigma')}$ realizes $\mathbf{x}$. □

Using this lemma, we can complete the proof of Theorem 2.

Proof of Theorem 2. For the forward direction, we argue the contrapositive. Suppose that $\mathbf{x}$ is feasible for (P_δ) for some valid δ. Suppose that $\mathbf{x}$ violates **[CS]**, so there are $a_0, a_1, \ldots, a_j = a_0 \in \mathcal{A}$ and $c_0, c_1, \ldots, c_j = c_0 \in C$ for which $x_{a_i,c_i} = 1$, $x_{a_{i+1},c_i} = 0$, and $a_{i+1} \succ_{c_i} a_i$ for each $0 \le i < j$. We construct an alternate solution $\mathbf{x}'$ with $x'_{a_i,c_i} = 0$ and $x_{a_{i+1},c_i} = 1$ for each $0 \le i < j$ and all other variables the same as $\mathbf{x}$. Note that $\mathbf{x}'$ is also feasible since $a' \succ_c a \implies a' \in \mathcal{E}_c$, and all categories and agents have the same total allocation. Since δ is consistent, we have $\delta_{a_{i+1},c_i} < \delta_{a_i,c_i}$ for each $0 \le i < j$, so the reassignment strictly increases the objective value. Thus, $\mathbf{x}$ is not optimal, so it is not a solution to (P_δ).

For the reverse direction, we must construct an assignment of perturbations δ that realize the allocation $\mathbf{x}$ as a solution. It will be convenient to argue using *positive* perturbations (i.e., a bonus rather than a penalty). That is, for every $a \in \mathcal{A}, c \in C$, we set the coefficient of $x_{a,c}$ in the objective as $1 + \rho_{a,c}$, such that $\rho_{a,c} \in [0, \rho_{\max}]$ for all eligible (a, c), and $\rho_{a,c} \ge \rho_{a',c}$ for all $a \succeq_c a'$. To convert the $\rho_{a,c}$ to valid perturbations $\delta_{a,c}$ (Definition 2), we can simply re-scale them by $\frac{1}{1+\rho_{\max}}$ to get $\delta_{a,c} = \frac{\rho_{\max} - \rho_{a,c}}{1+\rho_{\max}}$. Then, it is easy to check that these perturbations satisfy Positivity and Consistency. Also, by choosing $\rho_{\max} = \frac{1}{2|C||\mathcal{A}|}$, we ensure that $\sum_{a,c} \delta_{a,c} \le |C||\mathcal{A}| \cdot \rho_{\max}/(1 + \rho_{\max}) \le 1/2$; thus, the $\delta_{a,c}$ constitute a valid perturbation.

Let $v := V(\mathbf{x})$. By Lemma 1, $\mathbf{x} = \mathbf{x}_\sigma$ for some ordering $\sigma = (\sigma_1, \ldots, \sigma_q) \in \Sigma$. We may also, without loss of generality, assume that the first v entries of σ result in the allocation of an agent: note that any entry σ_i corresponding to a depleted category can be moved to the end of the ordering without affecting the agents available to any later entry.

Now, we set the perturbations as follows:

(1) Let a be the top-ranked agent in the category σ_1. We set $\rho_{a,\sigma_1} = \rho_{\max}$.
(2) In stage i, let $r \le i$ be the lowest rank of an unallocated agent in category σ_i. Let $r' < r$ be the rank of the agent most recently allocated in σ_i, with $r' = 0$ if no agent has yet been allocated through σ_i. For $j = r' + 1, r' + 2, \ldots, r$, let a_j be the agent with rank j in σ_i, and define $A_i = \{a_{r'+1}, a_{r'+2}, \ldots, a_r\}$. We set $\rho_{a_j,\sigma_i} = \rho_{\max}/(|\mathcal{A}| + 1)^{i-1} + (r - j) \cdot \varepsilon$, for some $\varepsilon \ll \rho_{\max}/(|\mathcal{A}| + 1)^{|\mathcal{A}|}$.

The main invariant maintained by the above construction is that at any stage i, the smallest perturbation $\rho_{a,c}$ for $c = \sigma_i$ and any $a \in A_i$ is greater than the *sum of all perturbations* of (a, c) pairs set in rounds $i' > i$. As a result, the optimal matching among pairs (a, c) considered in rounds i and greater must include at least one pair (a_j, σ_i) for some $a_j \in A_i$. Moreover, since the agents $a_{r'+1}, a_{r'+2}, \ldots, a_{r-1}$ were allocated in rounds prior to i, any optimal matching with respect to the $\rho_{a,c}$ must have $x_{a_r,\sigma_i} = 1$. This exactly corresponds to the outcome x_σ realized via Serial Dictatorship with order σ. Thus, x_σ is realized as a solution to (P_δ). □

The following lemma will be useful in our proof of Proposition 2.

Lemma 2. *Consider any valid allocation* $\mathbf{x}$ *with allocated agents* $\mathcal{A}_x = \{a \in \mathcal{A} : \sum_c x_{a,c} > 0\}$. *Then, for any agent* $a^* \in \mathcal{A}_x$ *who is partially allocated (i.e.,* $0 < \sum_c x_{a^*,c} < 1$*), there exists a valid allocation* $\mathbf{y}$ *with* $\mathcal{A}_y \subseteq \mathcal{A}_x$ *and in which* a^* *is fully allocated (i.e.,* $\sum_c y_{a^*,c} = 1$*).*

Proof. Let $c^* \in C$ be any category with $0 < x_{a^*,c^*} < 1$. We argue that there is a way to modify $\mathbf{x}$ which maintains validity, strictly increases x_{a^*,c^*}, and strictly decreases the number of non-integral allocation variables. Since the number of eligible category-agent pairs (and therefore, the number of non-integral allocations) is finite, we can repeatedly apply this modification until a^* is fully allocated.

To describe the modification, we first construct an undirected graph as follows.

- The nodes of the graph will correspond to (a, c) pairs with $0 < x_{a,c} < 1$.
- We color an agent a, and all its associated nodes, red if it is not fully allocated (i.e. $\sum_{c' \in C} x_{a,c'} < 1$), otherwise white.
- We add an edge between any two nodes that share a category.
- We add an edge between any two *white* nodes that share an agent.

Note that the third bullet implies that the connected components of this graph describe a partition of the categories. We argue that the red node (a^*, c^*) is in the same connected component as another red node. For sake of contradiction, suppose not. Note that the total quota of all categories associated with this component is an integer. In addition, the total allocation to all of the agents associated with this component is not an integer: the white agents each have allocation 1, and the singular red agent has a non-integral allocation. However, all of the quotas must be exhausted by the allocation. If not, a path from (a^*, c^*) to a node (a, c) where c has not exhausted its capacity describes a way to adjust the allocation to increase the total allocation to a^* and leave all other agents' total allocations unchanged, violating Pareto efficiency. However, this is a contradiction: the total quota of these categories cannot be both integral and non-integral.

Suppose that $(\hat{a}, \hat{c})$ is another red node in (a^*, c^*)'s connected component. By definition, there is a path between these two nodes. Moreover, the structure of the graph allows us to assume (without loss of generality) that the edges in this graph alternate between connecting nodes that share an agent and nodes that share a category. Since red nodes are only connected to nodes with which they share a category, this path has an odd length. We modify the allocation by following the path from $(\hat{a}, \hat{c})$ to (a^*, c^*). First, we subtract $\varepsilon > 0$ from $x_{\hat{a},\hat{c}}$. Then, we add ε to the variable corresponding to the next node on the path, repeating this process until we add ε to (a^*, c^*): the first bullet point allows us to choose ε such that one of these modifications results in a variable assuming value in $\{0, 1\}$.

To finish the proof, we must argue that this modification results in another valid allocation, which we denote by $\mathbf{x}'$. First, note that the modification did not change the total allocation of any category; it only transferred quota from one agent to another. Thus, $\mathbf{x}'$ satisfies **[QR]**. In addition, we conclude by Proposition 1 that $\mathbf{x}'$ satisfies **[PE]**: it is also a maximal allocation. Next, note that the modification does not transfer any quota to an (a, c) node with $x_{a,c} = 0$, so $\mathbf{x}'$ satisfies **[ER]**. Finally, note that the only agent whose total allocation can decrease from the modification is $\hat{a}$, who is red. Therefore, $\mathbf{x}'$ maintains **[PR]**. □

We are ready to prove Proposition 2.

Proof of Proposition 2. We argue this claim in two stages. First, we argue that $\mathbf{x}$ can be represented as a convex combination of valid allocations in which each agent has an integer total allocation. This follows from Lemma 2. In this proof, we obtained an alternate allocation $\mathbf{x}'$ from $\mathbf{x}$ by perturbing nodes along a path by ε. Similarly, add $\varepsilon' > 0$ to $x_{\hat{a},\hat{c}}$, subtract ε' from the next node, repeating until

we subtract ε' from (a^*, c^*) to obtain an alternate valid allocation $\mathbf{x}''$: the first bullet point allows us to choose ε' such that one of these modifications results in a variable assuming value in $\{0, 1\}$. But then we can express $\mathbf{x}$ as the convex combination

$$\mathbf{x} = \tfrac{\varepsilon'}{\varepsilon+\varepsilon'} \cdot \mathbf{x}' + \tfrac{\varepsilon}{\varepsilon+\varepsilon'} \cdot \mathbf{x}''.$$

We can repeat this process with $\mathbf{x}'$ and $\mathbf{x}''$, just as in the proof of Lemma 2. Since each step strictly decreases the number of non-integral variables, eventually, we will be left with a convex combination of valid allocations $\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(\ell)}\}$ in which each agent has an integer total allocation. (This is the termination condition of the procedure from Lemma 2.) To conclude, we must further represent each $\mathbf{y}^{(i)}$ as a convex combination of valid integral allocations (i.e., allocations in which each allocated agent receives an entire unit from exactly one category). This is an application of the Birkhoff-von Neumann theorem: since each agent is fully allocated, we can interpret $\mathbf{y}^{(i)}$ as fractional matchings between the agents and categories in the subgraph of edges (a, c) with $\mathbf{y}^{(i)}_{a,c} > 0$. Validity is preserved since these integer matchings preserve the total allocation to each agent and category. □

B.3 Proofs from Section 4.1

Proof of Proposition 3. We argue the forward direction by its contrapositive. Suppose that $V^* = V^*_{\setminus a}$, and let $\mathbf{x}$ be a solution to (P_0) for the restricted instance $\mathcal{I}_{\setminus a}$. By Proposition 1, there must be another solution $\mathbf{y}$ that additionally respects priorities (i.e., is valid). Note that a is not eligible in any category in $\mathcal{I}_{\setminus a}$, so $y_{a,c} = 0$ for every $c \in \mathcal{C}$. However, $\mathbf{y}$ is also a valid allocation for the original instance $\mathcal{I}$: eligibility in $\mathcal{I}_{\setminus a}$ implies eligibility in $\mathcal{I}$; quota constraints are the same in $\mathcal{I}$ and $\mathcal{I}_{\setminus a}$; priorities are respected since the definition of restriction ensures that any eligible agent in $\mathcal{I}$ who is not fully allocated in $\mathbf{y}$ must be ranked below fully allocated agents in $\mathcal{I}_{\setminus a}$, and hence in $\mathcal{I}$; finally, $\mathbf{y}$ returns a matching of maximum size in $\mathcal{I}$, and so is Pareto efficient in $\mathcal{I}$. Thus we have located a valid allocation that does not include a, and hence a is not unanimous.

We also argue the reverse direction by its contrapositive. Suppose a is not unanimous — then, there is a valid allocation $\mathbf{x}$ in which a is not allocated. By definition, this allocation has value V^*. Since $\mathbf{x}$ satisfies **[PR]**, no category can allocate to an agent with lower priority than a. Thus, $\mathbf{x}$ is feasible for $\mathcal{I}_{\setminus a}$, so $V^* = V^*_{\setminus a}$. □

Proof of Proposition 4. We show this via a reduction from the X3C problem [Karp, 1972], which is defined as follows.

Definition 6 (X3C). *Given a ground set E of $3n$ elements and a collection of m subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, with each $|S_i| = 3$, the X3C problem asks whether there are n subsets $S_{i_1}, \ldots, S_{i_n}$ whose union is E.*

We consider the following reduction from X3C, which is visualized in Fig. 6.

- $\mathcal{A}$ consists of the following $5m - n + 1$ agents:
  - $3n$ agents representing the ground set elements $e \in E$
  - m agents $s_1, \ldots, s_m$ representing the subsets $S_i \in \mathcal{S}$
  - $4(m - n)$ *filler agents*, labeled $f_1, \ldots, f_{4(m-n)}$
  - the distinguished agent a
- $\mathcal{C}$ consists of $|\mathcal{C}| = m + 1$ categories: a *set category* α_i for each $S_i \in \mathcal{S}$ and a category β.
- Each set category has quota 4, and β has quota 1.
- Each set category α_i has $4(m - n + 1)$ eligible agents: the $4(m - n)$ filler agents, who have priority over agent s_i, who has priority over the 3 element agents in S_i.
- The category β has $3n + 1$ eligible agents: the $3n$ element agents, who have priority over agent a.

$$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$$
$$\mathcal{S} = \Big\{\{e_1, e_3, e_6\}, \{e_1, e_4, e_5\}, \{e_2, e_4, e_5\}\Big\}$$

$\longrightarrow$

α_1 (4)	α_2 (4)	α_3 (4)	β (1)
f_1	f_1	f_1	e_1
f_2	f_2	f_2	e_2
f_3	f_3	f_3	e_3
f_4	f_4	f_4	e_4
s_1	s_2	s_3	e_5
e_1	e_1	e_2	e_6
e_3	e_4	e_4	a
e_6	e_5	e_5	

Fig. 6. An example reduction from an X3C instance (with $n = 2$, $m = 3$) to a reserve allocation instance. This is a "yes" instance of X3C: the first and third sets form a partition of E. Accordingly, the reserve allocation instance on the right admits a valid allocation, visualized in red, that gives to a.

This reserve allocation instance has size which is polynomial in m and n, and it can be constructed in polynomial time. It remains to argue the correctness of the reduction. First, suppose that we are given a "yes" instance to the X3C problem; that is, there are $S_{i_1}, \ldots, S_{i_n}$ that disjointly cover E. Then, consider the following allocation:

- In each category α_{i_j} corresponding to a set S_{i_j}, allocate to agent s_{i_j} and the three element agents.
- In the remaining $m - n$ set categories, allocate to four (distinct) filler agents arbitrarily.
- In category β, allocate to agent a.

Note that this is a valid allocation. It satisfies **[QR]** and **[ER]** by construction, and it exhausts all quotas, so it is **[PE]**. It allocates to all filler and element agents, and to element agents only through categories whose set element is also allocated, so it is **[PR]**. This establishes that a is serviceable.

Conversely, suppose that we reduce to an allocation instance in which a is serviceable, so there is a valid integral (by Lemma 2) allocation φ in which $\varphi(a) \neq \perp$. By construction, a is eligible only in category β, so a must receive the only unit of β. For φ to respect priorities in β, it must allocate to each element agent, which must happen within the set categories. But then, to respect priorities in any set category, φ must allocate to all filler agents as well. In total, these required allocations comprise $4m - n + 1$ units, leaving n units to allocate to the set agents $\{s_i\}$. Since the allocation to one set agent permits the allocation to at most three additional element agents, to allocate to all $3n$ element agents, φ must allocate to *exactly* n set agents, and their corresponding sets must be pairwise disjoint. In summary, the n set agents allocated in φ, $s_{i_1}, \ldots, s_{i_n}$, correspond to n sets $S_{i_1}, \ldots, S_{i_n}$ that disjointly cover E, so we have reduced from a "yes" instance of X3C. □

B.4 Proofs from Section 4.3

Proof of Proposition 5. To see that $\mathbf{x}$ is a valid allocation, it suffices (by Theorem 1) to argue that δ is a valid perturbation. By construction, each $\delta_{a,c}$ is positive, and δ is consistent as $r_c(a) \leq r_c(a')$ if and only if $a \succeq_c a'$. Finally, to see that δ^A has small effect, note that each $r_c(a) \leq |\mathcal{A}|$, and hence

$$\sum_{a \in \mathcal{A}} \sum_{c \in C} \delta_{a,c} \leq \sum_{a \in \mathcal{A}} \sum_{c \in C} \frac{1}{2|C|\,|\mathcal{A}|} = \frac{1}{2}.$$

To conclude that $\mathbf{x}$ minimizes the sum of allocated agents' ranks (among all valid allocations), we consider the objective $V_\delta(\mathbf{x})$. We have

$$V_\delta(\mathbf{x}) = V(\mathbf{x}) - \sum_{a \in \mathcal{A}} \sum_{c \in C} \delta_{a,c} \cdot x_{a,c} = V(\mathbf{x}) - \frac{1}{2|C|\,|\mathcal{A}|^2} \cdot \Big(\sum_{a \in \mathcal{A}} \sum_{c \in C} r_c(a) \cdot x_{a,c} \Big).$$

$V(\mathbf{x})$ is the same for all valid (and thus maximal) allocations. The parenthesized expression is exactly the sum of allocated agents' ranks. Thus, allocations returned by (P_δ) minimize this sum. □

Proof of Proposition 6. To see that $\mathbf{x}$ is a valid allocation, it suffices (by Theorem 1) to argue that δ is a valid perturbation. As before, by construction, each $\delta_{a,c}$ is positive, and δ is consistent as $r_c(a) \le r_c(a')$ if and only if $a \succeq_c a'$, and $\delta_{a,c}$ is an increasing function in $r_c(a)$. Finally, δ has small effect since each $r_c(a) \le |\mathcal{A}|$, and so we have that $\left(\frac{1}{|\mathcal{A}|+1}\right)^{|\mathcal{A}|-r_c(a)} \le 1$. Thus,

$$\sum_{a\in\mathcal{A}}\sum_{c\in C}\delta_{a,c} \le \sum_{a\in\mathcal{A}}\sum_{c\in C}\frac{1}{2|C|\,|\mathcal{A}|} = \frac{1}{2}.$$

Let $R(\mathbf{x}) = \max_{(a,c):x_{a,c}=1}\left\{r_c(a)\right\}$ be the maximum rank over all allocated agents. To conclude that $\mathbf{x}$ minimizes $R(\mathbf{x})$ (among all valid allocations), consider the objective $V_\delta(\mathbf{x})$. We have

$$V_\delta(\mathbf{x}) = V(\mathbf{x}) - \sum_{a\in\mathcal{A}}\sum_{c\in C}\delta_{a,c}\cdot x_{a,c} = V(\mathbf{x}) - \frac{1}{2|C|\,|\mathcal{A}|\cdot(|\mathcal{A}|+1)^{|\mathcal{A}|}}\cdot\sum_{(a,c):x_{a,c}=1}(|\mathcal{A}|+1)^{r_c(a)}.$$

By the definition of $R(\mathbf{x})$, the sum in the last expression falls in the interval $\left[(|\mathcal{A}|+1)^{R(\mathbf{x})}, |\mathcal{A}|\cdot(|\mathcal{A}|+1)^{R(\mathbf{x})}\right]$. Since these intervals are non-overlapping, choosing an integral allocation maximizing V_δ is equivalent to minimizing this sum, and hence minimizing $R(\mathbf{x})$. □

Proof of Proposition 7. This result follows from an X3C reduction that is similar to that from Proposition 4. In particular, when $4(m-n) \ge 3n+1$, the same reduction works, as a is serviceable in the reduced instance if and only if the outer threshold of all categories is at least $3n+1$.

If $4(m-n) < 3n+1$, we need to add more filler agents to the α categories to push the tier of the last f agents past the a agent in category β. In category α_i, we add agents $g_{i,1},\ldots,g_{i,(7n-4m+1)}$, each in a separate rank tier above f_1. We also increase the category's quota to $7n-4m+5$. Again, we have that a is serviceable in the reduced instance (so the X3C instance has a partition by a straightforward modification of the proof of Proposition 4 to account for the $g_{i,j}$ agents) if and only if the outer threshold of all categories is at least $3n+1$. □

Proof of Proposition 8. This result again follows from an X3C reduction that is almost identical to that from Proposition 4. To the reduced instance, we add $4m$ additional filler agents $\{g_1,\ldots,g_{4m}\}$ and an additional category γ with quota $4m$ and all of these g agents in its first priority tier. In addition, we add all of these g agents below a in category β, each in a separate priority tier.

Note that if a remains unallocated, then the maximum possible sum of outer thresholds is $(4m-4n+5)\cdot m + 3n + 1$, where the first term comes from the m set categories, the second term comes from β, and the third term from γ. On the other hand, if a is allocated, then the sum of outer thresholds is at least $(4m-4n+1)\cdot m + (3n+1+4m) + 1$. Thus, a is serviceable in the reduced instance (so the X3C instance has a partition by the proof of Proposition 4) if and only if the sum over categories of the outer thresholds is at least $(4m-4n+1)\cdot m + (3n+1+4m) + 1$. □

Proof of Proposition 9. We perform a reduction from Clique. Given an undirected graph G and clique size k as input, construct an allocation instance $\mathcal{I}_G$ with $\mathcal{A} = V$, $q = k$, and a category c_e for each edge $e \in E$ whose only two eligible agents are the endpoints of e (in the same priority tier). Now G contains a k-clique if and only if the sum of outer thresholds in $\mathcal{I}_G$ equals $\binom{k}{2} + |E|$. □

B.5 Proofs from Section 5

Proof of Proposition 10. We consider a family of allocation instances parameterized by T. There is a single category with quota $q = \frac{T}{2}$ and three eligible types: $a \succ b \succ c$ with $p_a = p_b = p_c = \frac{1}{3}$.

Consider the arrival of an agent of type b "early in the sequence". Although it is almost certain that the hindsight-optimal allocation will accept roughly half of the arriving type-b agents, an online algorithm must reject this agent. To ensure that the hindsight allocation *always* respects priorities, the algorithm must guard against a future (which occurs with positive probability) in which all remaining agents are of type a. The algorithm can therefore never exhaust the quota in a way that would leave some of these agents unallocated and envious. By this reasoning, the algorithm must continue to reject all arriving agents of types b and c until the (random) stopping time τ at which

$$\tfrac{T}{2} - \sum_{t=1}^{\tau} \mathbb{1}\big(\theta_t = a\big) \;\geq\; T - \tau.$$

Here, the left-hand side is the number of remaining units that can be allocated, and the right-hand side is the number of agents to arrive after time τ. We can rearrange this inequality to get

$$\sum_{t=1}^{\tau} \mathbb{1}\big(\theta_t \neq a\big) \;\geq\; \tfrac{T}{2}.$$

If at most $\frac{3T}{8}$ agents among the first $\frac{7T}{8}$ arrivals have type a, then this inequality holds for $\tau = \frac{7T}{8}$, so applying Hoeffding's Inequality, we obtain that

$$\mathbb{P}\big[\tau \leq \tfrac{7T}{8}\big] \geq 1 - \mathbb{P}\big[\mathrm{Binom}(\tfrac{7T}{8}, \tfrac{1}{3}) > \tfrac{3T}{8}\big] \geq 1 - \exp\left(\tfrac{-T}{63}\right).$$

After arrival τ, an algorithm can begin to accept agents of type b (and possibly c). However, if the algorithm has rejected any agents of type b before time τ, it cannot accept any type-c agents. Since all agents of types other than a must have been rejected before time τ, the event that no type-b agents have been rejected before time τ coincides with the event that no such agents arrived. Because $\tau \geq \frac{T}{2}$, the probability of no type-b rejections is therefore at most $(\frac{2}{3})^{T/2}$. By a union bound, with probability at least

$$1 - \exp\left(\tfrac{-T}{63}\right) - \left(\tfrac{2}{3}\right)^{-T/2} = 1 - \exp(-\Omega(T)),$$

the algorithm rejects all type-c agents arriving after time $\tau \leq \frac{7T}{8}$. Again by Hoeffding's Inequality, with probability at least $1 - \exp(-\Omega(T))$, there are at least $\left(\frac{1}{24} - \epsilon\right) \cdot T$ such agents (for any constant $\epsilon < \frac{1}{24}$, e.g., $\epsilon = \frac{1}{100}$), resulting in $\Omega(T)$ loss in efficiency with all but exponentially small probability. □

C SCARF'S LEMMA AND STABLE MATCHING

Here, we compare the priority-respecting allocations problem and the stable matching problem through the lens of Scarf's lemma. To begin, we recall Scarf's Lemma, following the treatment of Nguyen and Vohra [2022].

Scarf's lemma considers an allocation setting with *n agents* and *m coalitions*; both of these descriptors apply rather abstractly, as our examples will illustrate. In this setting, coalitions comprise agents and the principal must decide how to allocate coalitions. There are budgetary constraints that ensure that no agent is over-allocated and agents may express preferences over the coalitions to which they belong. Formally, we have:

- A matrix $\mathbf{A} \in \mathbb{R}_+^{n\times m}$ has a row for each agent and a columns for each coalition.We interpret entry $\mathbf{A}_{ij}$ as a cost to agent i if (one unit of) coalition j is allocated. We assume that each row includes at least one positive entry.
- A vector $\mathbf{q} \in \mathbb{R}_+^n$ denotes the budget of each agent.
- Each agent i has a total preference order $\succeq_i$ over its coalitions $\{j \in [m] : \mathbf{A}_{ij} > 0\}$.
- A vector $\mathbf{x} \in \mathbb{R}_+^m$ stipulates to what extent each coalition is realized.

Thus, the principal must select $\mathbf{x}$ subject to the budgetary constraints $\mathbf{Ax} \le \mathbf{q}$. Within this set of feasible $\mathbf{x}$, we wish to further choose coalitions that enforce some notion of stability with respect to the agent preferences. For this, we introduce the notion of the domination of a coalition.

Definition 7. *Given an instance* $\left(\mathbf{A}, \mathbf{q}, (\succeq_i)_{i\in[n]}\right)$*, an allocation* $\mathbf{x} \ge 0$ *satisfying* $\mathbf{Ax} \le \mathbf{q}$ dominates *coalition* $j \in [m]$ *if there is some fully-allocated agent* i *for which every allocated coalition to which* i *belongs is weakly preferred by* i *to* j*.*

More formally, there is $i \in [n]$ *such that* $\sum_{k=1}^m \mathbf{A}_{ik}\mathbf{x}_k = \mathbf{q}_i$ *and for each* $j' \in [m]$*,*

$$\mathbf{A}_{ij'} > 0 \text{ and } \mathbf{x}_{j'} > 0 \implies j' \succeq_i j.$$

Domination expresses an inability to adjust $\mathbf{x}$ in a way that assigns more weight to coalition j without upsetting some agent i. Simply increasing $\mathbf{x}_j$ would violate i's budgetary constraint, and any shift in weight from any other coalition j' to which i belongs would come from a coalition preferable to j. Through this interpretation, if an allocation were to dominate all coalitions, it would exhibit a notion of *stability*; any adjustment of the coalition allocations would be either inefficient or disagreeable to some agent. The ensured existence of such stable allocations is the content of Scarf's Lemma.

Proposition 11 (Scarf [1967], Theorem 1)**.** *Given any allocation instance* $\left(\mathbf{A}, \mathbf{q}, (\succeq_i)_{i\in[n]}\right)$*, there is an extreme point of* $\{\mathbf{x} : \mathbf{Ax} \le \mathbf{q}\}$ *that dominates every coalition.*

Scarf's Lemma can be proven via a reduction to the existence of Nash Equilibria in two-person games. While the statement of this result is clean, allowing it to be specialized to many problems (as we discuss below), there is no assurance that this dominating extreme point can be easily computed. In fact, Kintali [2008] showed that a computational version of Scarf's Lemma is complete for the PPAD class. This implies that there is no polynomial-time algorithm to locate these extreme points unless PPAD $\subseteq$ P. Despite this, there are many special cases of Scarf's Lemma that admit polynomial algorithms.

One special case of Scarf's Lemma is the stable matching problem; it can be used to recover the result of Gale and Shapley [1962] that a stable matching exists in every instance. Using the context of n residents being matched to n hospitals, we take the set of agents to be the union of the residents and hospitals. The coalitions consist of each (resident, hospital) pair, and $\mathbf{A}$ is the $\{0, 1\}$ incidence matrix. The budget vector $\mathbf{q} \in \mathbb{R}_+^{2n}$ is the all-ones vector, which ensures that each agent is matched at most once. The agent rankings $(\succeq_i)$ order an agents incident pairs corresponding to

their preference list. In this construction, an undominated coalition corresponds to an instability. Note that the Birkhoff-von Neumann theorem ensures the integrality of the extreme points.

The priority-respecting allocation problem can also be interpreted as a special case of Scarf's lemma. However, the construction is less straightforward, as we must account for the lack of preferences of the agents. Here, the set of agents consists of all of the categories along with a copy of each of the agent for each possible ordering of their eligible categories. The set of coalitions consists of all eligible (agent, category) pairs, and $\mathbf{A}$ is again a $\{0, 1\}$ incidence matrix. The budget of each category is its quota, and the budget of each agent is 1. The preference order of a category c is any linear extension of $\succeq_c$. The preference order of each agent row corresponds to its ordering over its eligible categories.

C.1 LP Perturbations and Stable Matchings.

One problem with using Scarf's Lemma is that while it guarantees the existence of a dominating solution, it does not give an efficient algorithm for finding it. Apart from priority-respecting allocation, the other setting where the dominating solution was known to be efficiently computable was for stable matchings. The underlying reason behind the existence of an efficient algorithm in the two settings, however, appears to be very different. On one hand, stable matchings are known to form a convex set (and indeed, are realized as corner points of a natural modification of the matching LP [Vande Vate, 1989]), while as we show in Fig. 2, this is not the case for priority-respecting matchings. On the other hand, we show that the perturbation techniques we develop for locating priority-respecting allocations does not work for stable matchings.

A naïve way to locate stable matchings via LPs is to first compute a stable matching (which can be done efficiently via the Deferred-Acceptance procedure of Gale and Shapley [1962]), and then design edge weights to recover the same matching as a maximum-weight matching. More surprisingly, the work of Vande Vate [Vande Vate, 1989] shows that one can modify the matching polytope by adding additional linear constraints to get an LP whose corner points exactly correspond to all the stable matchings, and one can use the corresponding optimal dual variables to get perturbed objectives. The problem with these procedures, however, is that they compute perturbations that are *global*, i.e., based on the entire instance. This is in contrast to our technique for finding priority-respecting allocations, which is based on *local* perturbations: the objective coefficient on edges x_{mw} are functions only of the rank of m on w's preference list and the rank of w on m's preference list.

Thus, a more refined question is if given a stable matching instance with n men and n women, one can find a perturbation function $F\colon [n] \times [n] \to \mathbb{R}$ such that the resulting matching M that maximizes $V_F(M) = \sum_{(m,w)\in M} F(r_w(m), r_m(w))$ is necessarily stable. Unfortunately, we can answer this question in the negative.

Proposition 12. *For $n \geq 6$, for any local perturbation function $F\colon [n]\times[n] \to \mathbb{R}$, there exist instances such that any matching M maximizing $V_F(M) = \sum_{(m,w)\in M} F(r_w(m), r_m(w))$ is unstable.*

Proof. We consider two stable matching instances with $n = 6$. In both instances, the women are indexed by Roman letters $\{a, b, c, d, e, f\}$ and the men are indexed by Greek letters $\{\alpha, \beta, \gamma, \delta, \epsilon, \zeta\}$. The first instance has the following preference lists:

a	b	c	d	e	f
α	β	γ	α	α	α
$*$	$*$	$*$	δ	β	ϵ
$*$	$*$	$*$	ζ	ϵ	β
$*$	$*$	$*$	$*$	δ	ζ
$*$	$*$	$*$	$*$	$*$	$*$
$*$	$*$	$*$	$*$	$*$	$*$

α	β	γ	δ	ϵ	ζ
a	b	c	a	a	a
$*$	$*$	$*$	e	b	f
$*$	$*$	$*$	b	e	d
$*$	$*$	$*$	c	f	$*$
$*$	$*$	$*$	d	$*$	$*$
$*$	$*$	$*$	$*$	$*$	$*$

Here, the $*$ elements can be assigned arbitrarily to complete the matching instance. Note that in this instance, there is a unique stable matching, $M = \{(a,\alpha),(b,\beta),(c,\gamma),(d,\delta),(e,\epsilon),(f,\zeta)\}$. An alternate (non-stable) matching is $M' = \{(a,\alpha),(b,\beta),(c,\gamma).(d,\zeta),(e,\delta),(f,\epsilon)\}$; note the presence of instability (e,ϵ). For our function F to assign a higher value to matching M than M', we must have

$$\begin{aligned} V_F(M) &= F(1,1) + F(1,1) + F(1,1) + F(2,5) + F(3,3) + F(4,2) \\ >V_F(M') &= F(1,1) + F(1,1) + F(1,1) + F(3,3) + F(4,2) + F(2,4), \end{aligned}$$

which simplifies to the condition $F(2,5) < F(2,4)$.

We similarly consider our second stable matching instance.

a	b	c	d	e	f
α	β	γ	α	α	α
$*$	$*$	$*$	δ	β	ϵ
$*$	$*$	$*$	ζ	ϵ	β
$*$	$*$	$*$	$*$	δ	ζ
$*$	$*$	$*$	$*$	$*$	$*$
$*$	$*$	$*$	$*$	$*$	$*$

α	β	γ	δ	ϵ	ζ
a	b	c	a	a	a
$*$	$*$	$*$	e	b	f
$*$	$*$	$*$	b	e	d
$*$	$*$	$*$	d	c	$*$
$*$	$*$	$*$	$*$	f	$*$
$*$	$*$	$*$	$*$	$*$	$*$

In this instance, $M = \{(a,\alpha),(b,\beta),(c,\gamma),(d,\delta),(e,\epsilon),(f,\zeta)\}$ is again the unique stable matching. The alternate matching $M' = \{(a,\alpha),(b,\beta),(c,\gamma).(d,\zeta),(e,\delta),(f,\epsilon)\}$ is again unstable; note the presence of instability (e,ϵ). For our function F to assign a higher value to matching M than M', we must have

$$\begin{aligned} V_F(M) &= F(1,1) + F(1,1) + F(1,1) + F(2,4) + F(3,3) + F(4,2) \\ >V_F(M') &= F(1,1) + F(1,1) + F(1,1) + F(3,3) + F(4,2) + F(2,5), \end{aligned}$$

which simplifies to the condition $F(2,4) < F(2,5)$. Our two derived inequalities cannot be simultaneously satisfied. Hence, such a local perturbation function F cannot exist. □