Achieving Rawlsian Justice in Food Rescue

Aydın Alptekinoğlu
Smeal College of Business, The Pennsylvania State University, University Park, PA 16802
aydin@psu.edu

Gerdus Benadè
Questrom School of Business, Boston University, Boston, MA 02215
benade@bu.edu

We study a problem faced by a national food rescue platform that matches each donation to the first recipient who claims it. Recipients have very different response rates, leading to a few highly responsive recipients claiming the bulk of the donations. We ask whether *priority lists*, which control when the donation is announced to each recipient, are a remedy for such inequitable outcomes. We show that an n-stage priority list, with an individual notification time specific to each eligible recipient, can achieve any desired expected allocation, hence any fairness target, if the donation is non-perishable. Two-stage or binary priority lists, which notify eligible recipients in only two waves, are simpler to implement and administer but offer less fine-grained control over fairness outcomes. For both perishable and non-perishable donations, we give polynomial-time algorithms to find the n-stage and binary priority lists that optimize a class of Rawlsian objective functions – maximize the minimum value – representing a notion of fairness that focuses on the worst-off. The key insight that animates optimal priority list design throughout is to give higher priority (and thus more time to claim a donation) to recipients who have received less in the past and to those who were slower in responding to notifications. We prove that this idea can be codified into a simple index by which to rank order eligible recipients. Computational experiments calibrated by real data confirm that even simple, binary priority lists lead to significantly more fair allocations than the existing first-come-first-serve allocation system.

Key words: food rescue, food waste, food insecurity, fairness, nonprofit operations, donations, matching

---

1. Introduction

Food insecurity is a major societal issue worldwide and even in developed nations. According to an estimate by the U.S. Department of Agriculture (USDA), 1.14 billion people on the planet, about 14.1% of the world population, faced food insecurity in 2023 (Zereyesus et al. 2023). USDA also reports that 12.8% of U.S. households and 8.8% of U.S. households with children experienced food insecurity in 2022 (Rabbitt et al. 2023). At the same time, food waste is extremely prevalent. USDA estimates that 31% of all food produced in the U.S., amounting to at least 92 billion pounds of food, is wasted somewhere along the food supply chain (Buzby et al. 2014). Reducing food waste is among the United Nations 2030 Sustainable Development Goals (United Nations 2015, Goal 12.3).

A common tactic to mitigate food insecurity in the presence of ample surplus or excess food is *food rescue*. The idea is to match, in a timely manner, organizations that have excess food with organizations that can distribute the excess food where it is needed before it expires. Strengthening food rescue is one of seven key action areas that ReFED, a nonprofit organization dedicated to fighting food loss and waste, advocates (ReFED 2024).

Our paper concerns fairness in food rescue and is motivated by a collaboration with a national food rescue platform called FoodRecovery.org (hereon abbreviated as *FR*) operating in 50 states in the U.S. *FR* runs a two-sided platform where food donations are posted by organizations with surplus food and claimed on a first-come-first-serve (FCFS) basis by nonprofit organizations (typically food pantries or food kitchens) that serve communities in need. The platform alerts all potential recipients (those within a reasonable distance from the donor location, as defined by each recipient) by notification via email or text. The matching between the donor and a recipient then occurs automatically in FCFS fashion: Whoever claims the donated food first gets it. The fastest-claiming recipient makes arrangements to pick up the food from the donor's site, sometimes with *FR*'s logistical help. By connecting organizations with surplus food to organizations that serve communities in need, *FR* diverts edible food away from landfills, and helps mitigate food insecurity *and* reduce food waste, serving both social responsibility and environmental sustainability goals.

FCFS matching is operationally efficient and simple to execute. However, because donations do not necessarily go to the organization that needs it the most but rather to the fastest responder, this allocation policy may also raise fairness concerns. One fear is that it places organizations with more limited resources at a disadvantage. For example, smaller or volunteer-run organizations that lack the staff to constantly monitor email or text are often unable to claim quickly, resulting in fewer donations claimed by such organizations and potentially systemic inequities in how the donated food is distributed. These fears are justified: In a two-year sample of data from several counties in Florida, for example, we observe that the bottom 60% of recipient organizations receive roughly 13.5% of the total pounds distributed and only 8.5% of donations. The distribution of food handled by *FR* is visualized in Figure 1(a) with a Lorentz curve, where the point (x, y) indicates that the bottom x% of recipients (after sorting) received y% of the total. These numbers are reported considering only recipients who received at least one donation in those two years and are thus a conservative estimate of the inequity — a sizable fraction of recipients who signed up to the service never received any donations.

In this paper, we explore a practical idea to improve the fairness of how donated food gets distributed via food rescue platforms like *FR* that operate on an FCFS basis. At the core of our paper lies the idea of a *priority list*, which gives priority (first dibs in claiming a food donation) to a select set of recipients for a limited time window. Priority lists simply require a tiered notification system, in

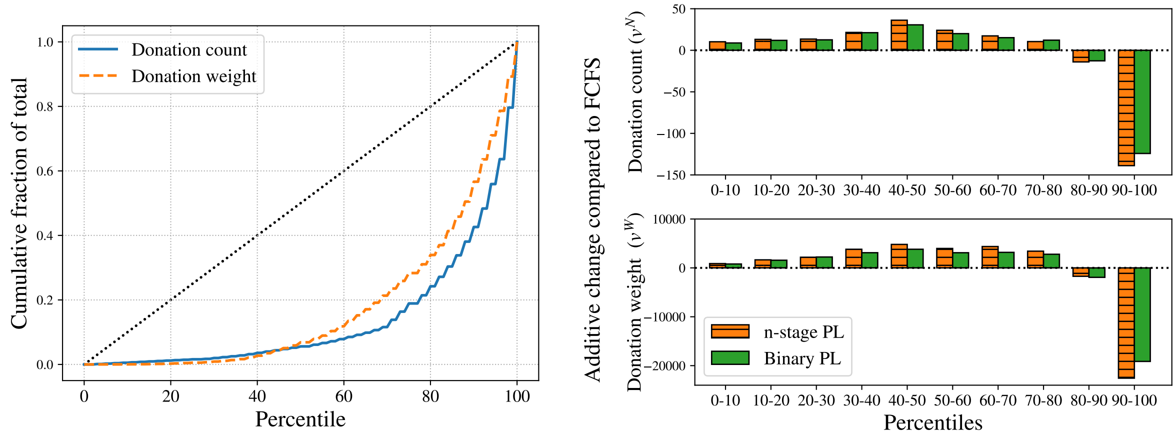

Figure 1 (a) Historical distribution of donations in a two-year period (May 2021 – September 2023) under FCFS matching. (b) The simulated additive change in the number and weight of donations received by each decile when using n-stage and binary priority lists instead of FCFS.

which every eligible recipient potentially gets assigned a specific notification time, allowing a donation to be announced to some recipients before others. Layering such recipient-specific notification times on top of FCFS matching satisfies *FR*'s desire for simplicity and minimal disruption to the existing IT system that handles the platform's matching and notification logic. It also continues to involve recipients in the matching process, something *FR* finds invaluable because it avoids inappropriate or wasteful allocations that may result from incomplete or imperfect information about the recipient's real-time operational constraints (e.g., logistics capabilities like storage and transportation). Furthermore, priority lists are easy to explain and, from the recipients' perspective, minimally disruptive since the process to claim a donation works exactly as before.

1.1. Our Contributions

Broadly speaking, our paper is a proof of concept on how priority lists can lead to more equitable distribution of food donations posted on an FCFS food rescue platform. To that end, our contributions include showing the structure of the optimal priority list design for perishable and non-perishable donations, and developing efficient algorithms to construct optimal priority lists with the objective of maximizing a class of Rawlsian objectives under exponentially distributed response times from recipients. Throughout the paper we also treat a special case (binary priority lists) that is appealing for implementation at *FR* because of its simplicity.

We first develop an analytical model of food rescue with priority lists. We then fully characterize – through structural results and algorithmic development – the optimal priority list for a given donation that maximizes the benefit or value (defined as pounds of food received, or number of donations claimed, among other possibilities) gained by the worst-off recipient on the platform. This objective reflects a Rawlsian notion of fairness. The model captures priority lists in their most general form:

For each donation posted on *FR*'s platform, it sets a notification time for each eligible recipient, after which they can claim the donation if it is yet to be claimed. We call this an *n-stage priority list*. Our model can also accommodate perishable donations, which must be claimed by a set time, and a more restricted version of priority lists that allow only two waves of notifications, called a *binary priority list*. Intuitively speaking, our structural results reveal that it is optimal (in a max-min sense) to give higher priority to recipients that have received less than others in the past (so, they are currently among the worst-off) and to recipients that are slower than others in responding to posted donations. A simple index combines these two characteristics of recipients and provides the optimal priority ordering. Remarkably, the same index applies to all the cases.

In §3.1.1 and §3.2.2, we study n-stage priority lists for non-perishable and perishable donations, respectively. We observe that any fractional allocation can be achieved by an n-stage priority list if the donation is non-perishable, but the same cannot be said of perishable donations. Despite this, we show for both types of donations that it is optimal to use the following priority ordering, which also determines the sequence of notifications: order recipients in decreasing order of desired fractional allocation to achieve a certain objective function value divided by response rate. So, the numerator is higher for those recipients who have received less in the past, and the denominator is lower for those who have smaller response rates, implying that larger need and slower response lead to higher priority, thus earlier notification. This observation, together with some basic algorithmic development, is enough to maximize the value delivered to the worst-off recipient.

In §3.1.2 and §3.2.1, we study 2-stage or binary priority lists, in which a donation is first announced to some subset of recipients (priority set) and, after some time, announced to the rest. Binary priority lists are simpler but less flexible than n-stage priority lists, and cannot achieve every fractional allocation. However, we show that the same priority ordering leads to optimal binary priority lists for both non-perishable and perishable donations: sort recipients by the ratio of their desired fractional allocation to their response rate and include them in the priority set in this order (up to some cut-off). This structural result enables efficient algorithms to compute the optimal binary priority list for any degree of perishability.

Finally, in §4, we test our notions of improving fairness on real data from *FR*'s Florida operations. We present counterfactuals through simulation that quantify the potential impact of implementing priority lists in practice. Using the historical data to calibrate the response rates and donation sizes, we simulate the existing FCFS system as well as the optimal or near-optimal n-stage and binary priority lists. Priority lists lead to a significantly more equitable distribution of donations across all metrics (see Tables 2 and 3, and Figures 4 to 6). For example, in our simulations the bottom 60% of recipients receive roughly 19.6% of the total pounds distributed and 23.1% of donations under the optimal n-stage priority list, 18.4% and 21.4% under the optimal binary priority list, significantly higher than 9% and 8.9% under FCFS. Compared to FCFS, both strategies are effective at decreasing the allocations of the best-off recipients and redistributing these donations to worse-off recipients, as visualized in Figure 1(b). About 22,500 (19,140) pounds of food, or 15.6% (13.2%) of the total allocated, get redistributed from the top 10% to the bottom 80% by n-stage (binary) priority lists. The two types of priority lists perform very similarly in our simulations. Thus, the simplicity of binary priority lists may be a good reason to prefer them in practice. The simulations also reveal a drawback of priority lists: They slow down claim times, suggesting that donors' expectations must be managed when priority lists are first deployed.

1.2. Related Work

We briefly discuss the related literature. Two streams of literature are especially relevant to our work: fairness in resource allocation and waste in food supply chains.

Fairness is a fairly big and interdisciplinary topic. We start by discussing a few papers that use a similar notion of fairness to ours or those that take motivation from food rescue and other nonprofit settings. Kawase and Sumita (2022) maximize minimum welfare in an online allocation problem and provide approximations to the optimal max-min welfare in hindsight, under the assumption that valuations are additive and item values are stochastic. Motivated by the centralized distribution of a stockpile of medical resources to states with needs during the COVID-19 pandemic, Manshadi et al. (2023) study the problem of maximizing the minimum fill rate in a model where a stationary pool of resources is allocated to recipients who arrive sequentially (and only once) with stochastic demands. They provide lower and upper bounds for the objective function and show that a simple adaptive policy achieves the upper bound while outperforming non-adaptive policies. Similar to both these papers, we maximize the minimum value delivered to recipients, and our results apply to both the objectives of welfare and fill rate, among others. In contrast, we focus on the effect of response rates in a sequence of single-shot FCFS problems where the allocation mechanism can control the (expected, fractional) allocation only indirectly through the construction of an appropriate priority list. Compared to Kawase and Sumita (2022), we do not require stochastic valuations. Unlike in Manshadi et al. (2023) where recipients arrive and request resources only once, in our setting the resources arrive and recipients may be eligible to receive resources in multiple time periods.

Several papers explore allocation mechanisms in real-world food rescue organizations. Prendergast (2022) analyzes a marketplace run by Feeding America, which lets each recipient bid some artificial currency on roughly 30 truckloads of food donations every day. In our context donations are smaller and less regular, making it harder to ask for (cognitively demanding) bids on the items. Lee et al. (2019), in collaboration with 412 Food Rescue, develop an algorithmic framework that lets dispatchers train a model that recommends who should receive a donation and how it should be delivered to them. Shi et al. (2020), working with the same organization, propose a machine learning algorithm to recommend which recipients should be contacted first for a given donation. Sinclair et al. (2022) draw inspiration from a food bank operating a mobile food pantry to study a model in which recipients arrive sequentially (very similar to Manshadi et al. (2023)) with the dual objectives of minimizing envy and maximizing efficiency. Orgut and Lodree (2023) develop a network flow model to study equitable distribution of perishable food items from a food bank in North Carolina to the charitable organizations they work with.

The fairness literature includes further studies of fair allocation of indivisible resources arriving online with an eye toward minimizing envy (Benadè et al. 2018) and simultaneously maintaining efficiency (Benadè et al. 2022b, Zeng and Psomas 2020), even in cases where one has only limited information about recipients' (stochastic) preferences (Benadè et al. 2022a). Closer to our focus, Banerjee et al. (2023) aim for proportional instead of envy-free allocations and ask if having predictions of recipients' values for goods helps. In contrast, our notion of fairness is maximizing the welfare of the worst-off recipient rather than proportional allocation or minimizing envy. We also only require valuations to be additive and monotone.

In the operations management (OM) literature, there has been increased attention and a recent call to action on food waste (Akkaş and Gaur 2022). One of the prime concerns is inventory control and replenishment decisions at different points of food retail supply chains (Akkaş and Honhon 2022, 2023, Belavina 2021). Some retail operations topics such as shelf space management (Akkaş 2019), store location density (Belavina 2021) and sales promotions (Wu and Honhon 2023) have also been studied in connection to food waste. One common thread among these papers is that decisions being modeled are profit-driven (as opposed to a social good objective in our setting) with some care placed on the resulting food waste or environmental impact. We contribute to this literature by exploring a tradeoff between fairness and operational efficiency at food rescue platforms, which are nonprofit organizations that have reducing food waste at the core of their business model.

We identified two papers in the OM literature that directly study some aspect of food rescue, and they both study the management of volunteers in food rescue settings. Manshadi and Rodilitz (2022) work with Food Rescue U.S. to improve their volunteers' engagement with the platform in the form of providing transportation help – taking donated food from the donor's location to a local recipient organization. The key tradeoff they address is between efficiency (making sure transportation help is provided) and apathy (not sending too many notifications to volunteers to lead them to disengagement, albeit temporarily). Ata et al. (2019) draw their motivation from a practice called *gleaning*, where farmers donate some of their crops that they deem uneconomical to harvest. This is a form of food rescue that occurs at the most upstream nodes of the food supply chain and it requires significant labor. Ata et al. (2019) develop a volunteer management model in the face of two key sources of uncertainty: the amount of crop to be picked and the supply of volunteer hours that can be reliably recruited.

2. A Model of Fairness in Food Rescue

In this section we develop a model that captures in sufficient detail *FR*'s current mode of operation, which is strictly FCFS, and how it would operate under priority lists. We then define the notion of fairness that we want to analyze, and state *FR*'s problem of designing optimal priority lists that maximize this fairness metric.

2.1. Donations and Recipients

Consider a sequence of m food donations arriving – being posted on *FR*'s system one after the other by various donors. For logistical reasons, they each are allocated in their entirety to a single potential recipient as soon as possible and irrevocably. A donation typically consists a quantity of food from one or more categories (for example, produce, dairy or prepared meals). Let s_j denote the size of donation $j = 1, \ldots, m$, typically measured in pounds. Donations may be perishable, meaning that donation j must be allocated within some time T_j after arrival ($T_j = \infty$ if it is non-perishable). In practice, these deadlines result from food expiration dates and, more commonly, a myriad of logistical constraints of the donor (e.g., warehouse space).

FR has a set of potential *recipients* that can put these food donations to good use. Let $\mathcal{N}$ denote the universal set of recipients and n the size of this set: $\mathcal{N} \equiv [n] \equiv \{1, \ldots, n\}$. Donation j is eligible to be claimed by a subset of recipients $R_j \subseteq \mathcal{N}$. Let $r_j = |R_j|$ be the size of this subset. Eligibility for a given donation depends on two factors: First, when signing up, recipients specify a travel radius within which they are willing to pick up donations; second, recipients indicate which food categories they are interested in. A recipient i is included in R_j when the donor is located within i's pickup radius and i indicated an interest in the category of the donation.

Recipients differ in how quickly they respond to notifications by *FR*. We define *response time* τ_{ij} as the time that elapses between recipient i being notified of donation j and moving to claim it, and assume that it follows an exponential distribution with parameter $\lambda_i > 0$, termed *response rate*, i.e., $\tau_{ij} \sim \text{Exp}(\lambda_i)$ for $i \in R_j$, with independence across recipients and donations.

Each recipient has a value function $v_i(\cdot)$ that may depend on the amount and type of food they receive. Consider donation j just arriving. Let $\underline{y}_i^{j-1}$ be the $(j-1)$-vector of past allocations to recipient i with elements $y_{ik} \in \{0, 1\}$ being 1 if donation k was allocated to recipient i and 0 otherwise for $k = 1, \ldots, j-1$. The historical allocation data are binary, because each donation is allocated in its entirety to a single recipient by assumption (and in practice). The allocation of the most recent arrival, on the other hand, is treated as fractional; that is, for expositional convenience, we treat it as a probability, and sometimes refer to it as a fractional allocation. Let $x_{ij} \in [0, 1]$ be the probability that recipient $i \in R_j$ claims donation j within the time period $[0, T_j]$ and $x_{0j} = 1 - \sum_{i \in R_j} x_{ij}$ be the remaining probability – that donation j goes unclaimed by time T_j. The expected value recipient i obtains from donations $1, \ldots, j$, $v_i(\underline{y}_i^{j-1}, x_{ij})$, is assumed to be a continuous function of the vector $(\underline{y}_i^{j-1}, x_{ij})$, strictly increasing in x_{ij} and additive (over donations), with marginals bounded by some constant.

This value function specification can capture many practical metrics that *FR* expressed interest in and found potentially feasible and insightful to track, including:

- The number of donations recipient i received, i.e., $v_i^{\mathrm{N}}(\underline{y}_i^{j-1}, x_{ij}) = \sum_{k=1}^{j-1} y_{ik} + x_{ij}$.
- The total weight (in pounds) recipient i received, i.e., $v_i^{\mathrm{W}}(\underline{y}_i^{j-1}, x_{ij}) = \sum_{k=1}^{j-1} y_{ik} s_k + x_{ij} s_j$.
- The fraction of recipient i's cumulative demand d_i (in pounds) fulfilled by donations, i.e., $v_i^{\mathrm{D}}(\underline{y}_i^{j-1}, x_{ij}) = (\sum_{k=1}^{j-1} y_{ik} s_k + x_{ij} s_j)/d_i$.
- The urgency or current need at recipient i's operations for the type of food donation j includes, i.e., $v_i^{U}(\underline{y}_i^{j-1}, x_{ij}) = \sum_{k=1}^{j-1} y_{ik} s_k u_{ik} + x_{ij} s_j u_{ij}$, where u_{ik} are the per-pound utility of donation $k = 1, \ldots, j$ for recipient i.

Currently *FR* is tracking only the first two metrics. If the cumulative demand data was available, the third metric (akin to *fill rate* in inventory control) would ensure that priority lists take into account the recipients' scale of operation. A large food bank serving thousands of meals a month versus a church that can accommodate a much smaller demand would then be compared on the percentage of the total demand they serve using donations through *FR*. (We are encouraging *FR* to eventually adopt this metric.) Due to varying nutritional values of food or cultural diversity, recipients may prefer specific types of food closer to the tastes or dietary requirements of the people they serve; some version of the fourth metric could cater priority lists towards such preferences. Throughout the paper we analyze general valuation functions of this form, for which we use the $v_i(\cdot)$ notation, and employ the superscripts only in examples and simulations to refer to one of the special cases above.

2.2. Allocations under FCFS (The Status Quo) and Priority Lists (The Future)

FR currently allocates donations to recipients on an FCFS basis: Arriving donations are announced immediately to all recipients in R_j and allocated in its entirety, or *matched*, to the first recipient who claims it. With $\tau_{ij} \sim \mathrm{Exp}(\lambda_i)$, recipient i's expected response time is $\mathbb{E}[\tau_{ij}] = 1/\lambda_i$ and her probability of claiming a donation j within t time units of its posting is $F_i(t) \equiv \mathbb{P}(\tau_{ij} \le t) = 1 - \exp(-\lambda_i t)$ when there are no other eligible recipients. For any subset of recipients $S \subseteq \mathcal{N}$, let $\lambda_S \equiv \sum_{i \in S} \lambda_i$ be their total response rate. Their independent exponential response times imply that the earliest response time from this group of recipients also has an exponential distribution, i.e., $\tau_S \equiv \min_{i \in S}\{\tau_{ij}\} \sim \mathrm{Exp}(\lambda_S)$, so $\mathbb{E}[\tau_S] = 1/\lambda_S$ and $F_S(t) \equiv \mathbb{P}(\tau_S \le t) = 1 - \exp(-\lambda_S t)$. Furthermore, conditional on a recipient in S responding, the probability that recipient $i \in S$ responds first is λ_i/λ_S. As a result, in the current FCFS regime, the fractional allocations are $x_{0j} = 0$ and $x_{ij} = \lambda_i/\lambda_{R_j}$ $\forall i \in R_j$ for a non-perishable donation j, and $x_{0j} = \exp\big(-\lambda_{R_j} T_j\big)$ and $x_{ij} = \big[1 - \exp\big(-\lambda_{R_j} T_j\big)\big] \cdot \lambda_i/\lambda_{R_j}$ $\forall i \in R_j$ for a perishable donation j with deadline $T_j < \infty$.

EXAMPLE 1. Consider three recipients with response rates $\lambda_i = i$, for $i = 1, 2, 3$, all eligible to receive a non-perishable donation j. FCFS results in the fractional allocation $(x_{0j}, x_{1j}, x_{2j}, x_{3j}) = (0, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}) \approx (0\%, 17\%, 33\%, 50\%)$. If this same donation was perishable with $T_j = \frac{1}{2}$, FCFS would then yield $(x_{0j}, x_{1j}, x_{2j}, x_{3j}) = (e^{-3}, \frac{1}{6}(1 - e^{-3}), \frac{2}{6}(1 - e^{-3}), \frac{3}{6}(1 - e^{-3})) \approx (5\%, 16\%, 32\%, 47\%)$. $\square$

FR's FCFS matching is a fairly passive allocation scheme which, as the previous example shows, can lead to inequitable outcomes because recipients vary in their response rates. To nudge the platform toward more fair allocation outcomes, where slow-responding organizations also have a good chance of claiming donations, we devise, in close collaboration with *FR*, a minimally invasive idea: Give some recipients a head start by notifying them earlier than other recipients about newly available donations (equivalently, by delaying the notification of other recipients). We call this tiered notification system a *priority list*. In its most general form, a priority list for donation j is a vector $\underline{t}_j$ that specifies a notification time $t_{ij} \in [0, T_j]$ for each and every eligible recipient $i \in R_j$. We call this a *general* or *n-stage priority list*. (Strictly speaking, there are up to $r_j \leq n$ stages.) Should it be necessary, excluding recipient(s) from notifications is possible by setting $t_{ij} = T_j$.

Considering the arrival of donation j as time 0, and letting (k) denote the index of the recipient with the k-th smallest notification time, we have

$$0 \leq t_{(1)j} \leq t_{(2)j} \leq \cdots \leq t_{(r_j)j}.$$

The priority ordering of eligible recipients for donation j implied by an arbitrary notification time vector $\underline{t}_j = (t_{1j}, t_{2j}, \ldots, t_{r_j j})$ is $\pi_j(\underline{t}_j) \equiv ((1), (2), \ldots, (r_j))$. This is the order in which they get notified: Recipient (1) gets the notification at time $t_{(1)j}$ (which is zero in the optimal solution, as a delay at the very beginning helps no one); recipient (2) at time $t_{(2)j}$ unless the donation is claimed by then, and so on. Let $\pi_j^k(\underline{t}_j) \equiv ((1), (2), \ldots, (k))$ be the first k recipients that get notified. These are the recipients that would be competing to claim the donation during *stage* k, which lasts from $t_{(k)j}$ to $t_{(k+1)j}$. For example, $\pi_j = (2, 3, 1)$ means that recipient 2 is notified first, 3 next, and 1 last; the first two to be notified are $\pi_j^2 = (2, 3)$, who are in play in stage 2, the time period $[t_{(2)j}, t_{(3)j}) = [t_{3j}, t_{1j})$.

When all the notification times for donation j are set to zero, i.e., $t_{ij} = 0$ for $i \in R_j$, then an n-stage priority list reduces to FCFS matching. There is another class of priority lists that especially interests *FR* for implementation purposes because of their simplicity. A *binary* or *2-stage priority list* has only two notification times: zero and some arbitrary time $t_B \in [0, T_j]$. By imposing the constraint that $t_{ij} \in \{0, t_B\}$, a binary priority list designates two subsets of eligible recipients to receive notifications: those who are notified immediately (they receive priority) and those who get a delayed notification at time t_B if the donation remains unclaimed by then.

2.3. A Rawlsian Notion of Fairness

FR is interested in maximizing Rawlsian welfare, as measured by the welfare of the worst-off recipient. Given their desire to preserve the self-organizing nature of their matching platform and the resulting allocations being irreversible, the problem is online in nature. We imagine a sequence of myopic decision problems; that is, we solve the priority list design problem for an arbitrary donation j, which boils down to optimizing a vector of notification times to act upon as soon as donation j arrives. The objective is to maximize the minimum welfare as measured by the value functions v_i of the eligible recipients – those affected by this donation.

For perishable donations, we impose a chance constraint that limits the probability of waste, which is a key efficiency metric for *FR*. That is, we require the probability that donation j is not claimed by its deadline (time T_j) be at most $\alpha \in (0,1)$ under any priority list. Equivalently, donation j must be claimed within T_j time units with a probability of at least $1-\alpha$. We refer to this constraint as the *waste constraint*. It is always satisfied for non-perishable donations as they are eventually claimed with probability one, i.e., $T_j = \infty$ implies $x_{0j} = 0$ for any priority list. If the waste constraint is impossible to satisfy – even when the donation is announced to all eligible recipients immediately – the priority list design problem has no feasible solution, and we simply default to the current FCFS system, which has the effect of minimizing x_{0j}.

Formally, the priority list design problem for donation j can be stated as follows:

$$z_j^* = \max_{\underline{t}_j \in [0,T_j]^{r_j}} \left\{ \min_{i \in R_j} \left\{ v_i(\underline{y}_i^{j-1}, x_{ij}(\underline{t}_j)) \right\} : \ x_{0j}(\underline{t}_j) \leq \alpha \right\} \tag{1}$$

Note that we explicitly indicate the dependence of fractional allocations x_{ij} and x_{0j} on the vector of notification times, $\underline{t}_j$, hence the priority list design. Note also that the minimization is over only those recipients eligible to receive donation j. The remaining recipients in $\mathcal{N} \setminus R_j$ are irrelevant in this problem: They are not eligible to receive the current donation and, for the valuation functions we consider, $v_i(\underline{y}_i^{j-1}, 0) = v_i(\underline{y}_i^{j-1})$.

3. Analysis of Fairness in Food Rescue

In this section we develop analytical characterizations of the optimal priority list and algorithms to compute the optimal priority list parameters – for both non-perishable and perishable donations and for both general (n-stage) and binary (2-stage) priority lists.

We first restate the general priority list design problem for a given (new) donation with a deadline T and a set of eligible recipients R, hereafter dropping the donation index j, as

$$z^* = \max_{\underline{t} \in [0,T]^r} \left\{ \min_{i \in R} \left\{ v_i(\underline{y}_i, x_i(\underline{t})) \right\} : \ x_0(\underline{t}) \leq \alpha \right\}, \tag{2}$$

where the notification times are the decision variables that determine the fractional allocations x_i of the current donation (with the vector $\underline{y}_i$ capturing all previous allocations to recipient i).

The problem can also be framed in reverse, with the fractional allocations $\underline{x} = (x_0, x_1, \ldots, x_r)$ as decision variables $x_i \in [0,1]$ that are further constrained to be *feasible* — in the sense that they result from some corresponding vector of notification times. With this reframing,

$$z^* = \max_{\underline{x} \in \mathcal{F}(T)} \left\{ \min_{i \in R} \left\{ v_i(\underline{y}_i, x_i) \right\} : x_0 \leq \alpha \right\}, \tag{3}$$

where $\mathcal{F}(T)$ is the set of fractional allocations for which there exists an n-stage priority list that achieves them by time T. By definition, $\mathcal{F}(T)$ is a subset of the r-dimensional probability simplex $\Delta_r \equiv \{\underline{x} \in [0,1]^{r+1} : \sum_{i=0}^{r} x_i = 1\}$. Of course, the entirety of Δ_r may or may not be achievable through priority lists, which is among the issues we settle in this section. Also, without loss of optimality, we set $x_0 = 0$ for a non-perishable donation, as a recipient would eventually claim it regardless of priority list design (as long as at least one recipient gets notified about it). Therefore, $\mathcal{F}(\infty) \subseteq \Delta_r^0$ where $\Delta_r^0 \equiv \{\underline{x} \in \Delta_r : \ x_0 = 0\}$.

We start in Section 3.1 by studying the priority list design problem for a non-perishable donation (so, $T = \infty$) to develop some important ideas before turning to the more general and practically relevant case of perishable donations in Section 3.2.

3.1. Non-perishable Donations

Our main goals in this section are to characterize the optimal priority list design within both n-stage and binary priority list classes and to show that it is computationally feasible to find the optimal priority list in either case. We find that *FR* does not give up any flexibility when it commits to using an n-stage priority list over alternative matching strategies, because any allocation is achievable. The same, however, is not true for binary priority lists, though they share a key structural property with n-stage priority lists. We show that the set of recipients that receive priority in the optimal binary priority list can be derived from the priority ordering used by the optimal n-stage priority list.

3.1.1. General n-stage priority lists for non-perishable donations. We start by introducing the concept of a *target allocation* $\hat{\underline{x}} \in \Delta_r^0$. Because the matching process, even when using priority lists, remains inherently an FCFS procedure that depends on agents' response rates, there is no guarantee that any particular target allocation is achievable. In particular, the priority order of an n-stage priority list impacts which target allocations are even feasible. We illustrate this with a simple example.

EXAMPLE 2. Take three recipients with response rates $\lambda_i = i$ for $i = 1, 2, 3$, and suppose the target allocation is $\hat{\underline{x}} = (0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Consider n-stage priority lists with priority ordering $\pi = (2, 3, 1)$. Being the last to get notified, recipient 1 cannot fare any better than when the donation is immediately announced to all three recipients. So, for any vector of notification times with this priority ordering, $x_1 \leq \frac{1}{6}$, implying that $\hat{\underline{x}}$ is not feasible. Consider, instead, $\pi = (1,2,3)$. Now $\hat{\underline{x}}$ is feasible and achieved with notification times $\underline{t} = (0, -\ln\frac{5}{6}, -\ln\frac{5}{6} - \frac{1}{3}\ln\frac{4}{5})$. To verify this, we analyze each stage separately. Stage 1 lasts $(-\ln\frac{5}{6})$ time units, so the probability that recipient 1 claims the donation in stage 1 is $\mathbb{P}(\tau_1 \leq -\ln\frac{5}{6}) = 1 - \exp\big(-1\cdot(-\ln\frac{5}{6})\big) = \frac{1}{6}$. Hence, stage 1's contribution to the fractional allocation vector is $(0, \frac{1}{6}, 0, 0)$. Using the memoryless property of the exponential distribution, recipient 1 or 2 claims the donation in stage 2 with probability $\mathbb{P}(\tau_1 \geq -\ln\frac{5}{6}) \cdot \mathbb{P}(\tau_{\{1,2\}} \leq -\frac{1}{3}\ln\frac{4}{5}) = (1-\frac{1}{6})\cdot(1 - \exp\big(-(1+2)(-\frac{1}{3}\ln\frac{4}{5})\big)) = \frac{5}{6}\cdot\frac{1}{5} = \frac{1}{6}$. Conditional on that happening, recipient 1 claims the donation with probability $\frac{\lambda_1}{\lambda_1+\lambda_2} = \frac{1}{3}$, and recipient 2 with probability $\frac{2}{3}$. As a result, stage 2's contribution to the fractional allocation vector is $(0, \frac{1}{18}, \frac{2}{18}, 0)$. Finally, the donation is claimed in stage 3 with the remaining probability $1 - \frac{1}{6} - \frac{1}{6} = \frac{2}{3}$, and it goes to each recipient with a conditional probability proportional to their response rate. We conclude that $\underline{t}$ results in the fractional allocation $\underline{x} = (0, \frac{1}{6} + \frac{1}{18} + \frac{2}{3}\cdot\frac{1}{6}, \frac{2}{18} + \frac{2}{3}\cdot\frac{2}{6}, \frac{2}{3}\cdot\frac{3}{6}) = (0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, which matches the target allocation. □

Intuitively speaking, in this example the key reason a priority list with $\pi = (1,2,3)$ achieves the uniform allocation with equal probabilities for all recipients is that they appear in the priority ordering from the slowest to the fastest response rate. This gives slower recipients a period of exclusive access to the donation with no competition from faster recipients. Theoretically, as we state in our first result, ordering recipients from the slowest to the fastest can always deliver the uniform allocation $\hat{\underline{x}} = (0, \frac{1}{r}, \ldots, \frac{1}{r})$. In fact, this is a special case of a more general result to follow.

Proposition 1. *The uniform target allocation $\hat{\underline{x}} = (0, \frac{1}{r}, \ldots, \frac{1}{r})$ of a non-perishable donation is always achievable by an n-stage priority list that prioritizes recipients from the slowest to the fastest, i.e., its priority ordering satisfies $\lambda_{(1)} \leq \lambda_{(2)} \leq \cdots \leq \lambda_{(r)}$.*

However, ordering recipients from the slowest to the fastest does not achieve all target allocations. For example, consider the recipients from Example 2 with $\pi = (1,2,3)$ and suppose that $\hat{x}_3 = \frac{2}{3}$. Recipient 3 has the greatest probability of receiving the donation when it is immediately announced to all recipients ($t_1 = t_2 = t_3 = 0$), but even then $x_3 = \frac{1}{2}$. So, achieving $\hat{x}_3 = \frac{2}{3}$ requires a priority ordering in which 3 does not appear last.

Recall that $\mathcal{F}(\infty)$ denotes the set of all fractional allocations of a non-perishable donation that can be achieved using an n-stage priority list. Let $\mathcal{F}^{\pi}(\infty)$ be the subset of $\mathcal{F}(\infty)$ that only uses n-stage priority lists with a particular priority ordering π. The following result characterizes this subset for a given priority ordering $\pi = ((1),(2),\ldots,(r))$, which is the full list of notifications in chronological order, while $\pi^k = ((1),(2),\ldots,(k))$ lists the first k recipients that get notified. We slightly overload the notation and use λ_{π^k} to denote $\lambda_{\{(1),(2),\ldots,(k)\}}$. All proofs appear in the appendix.

THEOREM 1. *For a non-perishable donation and a given priority ordering π, the set of fractional allocations that can be achieved by n-stage priority lists with priority ordering π is equal to the convex hull of r $(r+1)$-dimensional vectors Λ^k, for $k=1,\ldots,r$, with elements $\Lambda^k_0=0$, $\Lambda^k_{(\ell)}=\lambda_{(\ell)}/\lambda_{\pi^k}$ for $1\le\ell\le k$, and $\Lambda^k_{(\ell)}=0$ for $\ell>k$, i.e., $\mathcal{F}^\pi(\infty)=Conv[\Lambda^1,\ldots,\Lambda^r]$.*

For instance, in Example 2, the priority ordering $\pi=(1,2,3)$ can achieve any fractional allocation in $\mathcal{F}^{(1,2,3)}(\infty)=\mathrm{Conv}[(0,1,0,0),(0,\frac{1}{3},\frac{2}{3},0),(0,\frac{1}{6},\frac{2}{6},\frac{3}{6})]$. Clearly, $\hat{x}_3=\frac{2}{3}$ implies that $\underline{\hat{x}}\notin\mathcal{F}^{(1,2,3)}(\infty)$. On the other hand, using the priority ordering $\pi=(1,3,2)$ makes $\hat{x}_3=\frac{2}{3}$ achievable, because the feasible set becomes $\mathcal{F}^{(1,3,2)}(\infty)=\mathrm{Conv}[(0,1,0,0),(0,\frac{1}{4},0,\frac{3}{4}),(0,\frac{1}{6},\frac{2}{6},\frac{3}{6})]$.

These observations about the feasible set suggest that both a recipient's response rate and the magnitude of their target allocation must be considered when designing the optimal priority list. A core insight that keeps recurring in our paper is that the optimal priority list prioritizes recipients in decreasing order of $\hat{x}_i/\lambda_i$. That is, the implied priority ordering satisfies $\frac{\hat{x}_{(1)}}{\lambda_{(1)}}\ge\frac{\hat{x}_{(2)}}{\lambda_{(2)}}\ge\cdots\ge\frac{\hat{x}_{(r)}}{\lambda_{(r)}}$ and gives priority to recipients with a larger target allocation and a smaller response rate. Note that Proposition 1 is a direct corollary of this insight: When $\underline{\hat{x}}$ is uniform, recipients should be prioritized in decreasing order of $1/\lambda_i$, hence from the slowest to the fastest.

In the remainder of this subsection, we show that n-stage priority lists can achieve any target allocation $\underline{\hat{x}}$ of the current donation using this prioritization logic, which we then use to construct an algorithm that searches for the right target allocation and, in the process, computes the optimal n-stage priority list for a non-perishable donation. The following result precisely specifies how to achieve any given target allocation.

THEOREM 2. *Any target allocation $\underline{\hat{x}}\in\Delta^0_r$ of a non-perishable donation can be achieved by an n-stage priority list whose priority ordering satisfies $\frac{\hat{x}_{(1)}}{\lambda_{(1)}}\ge\frac{\hat{x}_{(2)}}{\lambda_{(2)}}\ge\cdots\ge\frac{\hat{x}_{(r)}}{\lambda_{(r)}}$, thus prioritizing recipients with a larger target allocation and slower average response. This priority list uses the following notification times: $t_{(1)}=0$ and*

$$t_{(k)}=t_{(k-1)}+\frac{1}{\lambda_{\pi^{k-1}}}\cdot\ln\left(\frac{1-\sum_{j=1}^{k-2}c_j}{1-\sum_{j=1}^{k-1}c_j}\right)\ \textit{for } k=2,\ldots,r,$$

where $c_k=\lambda_{\pi^k}\left[\frac{\hat{x}_{(k)}}{\lambda_{(k)}}-\frac{\hat{x}_{(k+1)}}{\lambda_{(k+1)}}\right]$ is the claim probability in stage $k=1,2,\ldots,r$, during which the top k recipients in priority (recipients $i\in\pi^k$) are competing to claim the donation, and $\frac{\hat{x}_{(r+1)}}{\lambda_{(r+1)}}=0$.

This result establishes the core idea in optimal design of n-stage priority lists: To achieve any target allocation $\underline{\hat{x}}\in\Delta^0_r$ for a set of r eligible recipients, the right priority ordering is to announce to them in decreasing order of $\hat{x}_i/\lambda_i$. Moreover, given this priority ordering, it is straightforward to compute the notification times $\underline{t}$ that yield $\underline{\hat{x}}$. All that remains to find is the right target allocation vector, i.e., to find the fractional allocation $\underline{x}^*$ that optimizes $\max_{\underline{x}\in\mathcal{F}(\infty)}\left\{\min_{i\in R}\left\{v_i(\underline{y}_i, x_i)\right\} : x_0 \le \alpha\right\}$, where $\mathcal{F}(\infty)$ can be replaced with Δ_r^0 due to Theorem 2.

Theorem 2 allows us to decouple the priority list design problem into two parts: (1) find the optimal target allocation; (2) find the notification times (and the implied priority ordering) that achieve the optimal target allocation. The second part is resolved by Theorem 2. As for the first part, Theorem 2 gives license to seek a target allocation in the entire simplex Δ_r^0, because it shows that any target allocation can be achieved by n-stage priority lists (as we establish later, the non-perishability assumption is crucial for this assertion to be true).

Our final result of this subsection is that the optimal target allocation x^* solving (3) can be found with a simple water-filling algorithm (Algorithm 1). The algorithm considers the eligible recipients in increasing order of value received so far from the lowest to the highest (from $\arg\min_{i\in R} v_i(\underline{y}_i)$ to $\arg\max_{i\in R} v_i(\underline{y}_i)$), and, for ease of exposition, relabels them from 1 to r in this order. First, it adds recipient 1 to the set of *active* recipients A, defined to be those that determine the objective function value. Then it computes how much the value received by 1 must increase to equal the value received by 2, and, assuming that this is possible, increases x_1 until the values of recipients 1 and 2 are equal. Now recipient 2 is added to the set of active recipients, the algorithm checks the increase required to reach the value of recipient 3, and raises the allocations of recipients 1 and 2 simultaneously. At some point it may not be possible to increase all the active recipients to match the next target value, in which case their values are simultaneously increased as much as possible until $\sum_{i\in R} x_i = 1$.

THEOREM 3. *The output of Algorithm 1, denoted by $\underline{x}^*$, is an optimal target allocation that solves the n-stage priority list design problem stated in (3) for a non-perishable donation (with $T = \infty$).*

To recap, the optimal n-stage priority list for a non-perishable donation can be found by using Algorithm 1 in conjunction with Theorem 2. Algorithm 1 provides the optimal target allocation; Theorem 2 provides the accompanying priority ordering of eligible recipients and their notification times. We close by illustrating the solution method in a small example.

EXAMPLE 3. Consider a problem instance with three recipients whose response rates are $\lambda_i = i$ for $i = 1, 2, 3$ (as in previous examples) and value functions are in the form of $v_i^{\mathrm{W}}(\cdot)$. Suppose the values received so far are $v_1^{\mathrm{W}}(\underline{y}_1) = 2, v_2^{\mathrm{W}}(\underline{y}_2) = 4$ and $v_3^{\mathrm{W}}(\underline{y}_3) = 8$. Let the current donation be of size $s = 6$. Initially $\underline{x} = \underline{0}$. First, 1's allocation (x_1) is increased until 1's value is equal to 2's. This happens at $x_1 = \frac{1}{3}$ because $v_1^{\mathrm{W}}(\underline{y}_1, \frac{1}{3}) = 2 + \frac{1}{3}\cdot 6 = 4 = v_2^{\mathrm{W}}(\underline{y}_2)$. Next, 1's and 2's allocations (x_1 and x_2) are increased simultaneously until 1's and 2's values reach $v_3^{\mathrm{W}}(\underline{y}_3)$ or the entire donation is allocated. Here the latter happens, thus the algorithm terminates with $x_1 = \frac{2}{3}$, $x_2 = \frac{1}{3}$, and $x_3 = 0$, which result in $v_1^{\mathrm{W}}(\underline{y}_1, x_1) = 2 + \frac{2}{3}\cdot 6 = 6$, $v_2^{\mathrm{W}}(\underline{y}_2, x_2) = 4 + \frac{1}{3}\cdot 6 = 6$, and $v_3^{\mathrm{W}}(\underline{y}_3, x_3) = 8 + 0 = 8$. According to Theorem 2, achieving the optimal target allocation $\underline{x}^* = (0, \frac{2}{3}, \frac{1}{3}, 0)$ requires using the

Algorithm 1: A water-filling algorithm to find the optimal target allocation, $\underline{x}^*$, the vector that solves the n-stage priority list design problem in (3) for a non-perishable donation

Data: Previous allocations $\underline{y}_i$ and value functions $v_i(\cdot)$ for all eligible recipients $i \in R$, which are indexed so that $v_1(\underline{y}_1) \le v_2(\underline{y}_2) \le \cdots \le v_r(\underline{y}_r)$.

$\underline{x} \leftarrow \underline{0}$; $i \leftarrow 1$;
$X \leftarrow 1$; `/* remaining fraction to allocate */`
$A \leftarrow \{1\}$; `/* set of active recipients, those with the minimum value */`
while $X > 0$ **do**
  $\delta_j \leftarrow \{\delta \ge 0:\ v_j(\underline{y}_j, x_j + \delta) = v_{i+1}(\underline{y}_{i+1})\}$ for each $j \in A$; `/* find what each active recipient needs to raise them to the next lowest value */`
  if $\sum_{j\in A} \delta_j \le X$ **then** `/* increase values of j in A to match i+1 */`
    $x_j \leftarrow x_j + \delta_j$ for $j \in A$;
    $X \leftarrow X - \sum_{j\in A} \delta_j$;
    $A \leftarrow A \cup \{i+1\}$;
    $i \leftarrow i+1$;
  else `/* increase values evenly with what remains */`
    $\underline{\gamma} = \arg\max_{\underline{\gamma}\ge 0}\{V:\ v_j(\underline{y}_j, x_j + \gamma_j) = V;\ \sum_{j\in A}\gamma_j = X\}$;
    $x_j \leftarrow x_j + \gamma_j$ for $j \in A$;
    $X \leftarrow 0$;
  return $\underline{x}$

priority ordering $\pi^* = (1,2,3)$, because $x_1^*/\lambda_1 = \frac{2}{3} > \frac{1}{6} = x_2^*/\lambda_2$ and $x_3^*/\lambda_3 = 0$. It also provides the optimal notification times $\underline{t}^* = (0, \ln 2, \infty)$, which result in claim probabilities of $c_1 = \frac{1}{2}$, $c_2 = \frac{1}{2}$, and $c_3 = 0$ in stages 1, 2, and 3, respectively. Thus, one can directly verify for recipients 1 and 2 that their probability of claiming the donation matches their optimal target allocation: $x_1^* = \frac{2}{3} = c_1 + c_2 \cdot \frac{\lambda_1}{\lambda_1+\lambda_2}$ and $x_2^* = \frac{1}{3} = c_2 \cdot \frac{\lambda_2}{\lambda_1+\lambda_2}$. $\square$

3.1.2. Binary priority lists for non-perishable donations. The binary (2-stage) priority list design problem is a special case of the general (n-stage) problem stated in (2) in that it is more constrained. It requires selecting a time $t_B \in [0,T]$ and constraining all the notification times to two possibilities $\{0, t_B\}$ – resulting in up to two *stages* or waves of notifications. The first wave is immediate notification at time zero, and the second a delayed notification at time $t_B \in (0,T)$. Thus the problem becomes:

$$z_B^* = \max_{\substack{t_B\in[0,T],\\ \underline{t}\in\{0,t_B\}^r}} \left\{ \min_{i\in R} \left\{ v_i(\underline{y}_i, x_i(\underline{t})) \right\} :\ x_0(\underline{t}) \le \alpha \right\}. \tag{4}$$

Let $R_1 \equiv \{i \in R:\ t_i = 0\}$ be the set of recipients to be notified immediately at the beginning of stage 1, which is the time range $[0, t_B)$. Similarly, let $R_2 \equiv \{i \in R:\ t_i = t_B\}$ be the set of recipients to be notified at time t_B, i.e., at the beginning of stage 2, which occurs over $[t_B, T)$. We call R_1 the *priority set* and R_2 the *non-priority set*.

In the space of fractional allocations the problem can be restated as follows:

$$z_B^* = \max_{\underline{x} \in \mathcal{F}_B(T)} \left\{ \min_{i \in R} \left\{ v_i(\underline{y}_i, x_i) \right\} : x_0 \leq \alpha \right\} \tag{5}$$

where $\mathcal{F}_B(T) \subseteq \Delta_r$ is the set of feasible fractional allocations for which there exists a binary priority list that achieves them by time T.

In this subsection we solve the binary priority list design problem – stated in (4) and (5) for any donation – for a non-perishable donation ($T = \infty$). Due to their simple and intuitive design, binary priority lists attracted more interest for a potential implementation at *FR*. But it is clear that there may be some performance loss compared to using n-stage priority lists (i.e., $z_B^* \leq z^*$), because, in contrast to n-stage priority lists, binary priority lists do restrict the space of feasible allocations even for non-perishable donations, as we illustrate next.

Example 4. Consider a problem instance with three recipients with response rates $\lambda_i = i$, for $i = 1, 2, 3$, and target allocation $\hat{\underline{x}} = (0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, which we know is achievable by an n-stage priority list (Example 2). The only hope of achieving $\hat{\underline{x}}$ is to include the slowest recipient, 1, in R_1. Suppose $R_1 = \{1\}$. Now the donation is simultaneously announced to 2 and 3, but since 3 is faster than 2 it is impossible for their allocations to be equal. If, instead, $R_1 \supset \{1\}$, then at least one of $\{2, 3\}$ will receive a larger fractional allocation than 1 will. Therefore, it must be that $\hat{\underline{x}} \notin \mathcal{F}_B$ is infeasible. □

This creates a dilemma. The optimal fractional allocation x^* (the output of Algorithm 1) may not be achievable by any binary priority list. This could mean, in the worst case, considering all possible binary priority lists, hence contending with an exponentially many subsets of R to pick the priority set R_1. We show that this is not necessary: Using insights from the previous subsection, we identify r nested subsets of eligible recipients, one of which must be optimal. Moreover, for a feasible target allocation $\hat{\underline{x}}$ and a fixed subset R_1, a linear program can be used to find the appropriate notification time t_B. These two results allow us to find the optimal binary priority list efficiently.

We first present our structural result about R_1 at any feasible solution.

Theorem 4. *Any feasible target allocation $\hat{\underline{x}} \in \mathcal{F}_B(\infty)$ of a non-perishable donation can be achieved by a binary priority list with a priority set $R_1 = \{(1), \ldots, (k)\}$ for some $k \in [r]$ and priority ordering $\pi = ((1), (2), \ldots, (r))$ that satisfies the condition $\frac{\hat{x}_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$.*

Therefore, binary priority lists share an important structural property with n-stage priority lists. Given a feasible target allocation $\hat{\underline{x}}$, the binary priority list that achieves it uses the same priority ordering that the n-stage priority list achieving $\hat{\underline{x}}$ would use: the one that prioritizes recipients in decreasing order of $\hat{x}_i/\lambda_i$. This greatly simplifies the search for optimal R_1: For a fixed $\underline{\hat{x}}$, it can only be one of r possibilities — the r prefixes of the priority ordering defined by $\hat{x}_i/\lambda_i$.

Next we show that, given a target allocation $\underline{\hat{x}}$ and a priority set R_1, checking if there is a notification time t_B resulting in a better allocation $\underline{x} \geq \underline{\hat{x}}$ is a matter of solving a simple linear program. The non-priority set $R_2 = R \setminus R_1$ is implied. Let $\gamma \in [0,1]$ be the probability that the donation is claimed in stage 1 – before it is open to all recipients in R at time t_B. Consider the linear program:

$$\max \quad \gamma \quad s.t. \quad \hat{x}_i \leq \lambda_i \left[\frac{\gamma}{\lambda_{R_1}} + \frac{1-\gamma}{\lambda_R} \right], \qquad \forall\, i \in R_1 \qquad \text{(Time-LP)}$$
$$\hat{x}_j \leq \lambda_j \cdot \frac{1-\gamma}{\lambda_R}, \qquad \forall\, j \in R_2$$
$$0 \leq \gamma \leq 1.$$

The first set of constraints require the probability of each recipient i in the priority set R_1 claiming the donation (the right-hand-side) to exceed her target allocation. The second set of constraints require the same for each recipient j in the non-priority set R_2. This linear program has a single decision variable and at most $r+2$ constraints.

If Time-LP is feasible, then the target allocation $\underline{\hat{x}}$ is achievable using a binary priority list with R_1 and $t_B = F_{R_1}^{-1}(\gamma^*)$, where γ^* is the optimal solution to Time-LP, and $F_{R_1}(t) = 1 - \exp(-\lambda_{R_1} t)$ is the probability that a recipient in R_1 claims the donation by time t. Thus, $t_B = -\ln(1-\gamma^*)/\lambda_{R_1}$ is the notification time for the recipients in R_2 that, with R_1, result in an allocation of at least $\underline{\hat{x}}$.

Finally, we describe our solution procedure, which combines the structure of the optimal priority set R_1 with Time-LP in a binary search to find the optimal binary list design. We first need to define an allocation vector tied to an objective function value. For any objective function value z, we define a *partial allocation* vector $\underline{\hat{x}}^z = (0, \hat{x}_1^z, \ldots, \hat{x}_r^z)$, where $\hat{x}_i^z = \min\{x \in [0,1] : \; v_i(\underline{y}_i, x) \geq z\}$ is the minimal fractional allocation that recipient i requires to attain value z, if it exists. A straightforward extension of Theorem 4 implies that, if $\underline{\hat{x}}^z \leq \underline{x}$ for any $\underline{x} \in \mathcal{F}_B(\infty)$, in other words if z is feasible, then z can be achieved by a binary priority list using as priority set some prefix of the priority ordering that satisfies $\frac{\hat{x}_{(1)}^z}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}^z}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}^z}{\lambda_{(r)}}$. If $v_i(\underline{y}_i, 1) < z$, recipient i could not attain value z even if the donation was given to recipient i with probability one and we set $\hat{x}_i^z = \infty$. Achieving z is trivially infeasible if $\sum_{i \in R} \hat{x}_i^z > 1$. When $\sum_{i \in R} \hat{x}_i^z \leq 1$, the existence of an allocation vector $\underline{x} \in F_B(\infty)$ with $\underline{x} \geq \underline{\hat{x}}^z$ depends on the existence of a feasible notification time t_B, which can be checked by solving Time-LP with partial allocation vector $\hat{x}^z$.

Putting it all together, we propose finding the optimal binary priority list by conducting a binary search on the objective function value z. The initial lower bound can be set to $\min_{i \in R} v_i(\underline{y}_i, 0)$, which is the current objective function value, and the upper bound to $\min_{i \in R} v_i(\underline{y}_i, 1)$.Each iteration of the binary search procedure requires finding $\underline{\hat{x}}^z$, and solving Time-LP for $\underline{\hat{x}}^z$ (if $\sum_{i \in R} \hat{x}_i^z \leq 1$) and each of the r prefixes of the priority ordering that puts recipients in decreasing order of $\hat{x}_i^z/\lambda_i$ (these are the only alternatives for R_1). Time-LP either outputs a feasible t_B for a given $(\underline{\hat{x}}^z, R_1)$ pair or concludes that it is infeasible. If Time-LP is feasible for at least one R_1 alternative, then $\underline{\hat{x}}^z$ is feasible and z increases in the next iteration of the binary search. On the other hand, if none of the r alternatives for R_1 is feasible for some $\underline{\hat{x}}^z$, then we conclude that z is not achievable by binary priority lists (i.e., $\underline{\hat{x}}^z \notin \mathcal{F}_B$) and reduce z in the next iteration of the binary search. We formally state the proposed solution procedure and its run-time complexity.

Theorem 5. *Consider a non-perishable donation (with $T = \infty$). Fix a constant $\epsilon > 0$, which can be arbitrarily close to zero. An ϵ-optimal binary priority list $(R_1^\circ, R_2^\circ, t_B^\circ)$ resulting in an objective function value z_B° that satisfies $|z_B^\circ - z_B^*| \leq \epsilon$ can be computed by executing $O(\ln(1/\epsilon))$ binary search iterations. Each iteration consists of a feasibility check taking time $O(r \cdot \mathcal{LP})$, where $\mathcal{LP}$ is the polynomial time to solve Time-LP.*

Example 5. Take Example 3; recall $v_1^{\mathrm{W}}(\underline{y}_1) = 2, v_2^{\mathrm{W}}(\underline{y}_2) = 4$, $v_3^{\mathrm{W}}(\underline{y}_3) = 8$, and $s = 6$. The optimal n-stage priority list $\underline{t}^* = (0, \ln 2, \infty)$ is already binary, and it results in the allocation $\underline{x}^* = (0, \frac{2}{3}, \frac{1}{3}, 0)$. Now, if one keeps everything the same except change the recipient 3's current value to $v_3^{\mathrm{W}}(\underline{y}_3) = 4$, the optimal n-stage priority list is no longer binary. The new optimal target allocation becomes $\underline{x}^* = (0, \frac{5}{9}, \frac{2}{9}, \frac{2}{9})$, and it requires using the notification times $t_1^* = 0$, $t_2^* = \ln(9/5) \approx 0.588$, and $t_3^* = t_2 + \ln(5/4)/3 \approx 0.662$, resulting in equal values $v_i^{\mathrm{W}}(\underline{y}_i, x_i^*) = 16/3$ for all three recipients. Contrast this with the ϵ-optimal binary priority list for $\epsilon = 0.01$. Applying the solution procedure developed in this subsection, we obtain $R_1^\circ = \{1\}$, $R_2^\circ = \{2, 3\}$, and $t_B^\circ = 0.560$. This binary priority list results in the allocation $(0.524, 0.190, 0.286)$ and the expected values $(5.143, 5.143, 5.714)$. $\square$

3.2. Perishable Donations

We now turn to the instances of the priority list design problem stated in (2) and (3) where donations are perishable, meaning that a new donation can only be claimed until a finite deadline $T < \infty$, either due to logistical constraints on the donor side or because the food is no longer safe to consume after time T. In this case, even if the donation is immediately announced to all recipients, there may be some probability of it being wasted (not claimed by the time it expires). An immediate observation is that perishability can make achieving fairness harder.

Example 6. Consider three recipients with response rates $\lambda_i = i$, for $i = 1, 2, 3$. Example 2 shows $\underline{\hat{x}} = (0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}) \in \mathcal{F}(\infty)$. Now suppose the donation has expiration time $T = 0.5$. The probability that it is claimed by time 0.5 if announced immediately to all recipients is $\mathbb{P}\{\tau_R \leq 0.5\} = 1 - \exp(-6/2) \approx 0.95$. It follows that the closest one can get to a uniform allocation, subject to the waste constraint with $\alpha = 0.05$, is $\underline{x} = (0.05, \frac{0.95}{6}, \frac{2 \cdot 0.95}{6}, \frac{3 \cdot 0.95}{6})$. $\square$

This example shows that perishability leads to a tension between fairness and efficiency (not wasting donations by successfully matching recipients to donors). When the deadline T is too short, or the eligible recipients have very slow response rates, it is possible that $F_R(T) < 1-\alpha$, meaning the waste constraint is infeasible even when the donation is immediately announced to all eligible recipients, hence surely infeasible when any priority list is deployed. In such cases, we do not use priority lists but simply maintain the status quo (FCFS allocation), which maximizes the probability of a successful match. So, in the remainder of this subsection we assume that $F_R(T) \geq 1-\alpha$ and show how to find the priority list that solves problem (2) or, equivalently, problem (3).

3.2.1. Binary priority lists for perishable donations. We first analyze the binary priority lists, because the key building block that gives us the optimal design turns out to be useful also when thinking about n-stage priority lists. The following structural result, which we call the *decomposition lemma*, allows us to obtain the priority ordering associated with the optimal binary list.

LEMMA 1. *Consider a stage of length $t < T$ with recipients in $S \subset R$ competing to claim a donation. Identify any pair of subsets of eligible recipients S_1 and S_2 such that $S_1 \subset S \subset S_2 \subseteq R$. One can then decompose this single stage into two new stages, the first with recipients S_1 and length $t_1 < t$, and the second with recipients S_2 and length $t_2 = t - t_1$, so that*

1. *the probability that the donation is claimed by time t remains unchanged; and*
2. *each recipient $i \in S_1$ is at least as likely to claim the donation by time t in the new two-stage plan as in the original one-stage plan.*

The first property ensures the decomposition does not lead to a violation of the waste constraint. One immediate implication is that if the original single stage of length t is embedded in a priority list design that is waste-constraint feasible, then property 1 in Lemma 1 together with the memoryless property of the exponential distribution guarantee that the new priority list design with a modified two-stage plan for the period of time in question remains waste-constraint feasible.

The second property is crucial to prove that the decomposition, under special conditions, does not make the objective function value any worse. After the decomposition, the allocation probabilities of recipients in S_1 weakly increase (by property 2 in Lemma 1); the allocation probabilities of recipients in $S_2 \setminus S$ increase (as they were not notified of the donation at all during $[0,t)$ in the original plan); and the allocation probabilities of recipients in $S \setminus S_1$ decrease. In order for the resulting decrease in the values of recipients in $S \setminus S_1$ not to affect the objective function value, we have to guarantee that there is some other recipient worse off than all $i \in S \setminus S_1$ in the modified plan, implying that the minimal value is not determined by any $i \in S \setminus S_1$. For this, we leverage the $\hat{x}_i/\lambda_i$ ordering, which, informally speaking, orders recipients by how long they need access to the donation (the ratio of their demand and rate of accumulating allocation probability) to reach objective function value z. We next illustrate this idea with an example.

Example 7. Consider three eligible recipients $R=\{1,2,3\}$ and a finite deadline T. Suppose a binary priority list with priority set $S=\{1,3\}$ and notification time $t<T$ for the non-priority set $R\setminus S=\{2\}$ achieves an objective function value z by the deadline T. According to Lemma 1, if we select $S_1=\{1\}$ and $S_2=R=\{1,2,3\}$, there exists a time $t_1<t$ that divides this single stage over $[0,t)$ into two stages. In the new plan, only recipient 1 is aware of the donation in period $[0,t_1)$, and all three recipients are aware of it in periods $[t_1,t)$ and $[t,T)$. Notice how this constitutes a valid binary priority list with priority set S_1 and notification time t_1 for the non-priority set $R\setminus S_1$. By Lemma 1, recipient $\{1\}=S_1$ is weakly better off by time t and the overall claim probability is unchanged. Since period $[0,t)$ affects period $[t,T)$ only in the conditional probability that the donation is unallocated by t, recipient 1 is weakly better off in the new priority list. Similarly, recipient $\{2\}=R\setminus S$ is better off, since they can now claim the donation in $[t_1,T)$ compared to $[t,T)$ originally, while accumulating as much probability in $[t,T)$ as before. However, recipient $\{3\}=S\setminus S_1$ is clearly worse off, as they can claim the donation in $[t_1,T)$ now, compared to $[0,T)$ before. Imagine a target allocation $\underline{\hat{x}}$ such that $\hat{x}_3/\lambda_3\le\hat{x}_2/\lambda_2$. Because recipients 2 and 3 are able to claim the donation in exactly the same stage of the new priority list, 3's value becomes at least as high as 2's, which exceeds z (since 2 is more likely to receive the donation than in the original priority list). So, the condition $\hat{x}_3/\lambda_3\le\hat{x}_2/\lambda_2$ has the desired implication that the new objective function value is at least z. □

The key to our main structural result on optimal binary priority lists for perishable donations is a condition that ensures recipients in $S\setminus S_1$ of Lemma 1 – like recipient 3 in Example 7 – are not so much worse off that they determine the objective function value. It turns out, this observation can be leveraged in general to establish a complete priority ordering for optimal binary priority list design for perishable donations: Prioritize eligible recipients in decreasing order of $\hat{x}_i/\lambda_i$. In Example 7 it enables us to detect that giving priority to recipient 3 is wasteful, which, together with Lemma 1, leads to a feasible priority list achieving the same or higher objective function value without prioritizing recipient 3. Building on this idea, we formally show that any feasible target allocation can be achieved by a binary priority list that prioritizes some prefix of the $\hat{x}_i/\lambda_i$ ordering.

Theorem 6. *Any feasible target allocation $\underline{\hat{x}}\in\mathcal{F}_B(T)$ of a perishable donation can be achieved by using a binary priority list with a priority set $R_1=\{(1),\dots,(k)\}$ for some $k\in[r]$ and priority ordering $\pi=((1),(2),\dots,(r))$ that satisfies the condition $\frac{\hat{x}_{(1)}}{\lambda_{(1)}}\ge\frac{\hat{x}_{(2)}}{\lambda_{(2)}}\ge\cdots\ge\frac{\hat{x}_{(r)}}{\lambda_{(r)}}$.*

Hence, using the $\hat{x}_i/\lambda_i$ ordering in binary priority list design, which is optimal for non-perishable donations (Theorem 4), remains optimal in the case of perishable donations. Given a target objective function value z, recall (from §3.1.2) that $\underline{\hat{x}}^z=(0,\hat{x}^z_1,\dots,\hat{x}^z_r)$ is the minimal partial allocation vector required to attain objective function value z. As before, a straightforward extension of Theorem 6 implies that, if $\underline{\hat{x}}^z$ is dominated by any $\underline{x} \in \mathcal{F}_B(T)$, then z can be achieved by a binary priority list using as priority set some prefix of the priority ordering satisfying $\frac{\hat{x}^z_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}^z_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}^z_{(r)}}{\lambda_{(r)}}$.

Taking advantage of this structural property, we now flesh out our solution procedure. It is again based on a binary search over the space of objective function values which, for objective function value z, checks if it can be achieved with a binary priority list design. The first part of this feasibility check comes from entertaining all r prefixes of the priority ordering satisfying $\frac{\hat{x}^z_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}^z_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}^z_{(r)}}{\lambda_{(r)}}$ as candidates for the priority set. The second part requires, for each prefix, identifying a feasible notification time t_B for the non-priority set or showing that none exists. Expressing the expected fractional allocations as a function of t_B, we obtain three expressions, each implying a bound on t_B:

$$
\begin{aligned}
x_0(t_B) &= \bar{F}_{R_1}(t_B) \cdot \bar{F}_R(T - t_B) \\
x_i(t_B) &= \frac{\lambda_i}{\lambda_{R_1}} \cdot F_{R_1}(t_B) + \frac{\lambda_i}{\lambda_R} \cdot \bar{F}_{R_1}(t_B) \cdot F_R(T - t_B) \quad \forall\, i \in R_1 \\
x_j(t_B) &= \frac{\lambda_j}{\lambda_R} \cdot \bar{F}_{R_1}(t_B) \cdot F_R(T - t_B) \quad \forall\, j \in R_2
\end{aligned}
$$

where $F_S(t)$ is the *cdf* of $Exp(\lambda_S)$ and $\bar{F}_S(t) = 1 - F_S(t)$ for $S \subseteq R$ and $t \geq 0$. The first bound comes from the waste constraint: Noting that $x_0(t_B)$ is monotonically increasing in t_B, and defining $t_0 \equiv \max\{t_B :\ x_0(t_B) \leq \alpha\}$, a feasible notification time must be smaller than t_0. The second bound comes from the recipients in the priority set. Because $x_i(t_B)$ is monotonically increasing for $i \in R_1$, a feasible notification time must be larger than $t_{LB} \equiv \min\{t_B :\ x_i(t_B) \geq \underline{\hat{x}}^z_i\ \forall i \in R_1\}$. The third bound comes from the recipients in the non-priority set. Because $x_j(t_B)$ is monotonically decreasing for $j \in R_2$, a feasible notification time must be smaller than $t_{UB} \equiv \max\{t_B :\ x_j(t_B) \geq \underline{\hat{x}}^z_j\ \forall j \in R_2\}$. Thus, the notification time to accompany a given (z, R_1) pair must belong to the set $\{t :\ t \leq t_0,\ t_{LB} \leq t \leq t_{UB}\}$, which is nonempty when $t_{LB} \leq \min(t_0, t_{UB})$. If this set is empty for all r alternatives for R_1, then achieving z is infeasible using binary priority lists, and z must be reduced in the next iteration of the binary search. On the other hand, if it is nonempty for some R_1 (at least one prefix), then z is feasible and raised in the next iteration.

We are ready to state our final result which concerns how a binary search over the objective function value space leads to a near-optimal binary priority list.

THEOREM 7. *Consider a perishable donation with deadline $T < \infty$. Fix a constant $\epsilon > 0$, which can be arbitrarily close to zero. An ϵ-optimal binary priority list $(R_1^\circ, R_2^\circ, t_B^\circ)$ resulting in an objective function value z_B° that satisfies $|z_B^\circ - z_B^*| \leq \epsilon$ can be computed by executing $O(\ln(1/\epsilon))$ binary search iterations. Each iteration involves computing the bounds t_0, t_{LB}, and t_{UB}, and verifying if the set of feasible notification times $\{t :\ t \leq t_0,\ t_{LB} \leq t \leq t_{UB}\}$ is empty or not at most r times.*

We conclude with an example.

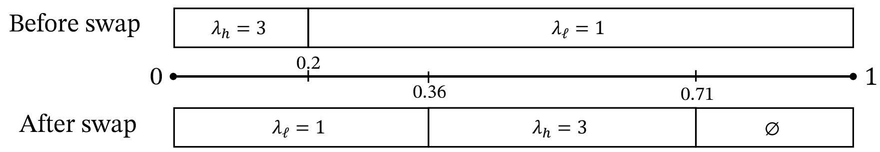

Figure 2 Example showing how stages can be sorted in increasing order of response rates.

EXAMPLE 8. Take Example 3; recall $v_1^{\mathrm{W}}(\underline{y}_1) = 2, v_2^{\mathrm{W}}(\underline{y}_2) = 4$, $v_3^{\mathrm{W}}(\underline{y}_3) = 8$, and $s = 6$. Set $T = 0.5$ and $\alpha = 0.15$. The ϵ-optimal binary priority list for $\epsilon = 0.01$ has $R_1^\circ = \{1\}$, $R_2^\circ = \{2,3\}$, and $t_B^\circ = 0.22$, resulting allocation $\underline{x}^* = (0.15, 0.307, 0.217, 0.326)$ and expected values $(3.839, 5.304, 9.957)$. Suppose one keeps everything the same except change recipient 1's current value to $v_1^{\mathrm{W}}(\underline{y}_1) = 4$. Now the ϵ-optimal binary priority list has $R_1^\circ = \{1\}$, $R_2^\circ = \{2,3\}$, and $t_B^\circ = 0.138$, resulting allocation $\underline{x}^* = (0.1, 0.257, 0.257, 0.386)$ and expected values $(5.544, 5.544, 10.316)$. $\square$

Building on the ideas developed for two stages, we now tackle the n-stage case. The decomposition lemma (Lemma 1) again plays an important role in proving the key structural result.

3.2.2. General n-stage priority lists for perishable donations. Finally, we consider the problem of finding the optimal n-stage priority list for a perishable donation subject to the waste constraint. We first establish an intuitive structural result, which we call the *sorting lemma*, that shows it is most efficient to sort stages in increasing order of their response rates (the sum of λ_i over the recipients competing in a stage).

LEMMA 2. *Consider a stage with high response rate λ_h, length s_h, and claim probability p_h, followed by a stage with low response rate λ_ℓ ($\lambda_\ell < \lambda_h$), length s_ℓ and claim probability p_ℓ. The two stages can be reordered and resized so that the low-response-rate stage occurs first, the new claim probabilities p'_ℓ and p'_h equal the old ones (i.e., $p'_\ell = p_\ell$, $p'_h = p_h$), and the new lengths s'_ℓ and s'_h are such that the total time required does not increase (i.e., $s'_\ell + s'_h \le s_\ell + s_h$).*

Next, we illustrate the sorting lemma in an example.

EXAMPLE 9. Consider a stage with rate $\lambda_h = 3$ and length $s_h = 0.2$, followed by a stage with rate $\lambda_\ell = 1$ and length $s_\ell = 0.8$. The respective allocation probabilities are $p_h = 0.45$ and $p_\ell = 0.30$. When the stages are reordered, the allocation probabilities can be maintained with $s'_\ell = 0.36$ and $s'_h = 0.35$. So, sorting as described decreases the total time required from 1 to 0.71 while keeping the allocation probabilities exactly the same. Figure 2 depicts this instance. $\square$

An immediate implication of Lemma 2 is that it can be applied to multiple stages: Sorting multiple stages in increasing order of total response rates would not change the probability of waste. This is because the probability of the donation not being allocated after stages $k = 1, \ldots, r$ is $\prod_{k=1}^{r}(1 - p_k)$. Since allocation probabilities of individual stages remain unchanged when sorting, the probability that the donation remains unclaimed after k stages is also unchanged.

By definition, n-stage priority lists sort stages in increasing order of total response rates by adding one recipient at a time to the set of recipients who are aware of a donation. Lemma 2 is useful in conjunction with Lemma 1. In particular, we use Lemma 1 to convert a stage, during which an arbitrary set of recipients $S \subseteq R$ are aware of the donation, into a sequence of substages of increasing rates culminating in $S' \supseteq S$ competing for the donation in the final substage. This can create a schedule in which a recipient competes for the donation during some substage but not in the next one, something priority lists do not allow by definition. Lemma 2 enables us to sort such a schedule into one that can be implemented using a priority list. Together, Lemmas 1 and 2 yield a constructive proof about the structure of the optimal n-stage priority list.

THEOREM 8. *Any feasible target allocation $\underline{\hat{x}} \in \mathcal{F}(T)$ of a perishable donation can be achieved by using an n-stage priority list with priority ordering $\pi = ((1), (2), \ldots, (r))$ that satisfies the condition $\frac{\hat{x}_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$.*

Theorem 8 is stated in terms of achieving a feasible target allocation but, as before, it could equivalently be stated in terms of achieving a particular feasible objective function value: any feasible objective function value z can be achieved using the priority ordering $\frac{\hat{x}^z_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}^z_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}^z_{(r)}}{\lambda_{(r)}}$. As an illustrative example, we show how any priority list not following the $\hat{x}_i/\lambda_i$ ordering can be transformed into one that does without hurting the objective function value or waste-constraint feasibility.

EXAMPLE 10. Consider the priority list in Figure 3 with $\pi = (3, 1, 4, 2)$. Suppose this list achieves some objective function value z via target allocation $\underline{\hat{x}}$, and the recipients are indexed such that $\frac{\hat{x}_1}{\lambda_1} \geq \frac{\hat{x}_2}{\lambda_2} \geq \frac{\hat{x}_3}{\lambda_3} \geq \frac{\hat{x}_4}{\lambda_4}$. For the first stage of this priority list, applying the decomposition lemma (Lemma 1) with $S = \{3\}$, $S_1 = \emptyset$ and $S_2 = \{1, 3\}$, we obtain two stages without lowering the objective function value or the waste probability. For the second stage, we use $S = \{1, 3\}$, $S_1 = \{1\}$ and $S_2 = \{1, 2, 3\}$. We apply such decompositions until the resulting stages are all prefixes of the $\hat{x}_i/\lambda_i$ ordering (see the third row). Note that we always pick S_1 as the biggest prefix in S, and include in S_2 the recipient who is next in the $\hat{x}_i/\lambda_i$ ordering. Then, at the final step, we apply the sorting lemma (Lemma 2) to obtain a new priority list, which obeys the $\hat{x}_i/\lambda_i$ ordering and is strictly better than what we started with (has an objective function value higher than z and the same waste probability). $\square$

Theorem 8 specifies the priority ordering for an n-stage priority list to achieve any feasible target objective function value z. Given a target objective function value z, Algorithm 2 acts as a feasibility check to find notification times (if any) achieving z. As before, this allows us to find an ϵ-optimal priority list using binary search.

Algorithm 2 immediately returns infeasible if it is obvious from $\underline{\hat{x}}^z$ that z can not be achieved (when $\sum_{i \in R} \hat{x}^z_i > 1$) or when it is impossible to satisfy the waste constraint even under FCFS with immediate notification of all recipients in R (when $\bar{F}_R(T) > \alpha$). If this does not prove infeasibility,

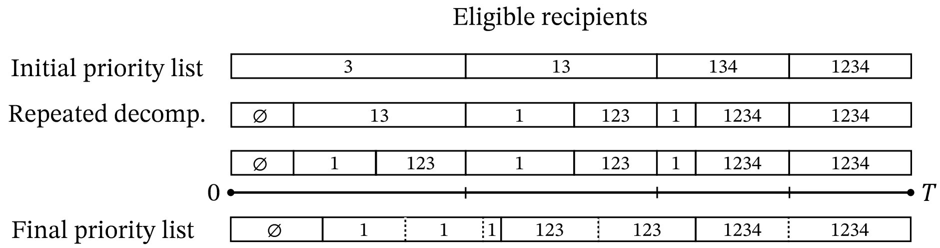

Figure 3 **Example showing how an initial priority ordering** $(3,1,4,2)$ **can be converted into another priority ordering** $(1,2,3,4)$ **while maintaining waste-constraint feasibility and raising the objective function value**

Algorithm 2: Feasibility check for a perishable donation using an n-stage priority list.

Data: Target objective function value z. Previous allocations $\underline{y}_i$ and value functions $v_i(\cdot)$ for all eligible recipients $i \in R$.

Result: Infeasible, or priority list $\underline{t}$ that yields objective function value z

1 Compute $\hat{x}_i^z = \min\{x \in [0,1]:\ v_i(\underline{y}_i, x) \geq z\}$ for all $i \in R$ or return Infeasible if no such $\hat{x}_i^z$ exists for some $i \in R$

2 **if** $\sum_{j\in R} \hat{x}_i^z > 1$ *or* $\exp(\lambda_R T) > \alpha$ **then**

3   **return** Infeasible

4 Relabel recipients if necessary so that $\frac{\hat{x}_1^z}{\lambda_1} \geq \frac{\hat{x}_2^z}{\lambda_2} \geq \cdots \geq \frac{\hat{x}_r^z}{\lambda_r}$;

5 $x_i \leftarrow 0$ for all $i \in R$; `/* probability of allocation to recipient` i `*/`

6 $t_1 \leftarrow 0$;

7 **for** i=*1,…,*r **do**

8   Find $s_{i,i+1}:\ (1-\sum_{j=1}^{i-1} x_j)\frac{\lambda_i}{\sum_{j=1}^{i}\lambda_j}\left[1-\exp\left(-\sum_{j=1}^{i}\lambda_i \cdot s_{i,i+1}\right)\right] + \frac{\lambda_i}{\lambda_{i+1}}\hat{x}_{i+1}^z = \hat{x}_i^z$

    ; `/* find` $s_{i,i+1}$ `so that` i `and` $i+1$ `reach` z `simultaneously */`

9   Find $s_{i,\alpha}:\ (1-\sum_{j=1}^{i-1} x_j)\exp\left(-\sum_{j=1}^{i}\lambda_j \cdot s_{i,\alpha}\right)\exp\left(-\sum_{j=1}^{r}\lambda_j(T-(t_i+s_{i,\alpha}))\right) = \alpha$

    ; `/* find` $s_{i,\alpha}$ `so that the waste constraint is not violated */`

10     $t_{i+1} \leftarrow t_i + \min\{s_{i,i+1}, s_{i,\alpha}\}$;

11     $\underline{x} \leftarrow \underline{x} + (1-\sum_{j=1}^{i-1} p_j)\cdot \Lambda^i\left[1-\exp\left(-\sum_{j=1}^{i}\lambda_j \cdot (t_{i+1}-t_i)\right)\right]$;

12 **if** $\min_{i\in R} v_i(\underline{y}_i, x_i) \geq z$ **then**

13   **return** $\underline{t}$

14 **return** Infeasible

---

recipients are relabeled in decreasing order of $\hat{x}_i^z/\lambda_i$ and their notification times are set one at a time in this order. The donation is announced to the highest-priority recipient immediately upon arrival ($t_1 = 0$). Then, for each recipient i in priority ordering, the algorithm computes two quantities to determine how long to wait before announcing to $i+1$. First, on line 8, it is determined how long to wait in order that recipients i and $i+1$ reach objective function value z at the same time

($s_{i,i+1}$). This leverages the fact that during any time period in which both i and $i+1$ are aware of the donation, their resulting expected fractional allocations differ only by the ratio of their response rates. Second, on line 9, $s_{i,\alpha}$ stores how long it is possible to wait before announcing the donation to the next recipient while still satisfying the waste constraint, assuming that the donation is then announced to all eligible recipients at time $t_i + s_{i,\alpha}$. The next notification time t_{i+1} is set to t_i plus $\min\{s_{i,i+1}, s_{i,\alpha}\}$ to ensure feasibility. Whenever a notification time is fixed, the probability that the donation is claimed in the corresponding stage is computed, and the algorithm maintains in $\underline{x}$ the total probability each recipient receives the item. Finally, the algorithm checks whether z is achieved.

THEOREM 9. *Consider a perishable donation with deadline $T < \infty$. Fix a constant $\epsilon > 0$, which can be arbitrarily close to zero. An ϵ-optimal n-stage priority list resulting in an objective function value z° that satisfies $|z^\circ - z^*| \leq \epsilon$ can be computed by executing $O(\ln(1/\epsilon))$ binary search iterations. Each iteration involves a feasibility check using Algorithm 2.*

We show the working of Theorem 9 on a small example.

EXAMPLE 11. Take Example 8; recall $v_1^{\mathrm{W}}(\underline{y}_1) = 2, v_2^{\mathrm{W}}(\underline{y}_2) = 4$, $v_3^{\mathrm{W}}(\underline{y}_3) = 8$, and $s = 6$. Set $T = 0.5$ and $\alpha = 0.15$. The ϵ-optimal n-stage priority list for $\epsilon = 0.01$ is binary, with $\underline{t} = (0, 0.22, 0.22)$ resulting allocation $\underline{x}^* = (0.15, 0.307, 0.217, 0.326)$ and expected values $(3.839, 5.304, 9.956)$. Suppose one keeps everything the same except change recipient 1's current value to $v_1^{\mathrm{W}}(\underline{y}_1) = 4$. Now the ϵ-optimal n-stage priority list has $\underline{t} = (0, 0.164, 0.258)$, resulting allocation $\underline{x}^* = (0.149, 0.303, 0.303, 0.245)$ and expected values $(5.815, 5.815, 9.469)$. $\square$

This concludes our development of structural results and algorithms to optimize priority lists for perishable donations. We now pivot to quantifying the potential impact of our ideas in practice.

4. Impact of Priority Lists on Fairness: A Simulation Study

We conduct simulations to estimate the impact of using the proposed priority lists on Rawlsian fairness – with the current FCFS allocations providing a benchmark. Our goal is two-fold: (1) to measure the improvement in fairness priority lists bring without any significant sacrifice in efficiency; and (2) to see how close binary priority lists are in performance to n-stage priority lists.

The simulation is calibrated with data provided by *FR* from their operations in Florida from May 2021 to September 2023. We focus on donations of at most 3,000 lbs (larger bulk donations are handled by a separate system) and small-to-medium-sized recipients who are largely constrained to picking up local donations (on average, within a 23-mile radius of their location). The historical data is summarized in Table 1. The average item weighs 170 lbs, is available for roughly 8 days and announced to 18 recipients. In this period 98.4% of donations were successfully matched to (mostly local) recipients; the average distance between donor and matched recipient is just under 16 miles.

	Min	Median	Mean	Max
Donation size (pounds)	3	75	169.6	2,500
Deadline (hours)	1.1	28	190.5	9,576
Eligible recipients	3	23	18.4	36
Recipients' maximum pickup range (miles)	9.3	31.1	37.0	155.4
Distance between matched donor and recipient (miles)	0.3	8.1	15.7	107.1

Table 1 Summary statistics of historical data from *FR*'s Florida operations (May 2021 - September 2023).

During the period under consideration, 48 unique organizations received donations through *FR*'s platform from 70 unique donors. We use these 48 recipients, 70 donors and their true locations and pickup radii in the simulations. Our simulations focus on perishable donations, since this aligns most closely to practice. The size, availability time, and donor location of every new donation in the simulation is sampled from the corresponding historical distributions.

We estimate historical response rates from the fraction of eligible donations received. More formally, the response time of agent i for every donation j is sampled from an exponential distribution with rate proportional to $\hat{\lambda}_i = \hat{m}_i / \sum_{j \in D} \mathbf{1}[i \in R_j]$, where $\hat{m}_i$ is the total number of donations received by recipient i in the data and the denominator is the total number of donations that recipient i was eligible for (as it fell within i's pickup radius) in the historical data D. Notice this reduces to the MLE for the simpler setting where every recipient has infinite pickup radius.

We compare three allocation schemes: the existing FCFS system, the optimal binary priority list (see §3.2.1), and the optimal n-stage priority list (see §3.2.2). Binary priority lists are included as they are perceived by *FR* to be significantly more practical to implement. We repeat the experiments for two valuation functions: maximizing the minimum number of donations (v^N) and the total weight of donations (v^{W}) received. We use $\alpha = 0.01$ for the waste constraint, allowing a decrease of at most 1% in successful allocations compared to the status quo of FCFS allocations. Each simulation consists of 1,000 donations. Results are aggregated over 16 repetitions.

4.1. Simulations from a Fresh Starting Point

The simulation results are summarized in Table 2. The first group of results in Table 2 summarize allocation characteristics: the decrease in the fraction of successful allocations due to use of priority lists, the average claim time, and the average of each donation's claim time divided by that donation's deadline. For both value functions, allocation times drastically increase due to priority lists delaying the announcement of each donation to rapid responders; from 4 hours on average for FCFS to 68 hours for binary priority lists and to 82 hours for n-stage priority lists. Despite this, we observe at most a 0.8% decrease in successful allocations under priority lists, and donations are not claimed particularly close to their deadlines (roughly by the midpoint of the availability periods).

	Number received			Pounds received		
	FCFS	BPL	n-PL	FCFS	BPL	n-PL
Change in % allocated	0.0	-0.5	-0.8	0.0	-0.6	0.0
Claim time (h)	4.2	67.7	82.0	4.1	76.5	83.9
Claim time / deadline	0.14	0.48	0.54	0.14	0.52	0.54
Δ_{obj}	0.01	0.075	0.077	1.31	11.58	10.95
Gini index	0.695	0.522	0.503	0.730	0.583	0.560
% of donations using priority list	0	60.5	60.4	0	60.3	60.3
Recipients on priority list	-	2.41	2.63	-	1.53	1.83
Priority period (h)	-	189.0	296.3	-	202.5	311.1
Deadline (h)	-	310.79	310.83	-	325.9	325.9

Table 2 **Comparison of FCFS against optimal priority lists (binary and n-stage) for two value functions v^{N} (number of donations received) and v^{W} (total pounds received) starting from a blank slate.**

The second group of results in Table 2 summarizes the fairness impact of the three allocation schemes. For each donation the objective is to maximize the minimum value delivered across all eligible recipients. For donation j, we measure the improvement in the objective function value as $\Delta^j_{obj} = \min_{i\in R_j} v_i(\underline{y}_i^{j-1}, x_{ij}) - \min_{i\in R_j} v_i(\underline{y}_i^{j-1}, 0)$. For v^N, Δ^j_{obj} is at most 1 when some recipient has received several fewer donations than the others, and at most $1/|R_j|$ when $\underline{y}_i^{j-1}$ are perfectly balanced. For v^{W}, Δ^j_{obj} is at most s_j for highly imbalanced allocations, and at most $s_j/|R_j|$ when $\underline{y}_i^{j-1}$ are perfectly balanced. The average improvement across donations is $\Delta_{obj} = \sum_{j=1}^{m} \frac{1}{m}\Delta^j_{obj}$. Table 2 reports the average Δ_{obj} across runs. Both priority lists are effective at improving the allocation of the worst-off recipient, improving over FCFS allocations by factors of 7-9. We also report the Gini index of the final allocations as an aggregate measure of inequity (recall that a Gini index of zero represents perfect equality). The n-stage priority lists show relative improvements of 23-27% over FCFS allocations in terms of Gini indexes, without significantly impacting the fraction of successfully allocated donations. Binary priority lists are slightly less effective, but still improve equitability of the allocations by 20-25% over FCFS allocations. Note that we do not explicitly optimize for Gini index; these improvements result indirectly from maximizing the worst-off recipient for each donation.

The final group of results in Table 2 summarize the characteristics of the priority lists and the donations where they are used. For both objective functions, priority lists are used on roughly 60% of donations, with one to three of the worst-off recipients receiving priority most of the time. The n-stage priority lists typically have *priority periods* (measured as the amount of time before all of R are notified of a donation) roughly 50% longer than binary priority lists and announce an unclaimed donation to all recipients within range approximately 14 hours before the donation becomes unavailable. This, together with the fact that donations are typically claimed within 4 hours once announced to all eligible recipients, is enough to ensure a high probability of successfully allocating

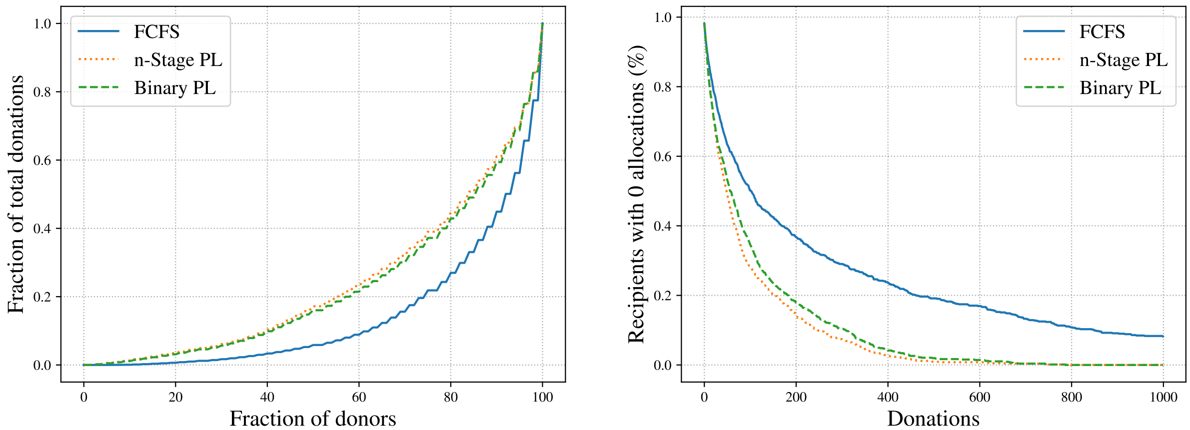

Figure 4 **Results for v^N. Left: the Lorentz curve where (x, y) represents an x fraction of recipients receiving a y fraction of donations. Right: the fraction of recipients who have not received any donations over time.**

each donation. Binary priority lists are significantly more conservative and announce unclaimed donations to all eligible recipients 4-5 days before the the end of the donation's availability period.

The FCFS and priority list allocations are further compared in Figures 4 and 5 for v^N; the results for v^W are similar and appear in the appendix. In the Lorentz-curve of allocations, visualized on the left panel of Figure 4, the point (x, y) indicates that the bottom x% of recipients received y% of donations. The priority list allocations comfortably dominate the FCFS allocation, for example, the bottom 60% of recipients receive more than 20% of donations, compared to approximately 10% under FCFS. The right panel of Figure 4 focuses on recipients who receive no donations and shows that priority lists are effective at improving the allocations of these worst-off recipients: After 700 donations, all recipients receive at least one donation under priority lists (and potentially more), while FCFS leaves some recipients without any donations throughout.

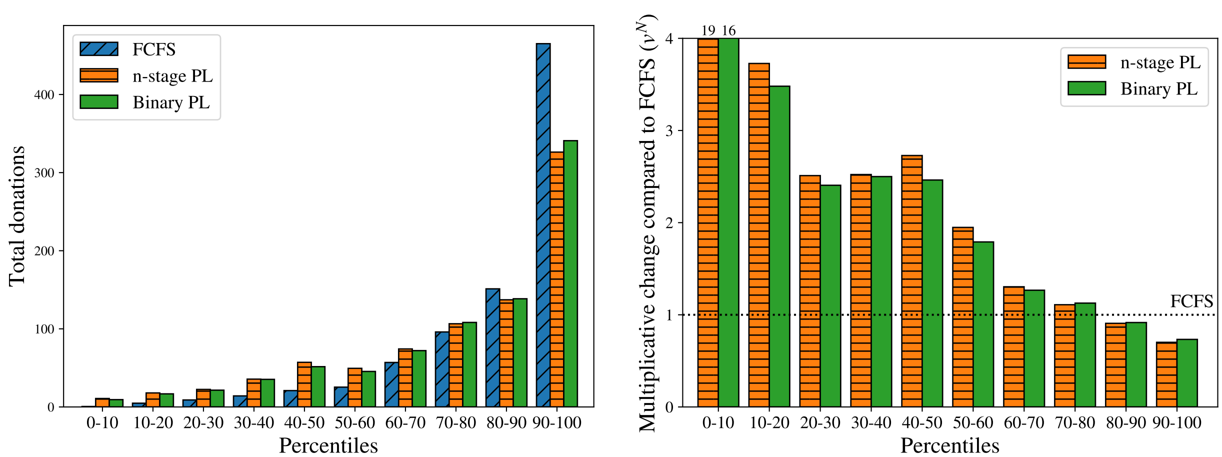

Figure 5 **Results for v^N. Left: Distribution of donations received per decile. Right: Multiplicative change in donations per decile compared to FCFS allocation.**

	Number received			Pounds received		
	FCFS	BPL	n-PL	FCFS	BPL	n-PL
Change in % allocated	0.0	-0.7	-0.2	0.0	-0.6	-0.9
Claim time (h)	4.8	89.2	91.4	4.8	98.8	107.2
Claim time / deadline	0.17	0.53	0.54	0.17	0.56	0.58
Δ_{obj} (avg)	0.01	0.05	0.06	5.55	8.1	9.53
Gini index	0.703	0.643	0.641	0.584	0.529	0.533
% of donations using priority list	0	55.7	55.7	0	55.7	55.7
Recipients on priority list	-	2.11	2.31	-	1.23	1.33
Priority period (h)	-	212.0	213.9	-	205.3	209.8
Deadline (h)	-	234.36	234.36	-	234.36	234.36

Table 3 Comparison of FCFS against optimal priority lists (binary and n-stage) for two value functions v^{N} (number of donations received) and v^{W} (total pounds received) using a historical snapshot as a starting point.

In Figure 5 the allocations are broken out by decile of recipients. The left panel shows the absolute number of donations received by recipients in each decile under each of the algorithms. The right panel shows the ratio of the number of donations received under each of the priority lists by that under FCFS allocation – a value of 1 means that recipients in that decile received exactly the same number of donations as under FCFS. Together with Figure 1(b), which shows the additive change by decile compared to FCFS, the figures tell a compelling tale: Under priority lists, donations are reallocated from the best-off 10-20% of recipients to the rest. From the right panel of Figure 5, we see that lower deciles benefit more from this redistribution, and the allocations going to each of the lowest five deciles (the bottom 50%) more than doubles.

4.2. Simulations Using a Historical Snapshot as a Starting Point

The previous simulations assume the system is starting afresh, that all recipients start with empty bundles. In practice, switching to a new allocation system will have to address any historical imbalances. This section attempts to evaluate the rate at which we may expect to see improvements, starting from an inequitable historical allocation. The simulations are designed to mimic a situation in which a new allocation algorithm is deployed on March 29 (six months before the end of the period captured in our data) and measure the performance of the algorithms over the following six months.

We follow the same methodology as in the previous subsection, with the following modifications. Response rates are calculated using historical allocations from May 2021 to March 2023. The starting allocation of the simulation is taken to be the historical allocation on March 28, 2023. Next, we simulate the allocation for each of the donations that arrive in the six-month period from March 29 to September 28, 2023. In other words, the first donation in the simulation is the first donation on or after March 29, including the size, donor, deadline, etc. Results are summarized in Table 3.

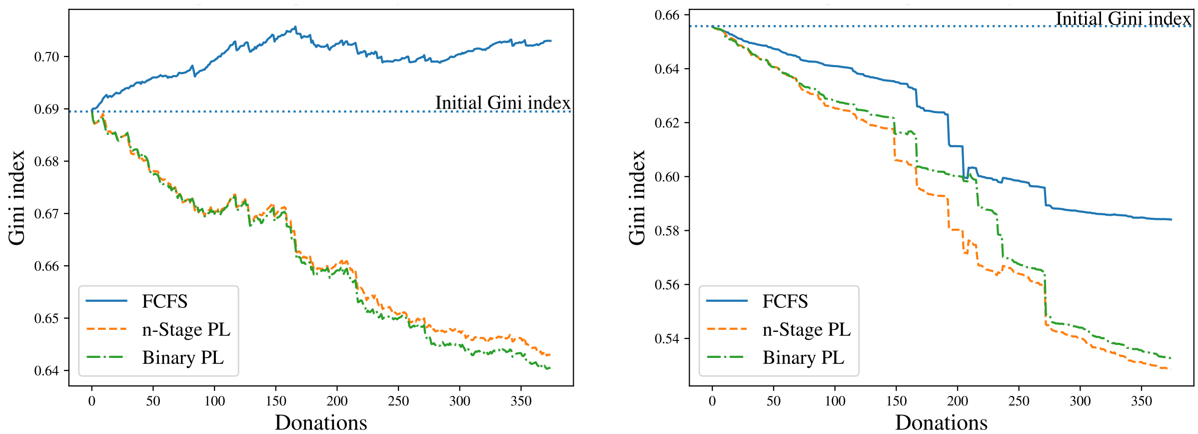

Figure 6 **Gini indexes over time for six months of historical donations for v^N (left) and v^{W} (right).**

Again, priority lists increase the fairness of the allocations without significantly decreasing the fraction of successful allocations. We highlight the main differences between the two sets of simulations. In terms of fairness, priority lists improve Δ_{obj} by a factor of 2–6. We observe smaller improvements in Gini index (roughly 8%) when starting from an inequitable allocation, compared to more than 20% before. This is unsurprising, since we are starting from an unbalanced allocation accumulated over multiple years and are trying to improve it using only six months worth of donations. As Figure 6 shows, the gap between the Gini indexes is consistently increasing over time; we expect this trend to continue on a longer time horizon. Binary and n-stage priority lists appear more similar in these simulations, both in terms of Gini indexes and the length and duration of the priority lists.

5. Conclusion

This paper is about improving fairness outcomes in the distribution of food donations through a food rescue platform, which has both social responsibility and environmental sustainability components. Fairness in distribution was the central research issue that came out of our collaboration with FoodRecovery.org, a national food rescue platform. *FR* runs a two-sided platform to match food donations with recipients on an FCFS basis. Thus, to the extent that there are structural differences among recipients that cause varying response rates, which there are, slow-responding recipients are at a constant disadvantage in the current FCFS regime. To nudge *FR*'s distributions towards more fair outcomes, we build an analytical model of food rescue and fully characterize an optimal policy that maximizes the minimum "value" (e.g., pounds of food or fraction of demand) delivered across recipients subject to an efficiency constraint that limits the probability of waste.

Our key tool to improve fairness in Rawlsian sense is a tiered notification system that gives disadvantaged organizations additional time to claim donations. We study two forms of this idea: an n-stage priority list, which controls the notification time for every recipient individually; and a two-stage or binary priority list, which has only two waves of notifications (and is much simpler to implement and administer). A key structural property of the optimal priority list design is to prioritize recipients that have received less in the past and those that were slower to claim donations. This basic insight is codified into an index which rank-orders the recipients for a given donation. Based on this structural property, we develop polynomial-time algorithms to find the optimal n-stage and binary priority lists for both perishable and non-perishable donations. We also conduct computational experiments, calibrated by field data from *FR*'s operations in Florida, to quantify the potential impact of priority lists. The results confirm that even simple, binary priority lists lead to significantly more fair allocations than the existing FCFS allocation rule, and that they perform almost as well as n-stage priority lists.

Food rescue platforms deserve further attention from OM scholars. For example, the merits of self-selection-based allocation schemes like *FR*'s FCFS matching over score-based allocation schemes that tend to exercise heavier control over allocations pose interesting tradeoffs. Food rescue platforms also struggle with how much and where to invest in the logistics of pickup and delivery of donations. Finally, it would be interesting to study food waste aspects more directly. *FR* currently has little visibility to how/where/when each donation is used. The general consensus is that, thanks to FCFS matching, donations are claimed only by those recipients who can use them in time. A deeper look into where food donations start and end may further minimize the chances of food waste.

References

Akkaş A (2019) Shelf space selection to control product expiration. *Production and Operations Management* 28(9):2184–2201.

Akkaş A, Gaur V (2022) OM forum — Reducing food waste: An operations management research agenda. *Manufacturing & Service Operations Management* 24(3):1261–1275.

Akkaş A, Honhon D (2022) Shipment policies for products with fixed shelf lives: Impact on profits and waste. *Manufacturing & Service Operations Management* 24(3):1611–1629.

Akkaş A, Honhon D (2023) Determining maximum shipping age requirements for shelf life and food waste management. *Production and Operations Management* 32(7):2173–2188.

Ata B, Lee D, Sönmez E (2019) Dynamic volunteer staffing in multicrop gleaning operations. *Operations Research* 67(2):295–314.

Banerjee S, Gkatzelis V, Hossain S, Jin B, Micha E, Shah N (2023) Proportionally fair online allocation of public goods with predictions. *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, 20–28.

Belavina E (2021) Grocery store density and food waste. *Manufacturing & Service Operations Management* 23(1):1–18.

Benadè G, Halpern D, Psomas A (2022a) Dynamic fair division with partial information. *Advances in Neural Information Processing Systems* 35:3703–3715.

Benadè G, Kazachkov A, Psomas A, Procaccia AD, Zeng D (2022b) Fair and efficient online allocations. *Operations Research* 72(4):1438–1452.

Benadè G, Kazachkov AM, Procaccia AD, Psomas CA (2018) How to make envy vanish over time. *EC'18: Proceedings of the 2018 ACM Conference on Economics and Computation*, 593–610.

Buzby JC, Farah-Wells H, Hyman J (2014) The estimated amount, value, and calories of postharvest food losses at the retail and consumer levels in the united states. *USDA-ERS Economic Information Bulletin* (121).

Kawase Y, Sumita H (2022) Online max-min fair allocation. *Algorithmic Game Theory: 15th International Symposium, SAGT 2022, Colchester, UK, September 12–15, 2022, Proceedings*, 526–543.

Lee MK, Kusbit D, Kahng A, Kim JT, Yuan X, Chan A, See D, Noothigattu R, Lee S, Psomas A, Procaccia AD (2019) Webuildai: Participatory framework for algorithmic governance. *Proc. ACM Hum.-Comput. Interact.* 3(CSCW).

Manshadi V, Niazadeh R, Rodilitz S (2023) Fair dynamic rationing. *Management Science* 69(11):6818–6836.

Manshadi V, Rodilitz S (2022) Online policies for efficient volunteer crowdsourcing. *Management Science* 68(9):6572–6590.

Orgut IS, Lodree EJ (2023) Equitable distribution of perishable items in a food bank supply chain. *Production and Operations Management* 32(10):3002–3021.

Prendergast C (2022) The allocation of food to food banks. *Journal of Political Economy* 130(8):1993–2017.

Rabbitt MP, Hales LJ, Burke MP, Coleman-Jensen A (2023) Household food security in the United States in 2022. *U.S. Department of Agriculture, Economic Research Service* (Report No. ERR-325).

ReFED (2024) Food waste is a solvable problem – here's how to do it. https://refed.org/food-waste/the-solutions/#key-action-areas, accessed 2024-02-20.

Shi ZR, Yuan Y, Lo K, Lizarondo L, Fang F (2020) Improving efficiency of volunteer-based food rescue operations. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 13369–13375.

Sinclair SR, Banerjee S, Yu CL (2022) Sequential fair allocation: Achieving the optimal envy-efficiency tradeoff curve. *ACM SIGMETRICS Performance Evaluation Review* 50(1):95–96.

United Nations (2015) Transforming our world: The 2030 agenda for sustainable development (a/res/70/1). https://sustainabledevelopment.un.org/content/documents/21252030%20Agenda%20for%20Sustainable%20Development%20web.pdf, accessed 2024-05-06.

Wu Q, Honhon D (2023) Don't waste that free lettuce! Impact of BOGOF promotions on retail profit and food waste. *Production and Operations Management* 32(2):501–523.

Zeng D, Psomas A (2020) Fairness-efficiency tradeoffs in dynamic fair division. *Proceedings of the 21st ACM Conference on Economics and Computation*, 911–912.

Zereyesus Y, Cardell L, Ajewole K, Farris J, Johnson M, Kee J, Valdes C, Zeng W (2023) International food security assessment, 2023–2033. *US Department of Agriculture, Economic Research Service* (GFA-34).

E-companion to "Achieving Rawlsian Justice in Food Rescue"

EC.1. Proofs: Non-perishable Donations, n-stage Priority Lists

THEOREM 1. *For a non-perishable donation and a given priority ordering π, the set of fractional allocations that can be achieved by n-stage priority lists with priority ordering π is equal to the convex hull of r $(r+1)$-dimensional vectors Λ^k, for $k=1,\dots,r$, with elements $\Lambda^k_0=0$, $\Lambda^k_{(\ell)}=\lambda_{(\ell)}/\lambda_{\pi^k}$ for $1\le\ell\le k$, and $\Lambda^k_{(\ell)}=0$ for $\ell>k$, i.e., $\mathcal{F}^\pi(\infty)=Conv[\Lambda^1,\dots,\Lambda^r]$.*

Proof of Theorem 1. We first show $\text{Conv}[\Lambda^1,\dots,\Lambda^r]\subseteq\mathcal{F}^\pi(\infty)$. Consider an arbitrary target allocation $\underline{\hat{x}}\in\text{Conv}[\Lambda^1,\dots,\Lambda^r]$. It is possible to write $\underline{\hat{x}}$ as a convex combination $\underline{\hat{x}}=\sum_{k=1}^r c_k\Lambda^k$ with $\sum_{k=1}^r c_k=1$. We may interpret c_k as the probability that the item is claimed/allocated in stage k. This gives a sequence of r equations:

$$
\begin{aligned}
c_1 &= 1-\exp\big(-\lambda_{(1)}t_{(2)}\big)\\
c_2 &= (1-c_1)(1-\exp\big(-(\lambda_{(1)}+\lambda_{(2)})(t_{(3)}-t_{(2)})\big)\\
c_3 &= (1-c_1-c_2)\left(1-\exp\big(-\lambda_{\pi^3}(t_{(4)}-t_{(3)})\big)\right)\\
&\vdots\\
c_r &= \left(1-\sum_{j=1}^{r-1}c_j\right)(1-\exp\big(-\lambda_R(t_{(r+1)}-t_{(r)})\big))=1-\sum_{j=1}^{r-1}c_j
\end{aligned}
$$

with unknowns $t_{(2)},\dots,t_{(r)}$, where the final transition follows from $t_{(r+1)}=\infty$. We can solve this system of equations to get $t_{(2)}=-\ln(1-c_1)/\lambda_{(1)}$ and, for $2\le k\le r-1$,

$$
t_{(k+1)}-t_{(k)}=\frac{1}{-\sum_{j=1}^k\lambda_{(k)}}\ln\left[1-\frac{c_k}{1-\sum_{j=1}^{k-1}c_j}\right]=\frac{1}{\lambda_{\pi^k}}\ln\left[\frac{1-\sum_{j=1}^{k-1}c_j}{1-\sum_j^k c_j}\right].
$$

It remains to check that these times are valid for a priority list, in particular, that $t_{(2)}\ge t_{(1)}=0$ and $t_{(k+1)}\ge t_{(k)}$ for all $k=2,\dots,r-1$. Clearly $t_{(2)}\ge0$, since $\ln(1-c_1)\le0$. Similarly, for $2\le k\le r-1$, $1-\sum_{j=1}^k c_j\le 1-\sum_{j=1}^{k-1}c_j$ implies that $t_{(k+1)}-t_{(k)}\ge0$. We conclude that $\underline{\hat{x}}$ is achievable by a priority list using priority ordering π, and $\text{Conv}[\Lambda^1,\dots,\Lambda^r]\subseteq\mathcal{F}^\pi(\infty)$.

We now show that $\mathcal{F}^\pi(\infty)\subseteq\text{Conv}[\Lambda^1,\dots,\Lambda^r]$. Consider an arbitrary $\underline{\hat{x}}\in\mathcal{F}^\pi(\infty)$, achieved by a priority list with priority ordering π and notification times $t_{(1)}\le t_{(2)}\le\cdots\le t_{(r)}$. Let c'_i be the probability that this priority list allocates the donation in stage i. Now we can write the probability of recipient (1) receiving the donation as

$$
\hat{x}_{(1)}=c'_1+\frac{\lambda_{(1)}}{\lambda_{(1)}+\lambda_{(2)}}c'_2+\cdots+\frac{\lambda_{(1)}}{\sum_{i=1}^r\lambda_{(i)}}c'_r=\sum_{i=1}^r\Lambda^i_{(1)}\cdot c'_i
$$

and similarly for any recipient (k)

$$\hat{x}_{(k)} = \sum_{i=k}^{r} \frac{\lambda_{(k)}}{\sum_{j=1}^{i} \lambda_{(j)}} c'_i = \sum_{i=k}^{r} \Lambda^i_{(k)} \cdot c'_i = \sum_{i=1}^{r} \Lambda^i_{(k)} \cdot c'_i,$$

where the final transition follows from the fact that $\Lambda^i_{(k)} = 0$ for $i < k$. It follows that $\underline{\hat{x}} \in \mathrm{Conv}[\Lambda^1, \ldots, \Lambda^r]$, implying $\mathcal{F}^\pi(\infty) \subseteq \mathrm{Conv}[\Lambda^1, \ldots, \Lambda^r]$. The result follows. $\square$

THEOREM 2. *Any target allocation $\underline{\hat{x}} \in \Delta^0_r$ of a non-perishable donation can be achieved by an n-stage priority list whose priority ordering satisfies $\frac{\hat{x}_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$, thus prioritizing recipients with a larger target allocation and slower average response. This priority list uses the following notification times: $t_{(1)} = 0$ and*

$$t_{(k)} = t_{(k-1)} + \frac{1}{\lambda_{\pi^{k-1}}} \cdot \ln\left(\frac{1 - \sum_{j=1}^{k-2} c_j}{1 - \sum_{j=1}^{k-1} c_j}\right) \text{ for } k = 2, \ldots, r,$$

where $c_k = \lambda_{\pi^k}\left[\frac{\hat{x}_{(k)}}{\lambda_{(k)}} - \frac{\hat{x}_{(k+1)}}{\lambda_{(k+1)}}\right]$ is the claim probability in stage $k = 1, 2, \ldots, r$, during which the top k recipients in priority (recipients $i \in \pi^k$) are competing to claim the donation, and $\frac{\hat{x}_{(r+1)}}{\lambda_{(r+1)}} = 0$.

Proof of Theorem 2. Let c_k be the probability that the donation is allocated during stage k and $c_{k,i}$ be the probability that the donation is allocated to recipient i during stage k for $i \in R$. Consider any stage k during which $\pi^k \subseteq R$ are aware of the donation and recall $t_i \leq t_{(k+1)}$ for all $i \in \pi^k$. For any two recipients $i, j \in \pi^k$,

$$c_{k,i} = \frac{\lambda_i}{\lambda_{\pi^k}} c_k = \frac{\lambda_i}{\lambda_j} \frac{\lambda_j}{\lambda_{\pi^k}} c_k = \frac{\lambda_i}{\lambda_j} c_{k,j}.$$

We apply this relationship to recipients (k) and $(k+1)$ to obtain

$$\hat{x}_{(k)} = c_{k,(k)} + \sum_{\ell=k+1}^{r} c_{\ell,(k)} = \frac{\lambda_{(k)}}{\lambda_{\pi^k}} c_k + \sum_{\ell=k+1}^{r} \frac{\lambda_{(k)}}{\lambda_{(k+1)}} c_{\ell,(k+1)} = \frac{\lambda_{(k)}}{\lambda_{\pi^k}} c_k + \frac{\lambda_{(k)}}{\lambda_{(k+1)}} \hat{x}_{(k+1)}$$
$$\Rightarrow \;\; c_k = \lambda_{\pi^k}\left[\frac{\hat{x}_{(k)}}{\lambda_{(k)}} - \frac{\hat{x}_{(k+1)}}{\lambda_{(k+1)}}\right]$$

for $1 \leq k < r$. And, for recipient (r), $c_r = \lambda_{\pi^r} \cdot \frac{\hat{x}_{(r)}}{\lambda_{(r)}} = \lambda_R \cdot \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$. Since $\frac{\hat{x}_{(k)}}{\lambda_{(k)}} \geq \frac{\hat{x}_{(k+1)}}{\lambda_{(k+1)}}$, $c_k \geq 0$, and $c_k = 0$ only when $\frac{\hat{x}_{(k)}}{\lambda_{(k)}} = \frac{\hat{x}_{(k+1)}}{\lambda_{(k+1)}}$, implying that (k) and $(k+1)$ should receive simultaneous access to the donation, or when $\hat{x}_{(k)} = 0$, implying that (k) and all subsequent recipients must not get access to the donation. It is easy to confirm that $\sum_{k=1}^{r} c_k = 1$. Given c_k for $1 \leq k \leq r$, we can compute $t_{(k)}$ as follows. Write $\mathcal{E}_k$ for the event that the donation is claimed in stage k. Now

$$c_k = \mathbb{P}[\mathcal{E}_k] = \mathbb{P}[\mathcal{E}_k \text{ and none of } \mathcal{E}_1, \ldots, \mathcal{E}_{k-1} \text{ happened}]$$
$$= (1 - \sum_{j=1}^{k-1} \mathbb{P}[\mathcal{E}_j]) \cdot \mathbb{P}[\mathcal{E}_k | \text{none of } \mathcal{E}_1, \ldots, \mathcal{E}_{k-1} \text{ happened}]$$

$$= \left(1 - \sum_{j=1}^{k-1} c_j\right) \cdot \mathbb{P}\left[\tau_{\pi^k} \le t_{(k+1)} - t_{(k)}\right] = \left(1 - \sum_{j=1}^{k-1} c_j\right)\left(1 - \exp\left[-\lambda_{\pi^k}(t_{(k+1)} - t_{(k)})\right]\right).$$

Solving for $t_{(k+1)} - t_{(k)}$ yields

$$t_{(k+1)} - t_{(k)} = -\frac{1}{\lambda_{\pi^k}} \ln\left(1 - \frac{c_k}{\left(1 - \sum_{j=1}^{k-1} c_j\right)}\right) = \frac{1}{\lambda_{\pi^k}} \ln\left(\frac{1 - \sum_{j=1}^{k-1} c_j}{1 - \sum_{j=1}^{k} c_j}\right).$$

Notice $t_{(k+1)} - t_{(k)} > 0$ when $c_k > 0$, which holds if $\frac{\hat{x}_{(k)}}{\lambda_{(k)}} > \frac{\hat{x}_{(k+1)}}{\lambda_{(k+1)}}$. When $\frac{\hat{x}_{(k)}}{\lambda_{(k)}} = \frac{\hat{x}_{(k+1)}}{\lambda_{(k+1)}}$, $t_{(k+1)} - t_{(k)} = 0$ so that (k) and $(k+1)$ get simultaneous access to the donation. When $\hat{x}_{(k)} > 0$ and $\hat{x}_{(k+1)} = 0$, implying $\hat{x}_{(j)} = 0$ for $j > k+1$, then $c_j = 0$ for all $j > k$. It follows that $\sum_{j=1}^{k} c_j = 1$ and $t_{(k+1)} - t_{(k)} = \infty$, so the donation is never announced to recipients $(k+1), \dots, (r)$. Finally, when both $\hat{x}_{(k-1)} = 0$ and $\hat{x}_{(k)} = 0$, $t_{(k)} = \infty$ by convention since there must be some $j < k-1$ for which $t_{(j)} = \infty$ and neither $(k-1)$ nor (k) will get access to the donation. We conclude the notification times are valid, i.e., $0 = t_{(1)} \le t_{(2)} \le \dots \le t_{(r)}$, and constitute an n-stage priority list with priority ordering π. $\square$

Theorem 3. *The output of Algorithm 1, denoted by $\underline{x}^*$, is an optimal target allocation that solves the n-stage priority list design problem stated in (3) for a non-perishable donation (with $T = \infty$).*

Given fractional allocation $\underline{x}$, let $z(\underline{x}) = \min_{i \in R} v_i(\underline{y}_i, x_i)$. Before proving Theorem 3, we establish an invariant regarding the values of the active recipients in A.

Lemma EC.1. *At the end of every iteration of Algorithm 1, $v_i(\underline{y}_i, x_i) = z(\underline{x})$ for all $i \in A$.*

Proof of Lemma EC.1. Let A^ℓ denote A at the end of iteration ℓ of the while-loop. Similarly, use $\underline{x}^\ell$, X^ℓ to refer to $\underline{x}$ and X, respectively, at that point. We prove by induction that at the end of iteration k, $v_i(\underline{y}_i, x_i^k) = z(\underline{x}^k)$ for all $i \in A^k$. As base case for the induction, consider iteration 1 of Algorithm 1. If $A^1 = \{1\}$, it means that $\delta_1 > 1$ so that the else-block is activated, and x_1^1 is incremented by γ_1 as in line 13, from which it follows that $v_1(\underline{y}_1, x_1^1) = V < v_2(\underline{y}_2)$. Recall that recipients are labelled in order of increasing values. It follows that $z(\underline{x}^1) = v_1(\underline{y}_1, x_1^1)$. Suppose instead that $A^1 = \{1, 2\}$. Now, $x_1^1 = \delta_1 < 1$ after the update in line 7, so that $v_1(\underline{y}_1, x_1^1) = v_2(\underline{y}_2)$. The required condition follows after observing that initially 1 had the smallest value, 2 the second smallest value, and $x_j^1 = 0$ for all $j > 1$.

Now, suppose the induction hypothesis holds for all iterations up to $k-1$. At the end of iteration $k-1$ (equivalently, the start of iteration k), $x_j^{k-1} = 0$ for all $j \ge k$ and (by the induction hypothesis) $v_i(\underline{y}_i, x_i^{k-1}) = v_j(\underline{y}_j, x_j^{k-1})$ for all $i, j \in A^{k-1}$. The active set A^{k-1} takes one of two possible forms: It either includes k or not, depending on the most recent result of the 'if' statement in line 6. If $k \notin A^{k-1}$, then the previous iteration executed the else-block, so $X^{k-1} = 0$, no further updates are made, the algorithm terminates with $A = A^{k-1}$ and $\underline{x} = \underline{x}^{k-1}$, and the induction hypothesis continues to hold.

Suppose instead that $k \in A^{k-1}$. Now $v_i(\underline{y}_i, x_i^{k-1}) = v_j(\underline{y}_j, x_j^{k-1})$ for all $i, j \in A^{k-1} = \{1, \dots, k\}$. If the if statement evaluates to True, then X^{k-1} is large enough to increase the allocation of all recipients in A^{k-1} until their values match $v_{k+1}(\underline{y}_{k+1})$. After the corresponding value update on line 7, at the end of the iteration, $v_j(\underline{y}_j, x_j^k) = v_{k+1}(\underline{y}_{k+1})$ for all $j \in A^{k-1}$. Values remain unchanged for the remainder of the iteration, and $A^k = A^{k-1} \cup \{k+1\}$. It follows, since recipients are ordered by increasing values, that $z(\underline{x}^k) = v_j(\underline{y}_j, x_j^k)$ for all $j \in A^k$ at the end of iteration k. Alternatively, the if-statement evaluates to False. Values are updated on line 13, and for arbitrary $j \in A^{k-1}$, $v_j(\underline{y}_j, x_j^k) = V$, that is, the values of all recipients in A^k increase by the same constant. No further updates are made to $\underline{x}^k$ or $A^k = A^{k-1}$. The values of all recipients in A^{k-1} start equal (by the induction hypothesis) and increase by the same constant, so their values at the end of iteration k remain equal. Moreover, $v_j(\underline{y}_j, x_j^k) < v_{k+1}(\underline{y}_{k+1}, x_{k+1}^k)$ for any $j \in A^k$. It follows that $z(\underline{x}^k) = v_j(\underline{y}_j, x_j^k)$ for all $j \in A^k$. □

We now show that Algorithm 1 returns an optimal fractional allocation.

Proof of Theorem 3. Consider output $\underline{x}^*$ with objective function value $z^* = z(\underline{x}^*)$. Denote with $A^* = \{i \in R : v_i(\underline{y}_i, x_i^*) = z^*\}$ the set of active recipients under $\underline{x}^*$ (those with values equal to z^*). Suppose for contradiction there exists an alternative fractional allocation $\underline{x}' \in \Delta_n^0$ with objective function value $z' = z(\underline{x}') > z^*$. By the monotonicity of the valuation functions, it follows that $x_i' > x_i^*$ for all $i \in A^*$, i.e., active recipients must receive strictly higher allocations in $\underline{x}'$ than in $\underline{x}^*$. Since $\sum_{i \in R} x_i' = 1 = \sum_{i \in R} x_i^*$, this implies there is some recipient $j \in R \setminus A^*$ for which $x_j' < x_j^*$. Since $x' \geq 0$, it follows that inactive recipient j has $x_j^* > 0$ and (since it is inactive) $v_j(\underline{y}_j, x_j^*) > z^*$. However, Algorithm 1 only increases the assignments of the active recipients in the set A, so at some point during execution this recipient j must have been added to A. By Lemma EC.1 and the fact that recipients are never removed from the set of active recipients, we conclude that $j \in A$ at the termination of the algorithm and thus $v_j(\underline{y}_j, x_j^*) = z^*$, contradicting the selection of $j \in R \setminus A^*$. It follows that x^* is optimal. □

EC.2. Proofs: Non-perishable Donations, Binary Priority Lists

THEOREM 4. *Any feasible target allocation $\hat{\underline{x}} \in \mathcal{F}_B(\infty)$ of a non-perishable donation can be achieved by a binary priority list with a priority set $R_1 = \{(1), \dots, (k)\}$ for some $k \in [r]$ and priority ordering $\pi = ((1), (2), \dots, (r))$ that satisfies the condition $\frac{\hat{x}_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$.*

Proof of Theorem 4. For ease of exposition, relabel recipients so that $\frac{\hat{x}_1}{\lambda_1} \geq \frac{\hat{x}_2}{\lambda_2} \geq \cdots \geq \frac{\hat{x}_r}{\lambda_r}$. After relabelling, the theorem statement is that the target allocation $\hat{\underline{x}}$ can be achieved by a binary priority list with priority set $[k]$ for some $k \in [r]$. Let (R_1, t_B) be the binary priority list that achieves the target allocation $\hat{\underline{x}}$ using the priority set R_1, the non-priority set $R_2 = R \setminus R_1$, and the notification time t_B for the recipients in R_2. Without loss of generality, we assume that $\emptyset \subset R_1 \subset R$ and $t_B > 0$. Recall that $F_{R_1}(t_B) = \mathbb{P}[\tau_{R_1} \leq t_B] = 1 - \exp(-\lambda_{R_1} t_B)$ is the probability the donation is claimed before time t_B. If $i \in R_1$, i receives the donation with probability $\frac{\lambda_i}{\lambda_{R_1}}$ during $[0, t_B)$ conditional on a claim by time t_B, or with probability $\frac{\lambda_i}{\lambda_R}$ during $[t_B, \infty)$ conditional on no claims by time t_B. If $i \in R_2$, i receives the donation with probability $\frac{\lambda_i}{\lambda_R}$ conditional on no claims by time t_B. Summarizing,

$$\hat{x}_i = \begin{cases} \lambda_i \left[\frac{F_{R_1}(t_B)}{\lambda_{R_1}} + \frac{1-F_{R_1}(t_B)}{\lambda_R}\right], & i \in R_1 \\ \lambda_i \cdot \frac{1-F_{R_1}(t_B)}{\lambda_R}, & i \in R_2. \end{cases}$$

Notice that $\frac{\hat{x}_i}{\lambda_i} = \frac{\hat{x}_j}{\lambda_j}$ for all $i, j \in R_1$, and similarly for all $i, j \in R_2$.

Suppose that there exists a pair of recipients ℓ and k that violate the condition in Theorem 4, i.e., $\ell \in R_1$, $k \in R_2$, and $\ell > k$. Consider a binary priority list (R_1', t_B') with the priority set $R_1' = R_1 \setminus \{\ell\}$, the non-priority set $R_2' = R_2 \cup \{\ell\}$, and the notification time t_B' for the recipients in R_2'. We argue that it is possible to choose t_B' so that it generates a contradiction.

If $R_1' = \emptyset$, set $t_B' = 0$ (i.e., don't use a priority list). For $i \neq \ell$, in other words for all $i \in R_2 = R \setminus \{\ell\}$, $x_i' = \lambda_i/\lambda_R \geq \lambda_i(1 - F_{R_1}(t_B))/\lambda_R = \hat{x}_i$. For recipients ℓ and k, $\frac{x_\ell'}{\lambda_\ell} = \frac{x_k'}{\lambda_k}$ since $\ell, k \in R_2'$, and $\frac{\hat{x}_k}{\lambda_k} \geq \frac{\hat{x}_\ell}{\lambda_\ell}$ because $k < \ell$. Putting it all together,

$$\frac{x_\ell'}{\lambda_\ell} = \frac{x_k'}{\lambda_k} \geq \frac{\hat{x}_k}{\lambda_k} \geq \frac{\hat{x}_\ell}{\lambda_\ell},$$

which implies $x_\ell' \geq \hat{x}_\ell$. So, all recipients obtain higher allocations under the binary priority list (R_1', t_B'). In particular, ℓ goes from having some priority to no priority in the modified binary priority list, and yet improves its allocation. Writing out $x_\ell' \geq \hat{x}_\ell$ explicitly,

$$\frac{\lambda_\ell}{\lambda_R} \geq \lambda_\ell \left[\frac{F_{R_1}(t_B)}{\lambda_{R_1}} + \frac{1 - F_{R_1}(t_B)}{\lambda_R}\right],$$

it holds if and only if $\lambda_{R_1} \geq \lambda_R$ (recall that $t_B > 0$ by assumption). This contradicts $R_1 \subset R$.

Now, assume $R_1' \neq \emptyset$, and choose t_B' so that $F_{R_1'}(t_B')/\lambda_{R_1'} = F_{R_1}(t_B)/\lambda_{R_1}$. This implies $F_{R_1'}(t_B') < F_{R_1}(t_B)$ and $1 - F_{R_1'}(t_B') > 1 - F_{R_1}(t_B)$. Let $\underline{x}'$ be the resulting allocation vector. By the same reasoning as before

$$x_i' = \begin{cases} \lambda_i \left[\frac{F_{R_1'}(t_B')}{\lambda_{R_1'}} + \frac{1-F_{R_1'}(t_B')}{\lambda_R}\right] = \lambda_i \left[\frac{F_{R_1}(t_B)}{\lambda_{R_1}} + \frac{1-F_{R_1'}(t_B')}{\lambda_R}\right] > \lambda_i \left[\frac{F_{R_1}(t_B)}{\lambda_{R_1}} + \frac{1-F_{R_1}(t_B)}{\lambda_R}\right], & i \in R_1' \\ \lambda_i \cdot \frac{1-F_{R_1'}(t_B')}{\lambda_R} > \lambda_i \cdot \frac{1-F_{R_1}(t_B)}{\lambda_R} & i \in R_2' \end{cases}.$$

Thus, $x_i' > x_i$ for $i \in R_1' = R_1 \cap R_1'$ and $i \in R_2' \setminus \{\ell\} = R_2 \cap R_2'$; that is, those recipients that receive priority or do not receive priority in both binary priority lists are strictly better off. Even recipient ℓ is better off under the binary priority list (R_1', t_B') because by a similar reasoning as before,

$$\frac{x_\ell'}{\lambda_\ell} = \frac{x_k'}{\lambda_k} \geq \frac{\hat{x}_k}{\lambda_k} \geq \frac{\hat{x}_\ell}{\lambda_\ell},$$

where the equality follows from $k, \ell \in R_2'$ and the final inequality from $k < \ell$. This implies $x_\ell' \geq \hat{x}_\ell$, which is a contradiction. Writing out $x_\ell' \geq \hat{x}_\ell$ explicitly,

$$\lambda_\ell \cdot \frac{1 - F_{R_1'}(t_B')}{\lambda_R} \geq \lambda_\ell \left[\frac{F_{R_1}(t_B)}{\lambda_{R_1}} + \frac{1 - F_{R_1}(t_B)}{\lambda_R} \right],$$

it contradicts the facts that $F_{R_1}(t_B) > F_{R_1'}(t_B')$ and $\lambda_R > \lambda_{R_1}$. Hence, by contradiction, there must not exist any such pair of recipients ℓ and k in the binary priority list (R_1, t_B) that violate the condition in Theorem 4. $\square$

THEOREM 5. *Consider a non-perishable donation (with $T = \infty$). Fix a constant $\epsilon > 0$, which can be arbitrarily close to zero. An ϵ-optimal binary priority list $(R_1^\circ, R_2^\circ, t_B^\circ)$ resulting in an objective function value z_B° that satisfies $|z_B^\circ - z_B^*| \leq \epsilon$ can be computed by executing $O(\ln(1/\epsilon))$ binary search iterations. Each iteration consists of a feasibility check taking time $O(r \cdot \mathcal{LP})$, where $\mathcal{LP}$ is the polynomial time to solve Time-LP.*

Proof of Theorem 5. For any fixed z we can find target allocation $\underline{\hat{x}}^z$. By very similar arguments to the proof of Theorem 4, if z is achievable with a binary priority list, then by Theorem 4 it is achieved by a binary priority list with priority set equal to one of the r prefixes of the priority ordering that satisfies $\frac{\hat{x}^z_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}^z_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}^z_{(r)}}{\lambda_{(r)}}$. Taking each of these prefixes as R_1, we can solve Time-LP r times with $\underline{x}^z$ in the left-hand sides of the constraints. Time-LP is feasible if and only if z can be achieved using a binary priority list with priority set R_1. If Time-LP is feasible for $R_1 = \{(1), \ldots, (k)\}$ with optimal solution γ^*, set $t_B = -\ln(1 - \gamma^*)/\lambda_{R_1}$, and $t_i = 0$ for $i \in R_1$ and $t_i = t_B$ for $i \in R \setminus R_1$. If Time-LP is infeasible for all $k \in [r]$, z is not achievable with a binary priority list. In other words, checking if z is achievable requires solving Time-LP r times.

We perform a binary search over z, using the above procedure as feasibility check. Start with initial bounds $z_- = \min_{i \in R} v_i(\underline{y}_i, 0)$ and $z_+ = \min_{i \in R} v_i(\underline{y}_i, 1)$, and candidate $z = (z_- + z_+)/2$. Whenever the feasibility check returns True (z is achievable with a binary priority list), the lower bound is updated $z_- \leftarrow z$ and the new candidate z is computed. When the feasibility check returns False, the upper bound is updated $z_+ \leftarrow z$ and a new candidate z is computed. In each iteration, it is known that z_- is achievable and z_+ is an upper bound. The process terminates when $z_+ - z_- \leq \epsilon$ and z_- is returned as the near-optimal objective function value. The binary search starts with $z_+ - z_- \leq C$ and this gap halves in each iteration. After $\lceil \log(C/\epsilon) \rceil$ iterations, $z_+ - z_- \leq \epsilon$. Taking C to be a constant, it follows that an ϵ-optimal solution is found in $O(\log(1/\epsilon))$ binary search iterations, where each iteration takes $O(r \cdot \mathcal{LP})$, giving an overall runtime in $O(r \cdot \mathcal{LP} \cdot \log(1/\epsilon))$. $\square$

EC.3. Proofs: Perishable Donations, Binary Priority Lists

Lemma 1. *Consider a stage of length $t < T$ with recipients in $S \subset R$ competing to claim a donation. Identify any pair of subsets of eligible recipients S_1 and S_2 such that $S_1 \subset S \subset S_2 \subseteq R$. One can then decompose this single stage into two new stages, the first with recipients S_1 and length $t_1 < t$, and the second with recipients S_2 and length $t_2 = t - t_1$, so that*

1. *the probability that the donation is claimed by time t remains unchanged; and*
2. *each recipient $i \in S_1$ is at least as likely to claim the donation by time t in the new two-stage plan as in the original one-stage plan.*

Proof of Lemma 1. Let $t_1 = \beta t$ for $0 < \beta \leq 1$ and $t_2 = t - t_1 = (1-\beta)t$. The total response rate is λ_S in the original plan, and λ_{S_1} and λ_{S_2} in the two stages of the modified plan, respectively. We set β such that the probability of the donation being claimed by time t – equivalently, the probability of waste by time t – remains constant:

$$\exp(-\lambda_S t) = \exp(-\lambda_{S_1} t_1) \cdot \exp(-\lambda_{S_2} t_2) = \exp(-(\lambda_{S_1}\beta + \lambda_{S_2}(1-\beta))t) \tag{EC.1}$$

Solving for β, we get $\beta = (\lambda_{S_2} - \lambda_S)/(\lambda_{S_2} - \lambda_{S_1})$, which yields t_1 and t_2.

Let $\underline{x}$ and $\underline{x}'$ be the expected allocations by time t under the original and modified plans, respectively. It remains to establish that recipients in S_1 are at least as well off in the modified plan as before, i.e., $x_i \geq x_i'$ for all $i \in S_1$. For arbitrary $i \in S_1$, since $i \in S_1 \cap S_2$, $x_i = \frac{\lambda_i}{\lambda_S} \cdot (1 - \exp(-\lambda_S t))$ under the original plan. In the modified plan this becomes

$$\begin{aligned}
x_i' &= \frac{\lambda_i}{\lambda_{S_1}}(1-\exp(-\lambda_{S_1}\beta t)) + \frac{\lambda_i}{\lambda_{S_2}}\exp(-\lambda_{S_1}\beta t)(1-\exp(-\lambda_{S_2}(1-\beta)t)) \\
&= \frac{\lambda_i}{\lambda_{S_1}}(1-\exp(-\lambda_{S_1}\beta t)) + \frac{\lambda_i}{\lambda_{S_2}}[\exp(-\lambda_{S_1}\beta t)) - \exp(-\lambda_S t)] && \text{(by (EC.1))} \\
&= \left(\frac{\lambda_i}{\lambda_{S_1}} - \frac{\lambda_i}{\lambda_{S_2}}\right)(1-\exp(-\lambda_{S_1}\beta t)) + \frac{\lambda_i}{\lambda_{S_2}}(1-\exp(-\lambda_S t)) \\
&= \frac{\lambda_i(1-\exp(-\lambda_S t))}{\lambda_S} \cdot \left[\lambda_S\left(\frac{\lambda_{S_2}-\lambda_{S_1}}{\lambda_{S_1}\lambda_{S_2}}\right)\frac{1-\exp(-\lambda_{S_1}\beta t)}{1-\exp(-\lambda_S t)} + \frac{\lambda_S}{\lambda_{S_2}}\right] \\
&= x_i\left[\lambda_S\left(\frac{\lambda_{S_2}-\lambda_{S_1}}{\lambda_{S_1}\lambda_{S_2}}\right)\frac{1-\exp(-\lambda_{S_1}\beta t)}{1-\exp(-\lambda_S t)} + \frac{\lambda_S}{\lambda_{S_2}}\right].
\end{aligned}$$

We want to show that the term in brackets is greater than equal to 1, which can be simplified as:

$$\frac{1-\exp(-\lambda_{S_1}\beta t)}{\lambda_{S_1}} \geq \beta \cdot \frac{1-\exp(-\lambda_S t)}{\lambda_S}.$$

Write $\lambda_{S_1} = \gamma\lambda_S$ and note $0 \leq \gamma < 1$ and $0 < \beta \leq 1$, so $\phi = \beta\gamma < 1$. Let $f(s) = 1 - \exp(-s)$, which is concave since $\exp(-s)$ is convex. The concavity of $f(s)$ implies

$$f(\phi s) = f(\phi s + (1-\phi)0) \geq \phi f(s) + (1-\phi)f(0) = \phi f(s).$$

Setting $s = \lambda_S t$ and plugging in $\phi = \beta\gamma$, we obtain

$$f(\beta\gamma \cdot \lambda_S t) \geq \beta\gamma \cdot f(\lambda_S t) = \beta \cdot \frac{\lambda_{S_1}}{\lambda_S} \cdot f(\lambda_S t) \;\Leftrightarrow\; \frac{1-\exp(-\beta\lambda_{S_1} t)}{\lambda_{S_1}} \geq \beta \cdot \frac{1-\exp(-\lambda_S t)}{\lambda_S}$$

which is what was required to show. It follows that $x_i \geq x_i'$ for all recipients $i \in S_1$. $\square$

THEOREM 6. *Any feasible target allocation $\underline{\hat{x}} \in \mathcal{F}_B(T)$ of a perishable donation can be achieved by using a binary priority list with a priority set $R_1 = \{(1), \ldots, (k)\}$ for some $k \in [r]$ and priority ordering $\pi = ((1), (2), \ldots, (r))$ that satisfies the condition $\frac{\hat{x}_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$.*

Proof of Theorem 6. For ease of exposition, relabel recipients so that $\frac{\hat{x}_1}{\lambda_1} \geq \frac{\hat{x}_2}{\lambda_2} \geq \cdots \geq \frac{\hat{x}_r}{\lambda_r}$. After relabelling, the theorem statement is that the target allocation $\underline{\hat{x}}$ can be achieved by a binary priority list with priority set $[k]$ for some $k \in [r]$. Let (R_1, t_B) be the binary priority list that achieves the target allocation $\underline{\hat{x}}$ using the priority set R_1, the non-priority set $R_2 = R \setminus R_1$, and the notification time t_B for the recipients in R_2. Without loss of generality, we may assume that $\emptyset \subset R_1 \subset R$ and $t_B > 0$. Suppose that there exists a pair of recipients ℓ and k that violate the condition in Theorem 6, i.e., $\ell \in R_1$, $k \in R_2$, and $\ell > k$. We show that this constitutes a contradiction.

Let $R_1' = R_1 \setminus \{\ell\}$ and $R_2' = R_2 \cup \{\ell\}$. By Lemma 1 we can create a new plan for the interval $[0, t_B]$ so that the recipients in R_1' compete for claiming the donation during $[0, t_B')$ for some $t_B' < t_B$, and all recipients in R during $[t_B', t_B)$. Note that the resulting plan also constitutes a binary priority list on $[0, T)$: It is the binary priority list (R_1', t_B') with the priority set R_1', the non-priority set R_2', and the notification time t_B' for the recipients in R_2'.

Let $x_i(s)$ and $x_i'(s)$ be the probability of recipient i claiming the donation by time $s \in [0, T]$, and $p(s) \triangleq \sum_{i \in R} x_i(s)$ and $p'(s) \triangleq \sum_{i \in R} x_i'(s)$ be the total probability that the donation is claimed by time $s \in [0, T]$ in the original and modified binary priority lists, respectively. By part (1) of Lemma 1, $p(t_B) = p'(t_B)$. Due to the memoryless property of exponential distribution and the fact that the same set of recipients (R) are competing for the donation during $[t_B, T)$ under both binary priority lists, the waste probability remains the same: $(1 - p(t_B))\exp(-\lambda_R(T - t_B)) = (1 - p'(t_B))\exp(-\lambda_R(T - t_B))$. So, both binary priority lists satisfy the waste constraint. By part (2) of Lemma 1, $x_i'(t_B) \geq x_i(t_B)$ for $i \in R_1' = R_1 \setminus \{\ell\}$. Again, since the plans are identical during $[t_B, T)$ and $p(t_B) = p'(t_B)$, we conclude that $x_i'(T) \geq x_i(T)$ for $i \in R_1'$. This is equivalent to $x_i' \geq \hat{x}_i$ for $i \in R_1'$, where x_i' is the allocation for recipient i under the modified binary priority list (R_1', t_B'). For $i \in R_2$, the allocation is $\hat{x}_i = \frac{\lambda_i}{\lambda_R}(1 - p(t_B))(1 - \exp(-\lambda_R(T - t_B))$ under the original binary priority list (R_1, t_B). Again for $i \in R_2$, under the modified binary priority list (R_1', t_B'), the allocation is

$$\begin{aligned} x_i' &= \frac{\lambda_i}{\lambda_R} \exp\Big(-\lambda_{R_1'} t_B'\Big)(1 - \exp(-\lambda_R(T - t_B'))) \\ &= \frac{\lambda_i}{\lambda_R} \exp\Big(-\lambda_{R_1'} t_B'\Big)\left[(1 - \exp(-\lambda_R(t_B - t_B'))) + \exp(-\lambda_R(t_B - t_B'))(1 - \exp(-\lambda_R(T - t_B)))\right] \end{aligned}$$

$$
\begin{aligned}
&= \frac{\lambda_i}{\lambda_R} \exp\Big(-\lambda_{R'_1} t'_B\Big)(1-\exp(-\lambda_R(t_B - t'_B))) + \frac{\lambda_i}{\lambda_R}(1-p'(t_B))(1-\exp(-\lambda_R(T-t_B))) \\
&= \frac{\lambda_i}{\lambda_R} \exp\Big(-\lambda_{R'_1} t'_B\Big)(1-\exp(-\lambda_R(t_B - t'_B))) + \frac{\lambda_i}{\lambda_R}(1-p(t_B))(1-\exp(-\lambda_R(T-t_B))) \\
&\geq \frac{\lambda_i}{\lambda_R}(1-p(t_B))(1-\exp(-\lambda_R(T-t_B))) = \hat{x}_i,
\end{aligned}
$$

where the second equality is due to the memorylessness of the exponential distribution and the penultimate transition to the binary priority lists having equal waste probability by time t_B. We conclude that $x'_i \geq \hat{x}_i$ for $i \in R_2$.

Thus, if any recipient is worse off under (R'_1, t'_B), it can only be ℓ. However, $\frac{x'_\ell}{\lambda_\ell} = \frac{x'_k}{\lambda_k}$, because the donation is announced simultaneously to recipients ℓ and k under (R'_1, t'_B), i.e., $\ell, k \in R'_2$. Since $k \in R_2$, it follows from the above that $x'_k \geq \hat{x}_k$. Furthermore, from the premise that $k < \ell$ and the labeling of the recipients upfront, it follows that $\frac{\hat{x}_k}{\lambda_k} \geq \frac{\hat{x}_\ell}{\lambda_\ell}$. Putting it together, $\frac{x'_\ell}{\lambda_\ell} = \frac{x'_k}{\lambda_k} \geq \frac{\hat{x}_k}{\lambda_k} \geq \frac{\hat{x}_\ell}{\lambda_\ell}$, implying $x'_\ell \geq \hat{x}_\ell$. This is a contradiction, as recipient ℓ cannot be better off under the modified binary priority list; recipient ℓ is no longer in the priority set, hence gets notified at a later time ($t'_B > 0$), and competes with all recipients for a longer duration ($T - t'_B > T - t_B$). To show the contradiction formally, we write out $x'_\ell \geq \hat{x}_\ell$ explicitly:

$$
\begin{aligned}
x'_\ell &= \frac{\lambda_\ell}{\lambda_R} \exp\Big(-\lambda_{R'_1} t'_B\Big)(1-\exp(-\lambda_R(t_B - t'_B))) + \frac{\lambda_\ell}{\lambda_R}(1-p'(t_B))(1-\exp(-\lambda_R(T-t_B))) \\
&\geq \frac{\lambda_\ell}{\lambda_{R_1}}(1-\exp(-\lambda_{R_1} t_B)) + \frac{\lambda_\ell}{\lambda_R}(1-p(t_B))(1-\exp(-\lambda_R(T-t_B))) = \hat{x}_\ell
\end{aligned}
$$

where the second terms cancel because $p(t_B) = p'(t_B)$. So, this inequality holds if and only if

$$
\exp\Big(-\lambda_{R'_1} t'_B\Big) \cdot \frac{1-\exp(-\lambda_R(t_B - t'_B))}{\lambda_R} \geq \frac{1-\exp(-\lambda_{R_1} t_B)}{\lambda_{R_1}}
$$

which contradicts $\exp\Big(-\lambda_{R'_1} t'_B\Big) < 1$ and $(1-\exp(-\lambda_R(t_B - t'_B)))/\lambda_R < (1-\exp(-\lambda_{R_1} t_B))/\lambda_{R_1}$. Hence, by contradiction, there must not exist any such pair of recipients ℓ and k in the binary priority list (R_1, t_B) that violate the condition in Theorem 6. □

Theorem 7. *Consider a perishable donation with deadline $T < \infty$. Fix a constant $\epsilon > 0$, which can be arbitrarily close to zero. An ϵ-optimal binary priority list $(R_1^\circ, R_2^\circ, t_B^\circ)$ resulting in an objective function value z_B° that satisfies $|z_B^\circ - z_B^*| \leq \epsilon$ can be computed by executing $O(\ln(1/\epsilon))$ binary search iterations. Each iteration involves computing the bounds t_0, t_{LB}, and t_{UB}, and verifying if the set of feasible notification times $\{t:\ t \leq t_0,\ t_{LB} \leq t \leq t_{UB}\}$ is empty or not at most r times.*

Proof of Theorem 7. For any fixed z we can find target allocation $\underline{\hat{x}}^z$. If z is achievable with a binary priority list, then by the same arguments as in Theorem 6 it is achieved by a binary priority list with priority set equal to one of the r prefixes of the priority ordering that satisfies

$\frac{\hat{x}^z_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}^z_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}^z_{(r)}}{\lambda_{(r)}}$. Taking each of these prefixes as R_1, we can compute t_0, t_{LB} and t_{UB} and check if the set $H(R_1) = \{t \in [0,T] : t \leq t_0, t_{LB} \leq t \leq t_{UB}\}$ is nonempty. If $H(R_1)$ is non-empty, then using R_1 along with a notification time $t_B \in H(R_1)$ gives a binary priority list achieving z. If $H(R_1)$ is empty, no binary priority list with priority set R_1 achieves z. It follows that checking whether a given z is achievable with a binary priority list requires computing t_0, t_{LB} and t_{UB} at most r times.

We perform a binary search over z, using the above procedure as feasibility check. Start with initial bounds $z_- = \min_{i\in R} v_i(\underline{y}_i, 0)$ and $z_+ = \min_{i\in R} v_i(\underline{y}_i, 1)$, and candidate $z = (z_- + z_+)/2$. Whenever the feasibility check returns True (z is achievable with a binary priority list), the lower bound is updated $z_- \leftarrow z$ and the new candidate z is computed. When the feasibility check returns False, the upper bound is updated $z_+ \leftarrow z$ and a new candidate z is computed. In each iteration, it is known that z_- is achievable and z_+ is an upper bound. The process terminates when $z_+ - z_- \leq \epsilon$ and z_- is returned as the near-optimal objective function value. The binary search starts with $z_+ - z_- \leq C$ and this gap halves in each iteration. After $\lceil \log(C/\epsilon) \rceil$ iterations, $z_+ - z_- \leq \epsilon$. Taking C to be a constant, it follows that an ϵ-optimal solution is found in $O(\log(1/\epsilon))$ binary search iterations. $\square$

EC.4. Proofs: Perishable Donations, n-stage Priority Lists

Lemma 2. *Consider a stage with high response rate λ_h, length s_h, and claim probability p_h, followed by a stage with low response rate λ_ℓ ($\lambda_\ell < \lambda_h$), length s_ℓ and claim probability p_ℓ. The two stages can be reordered and resized so that the low-response-rate stage occurs first, the new claim probabilities p'_ℓ and p'_h equal the old ones (i.e., $p'_\ell = p_\ell$, $p'_h = p_h$), and the new lengths s'_ℓ and s'_h are such that the total time required does not increase (i.e., $s'_\ell + s'_h \leq s_\ell + s_h$).*

Proof of Lemma 2. Consider any two adjacent stages as described in the lemma. We refer to the first stage as stage h and the second as stage ℓ. Without loss of generality, we may assume they are the first and second stages of an n-stage priority list. Define $\rho < 1$ and σ as constants that satisfy $\lambda_\ell = \rho\lambda_h$ and $s_\ell = \sigma s_h$. The claim probabilities are: $p_h = 1 - \exp(-\lambda_h s_h)$ and $p_\ell = \exp(-\lambda_h s_h)(1 - \exp(-\rho\lambda_h \cdot \sigma s_h))$. When placing stage ℓ first and resizing it to have length s'_ℓ, its claim probability changes to $p'_\ell = 1 - \exp(-\rho\lambda_h \cdot s'_\ell)$. Setting $p_\ell = p'_\ell$ and solving for s'_ℓ yields

$$s'_\ell = \frac{-\ln(1 - p_\ell)}{\rho\lambda_h} = \frac{-\ln(1 - \exp(-\lambda_h s_h) + \exp(-\lambda_h s_h(1 + \rho\sigma)))}{\rho\lambda_h}.$$

We want to similarly choose a new length s'_h for stage h, maintaining $p_h = p'_h$ subject to the constraint that $s'_\ell + s'_h \leq s_h + s_\ell = s_h + \sigma s_h$. To that end, we show that using $s'_h = s_h(1 + \sigma) - s'_\ell$, the longest feasible length for the new stage h, yields $p'_h \geq p_h$. Setting $s'_h = s_h(1 + \sigma) - s'_\ell$ and plugging in s'_ℓ, the claim probability in the new stage h becomes

$$p'_h = \exp(-\rho\lambda_h s'_\ell)(1 - \exp(-\lambda_h(s_h(1 + \sigma) - s'_\ell)))$$

$$= [1 - \exp(-\lambda_h s_h) + \exp(-\lambda_h s_h(1+\sigma\rho))] \left(1 - \frac{\exp(-\lambda_h s_h(1+\sigma))}{[1 - \exp(-\lambda_h s_h) + \exp(-\lambda_h s_h(1+\sigma\rho))]^{1/\rho}}\right)$$

$$= [1 - \exp(-\lambda_h s_h) + \exp(-\lambda_h s_h(1+\sigma\rho))] - \frac{\exp(-\lambda_h s_h(1+\sigma))}{[1 - \exp(-\lambda_h s_h) + \exp(-\lambda_h s_h(1+\sigma\rho))]^{1/\rho - 1}}$$

Suppose for contradiction $p'_h < p_h = 1 - \exp(-\lambda_h s_h)$. This inequality holds if and only if

$$(1 - \exp(-\lambda_h s_h)) \left[\exp\left(-\frac{\lambda_h s_h(1+\sigma\rho)\rho}{1-\rho}\right) - \exp\left(-\frac{\lambda_h s_h(1+\sigma)\rho}{1-\rho}\right)\right] < 0$$

which is a contradiction since both terms are nonnegative. We conclude that $p'_h \geq p_h$ even if the maximum allowable length for stage h was used. As a result, we can find a feasible s'_h such that $p'_h = p_h$ without exceeding the time constraint. This shows that it is possible to sort adjacent stages in order of increasing total response rates while maintaining the same claim probabilities and adjusting the length of stages without increasing the total length. □

Theorem 8. *Any feasible target allocation $\underline{\hat{x}} \in \mathcal{F}(T)$ of a perishable donation can be achieved by using an n-stage priority list with priority ordering $\pi = ((1),(2),\ldots,(r))$ that satisfies the condition $\frac{\hat{x}_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$.*

Proof of Theorem 8. For ease of exposition, relabel recipients so that $\frac{\hat{x}_1}{\lambda_1} \geq \frac{\hat{x}_2}{\lambda_2} \geq \cdots \geq \frac{\hat{x}_r}{\lambda_r}$. After relabelling, the theorem statement is that the target allocation $\underline{\hat{x}}$ can be achieved by an n-stage priority list with priority ordering $\pi = (1,2,\ldots,r)$. Let $\underline{\hat{t}}$ be the n-stage priority list that achieves the target allocation $\underline{\hat{x}}$ using the priority ordering $\hat{\pi}$. Suppose that $\hat{\pi}$ is not π. We show that this constitutes a contradiction.

There must exist a stage in $\underline{\hat{t}}$ with a set of recipients in $S \subset R$ competing to claim the donation within a certain time interval such that S does not comprise a prefix of π, i.e., $\nexists\, j: \; S = [j]$. Let S_1 be the largest prefix of π (which may be the empty set) included in S, i.e., $S_1 = [k]$ if $k = \max\{j \in [r] : [j] \subset S\}$ exists, or $S_1 = \emptyset$ and $k = 0$ otherwise. Set $S_2 = S \cup \{k+1\}$. Applying Lemma 1 to this stage, we can decompose it into two stages (with recipients S_1 and S_2) taking the same total time while maintaining the same claim probability within the associated time interval and making all recipients in S_1 better off. Note that after this decomposition, and performing other decomposition steps if necessary until all stages become prefixes of π, by Lemma 2 we can resort all the decomposed stages so that the resulting sequence of stages form a proper n-stage priority list without changing the claim probabilities or lengthening the stages. Thus, we can legitimately compare the new n-stage priority list with the original one, and the two lemmas directly imply $x'_i \geq \hat{x}_i$ for $i \in S_1$, where $\underline{x}'$ is the new allocation vector. Recipient $k+1$ is also better off, $x'_{k+1} \geq \hat{x}_{k+1}$, as she is now notified of the donation earlier than before – during the stage in question ($k+1 \in S_2$) rather than the next one ($k+1 \notin S$). However, any recipient $\ell \in S \setminus S_1$ (there must exist at least one) suffer from the decomposition, as they

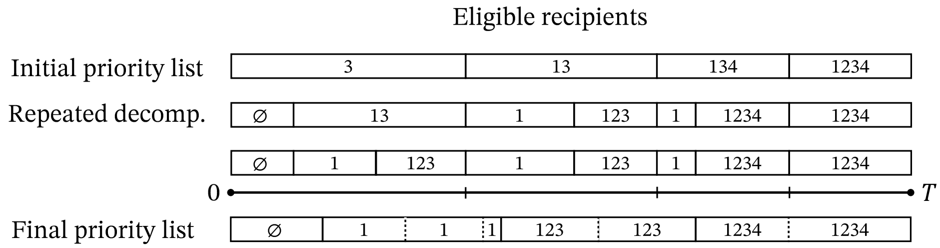

Figure EC.1 (Repeat of Figure 3) Example showing how an initial priority order 3142 can be converted into an priority list with priority order 1234 while maintaining the objective value and service level constraints.

are notified of the donation later than before, during the second of the decomposed stages, and they compete with one extra recipient $(k+1)$. We show that this observation stands in contradiction with a logical consequence of our premise. Since ℓ competes for the donation during (at least) every stage $k+1$ competes for it, the contribution of this stage to their expected allocations are proportional to their response rates, i.e., $\ell, k+1 \in S_2$ implies $x'_\ell/\lambda_\ell \geq x'_{k+1}/\lambda_{k+1}$. We already know $x'_{k+1} \geq \hat{x}_{k+1}$. We also know $\hat{x}_{k+1}/\lambda_{k+1} \geq \hat{x}_\ell/\lambda_\ell$, since $k+1$ comes before ℓ in the priority ordering. Putting it all together, $\frac{x'_\ell}{\lambda_\ell} \geq \frac{x'_{k+1}}{\lambda_{k+1}} \geq \frac{\hat{x}_{k+1}}{\lambda_{k+1}} \geq \frac{\hat{x}_\ell}{\lambda_\ell}$, from which we conclude that $x'_\ell \geq \hat{x}_\ell$ for $\ell \in S \setminus S_1$, which is a contradiction. Therefore, our premise that $\hat{\pi}$ is not π must be wrong. □

THEOREM 9. *Consider a perishable donation with deadline $T < \infty$. Fix a constant $\epsilon > 0$, which can be arbitrarily close to zero. An ϵ-optimal n-stage priority list resulting in an objective function value z° that satisfies $|z^\circ - z^*| \leq \epsilon$ can be computed by executing $O(\ln(1/\epsilon))$ binary search iterations. Each iteration involves a feasibility check using Algorithm 2.*

We start by showing the correctness of Algorithm 2.

LEMMA EC.2. *Given $\underline{y}, z, R$, suppose Algorithm 2 returns $\underline{t}$. Then the priority list with annoucement times $\underline{t}$ satisfies the waste constraint and achieves objective function value at least z.*

Proof. Suppose that Algorithm 2 returns $\underline{t}^{\text{ALG}}$. We show that $x(\underline{t}^{\text{ALG}}) \geq \hat{x}^z$ and $x_0(\underline{t}^{\text{ALG}}) \leq \alpha$.

First, notice that once some t_k is set based on $s_{k-1,\alpha}$ rather than $s_{k-1,k}$, all subsequent t_ℓ, $\ell > k$ will also be set based on $s_{\ell-1,\alpha}$. In fact, it will hold that $t_k = t_\ell$ for all $k < \ell \leq r$. It follows that there is some k^* so that $t_1, \ldots, t_{k^*}$ is set based on $s_{i-1,i}$ and $t_{k^*} + s_{k^*,\alpha} = t_{k^*+1} = \ldots = t_r$. Let $t^{[k]} = (t_1, \ldots, t_{k-1}, t_k, t_k, \ldots, t_k)$ be the vector of announcement times which results from executing Algorithm 2 for the first k iterations to determine $t_1, \ldots, t_k$, then announces the donation to all remaining recipients at the same time at time t_k. In this notation, $\underline{t}^{\text{ALG}} = t^{[k^*]}$.

By definition, for any $k = 1, \ldots, r-1$, $s_{k-1,\alpha}$ is the longest interval between announcing to $k-1$ and k which still allows the priority list to satisfy the waste constraint if the donation is immediately announced to all remaining recipients $k, \ldots, r$ at time $t_{k-1} + s_{k-1,\alpha}$. By construction, $t^{[k]}$ satisfies $x_0(t^{[k]}) \leq \alpha$ for any $k = 1, \ldots, k$ and, in particular, $x_0(t^{[k^*]}) \leq \alpha$.

Suppose for contradiction that $\underline{t}^{\mathrm{ALG}}$ does not achieve objective function value z. In each iteration of Algorithm 2, $\underline{p}$ is updated so that, at conclusion, $x(\underline{t}) = \underline{x}$. If there is a recipient i for which $v_i(\underline{y}_i, x_i) < z$ the if-statement in line 12 returns infeasible. It follows that if Algorithm 2 returns $\underline{t}^{\mathrm{ALG}}$, $v_i(\underline{y}_i, x_i(\underline{t}^{\mathrm{ALG}})) \geq z$ for all $i \in [r]$. We conclude that when Algorithm 2 returns $\underline{t}^{\mathrm{ALG}}$, the corresponding priority list satisfies the waste constraint and has objective function value at least z. □

Lemma EC.3. *Given $\underline{y}, z, R$, suppose Algorithm 2 returns Infeasible. Then no priority list satisfies the waste constraint and achieves objective function value at least z.*

Proof. We show the contrapositive, that if a priority list satisfies the waste constraint and achieves z then Algorithm 2 does not return infeasible.

Suppose some priority list satisfies the waste constraint and achieves z by T. By Theorem 8 a priority list satisfying $\frac{\hat{x}_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}_{(2)}}{\lambda_{(2)}} \geq \cdots \geq \frac{\hat{x}_{(r)}}{\lambda_{(r)}}$ is waste-feasible and achieves z. Relabel the recipients so that $\frac{\hat{x}_1}{\lambda_1} \geq \cdots \geq \frac{\hat{x}_r}{\lambda_r}$. There is at least one feasible announcement time vector $\underline{t}$ with $t_1 \leq \ldots \leq t_n$.

Suppose for contradiction that Algorithm 2 returns infeasible, which can happen as a result of either line 2 or line 12. Since $\underline{t}^*$ is feasible and achieves z, the if-statement on line 2 can not return infeasible. This means that the algorithm executed the for-loop and computed the corresponding $\underline{t}^{\mathrm{ALG}}$ and $\underline{p}$ and that there was some i for which $v_i(\underline{y}_i, x_i) = v_i(\underline{y}_i, x_i(\underline{t}^{\mathrm{ALG}})) < z$.

Among feasible announcement time vectors, let $\underline{t}^*$ be one with largest $j = \min\{j \in [r] : t_j^* \neq t_j^{\mathrm{ALG}}\}$ and, among those, the one with $|t_j^* - t_j^{\mathrm{ALG}}|$ as small as possible. From line 10, $t_j^{\mathrm{ALG}} = t_{j-1}^{\mathrm{ALG}} + s_{j-1,j}$ or $t_j^{\mathrm{ALG}} = t_{j-1}^{\mathrm{ALG}} + s_{j,\alpha}$; we analyze the cases separately.

Suppose $t_j^{\mathrm{ALG}} = t_{j-1}^{\mathrm{ALG}} + s_{j-1,j}$.

Case 1.1 $t_j^{\mathrm{ALG}} < t_j^*$: This implies $t_j^* \leq t_{j-1}^{\mathrm{ALG}} + s_{j,\alpha}$. Consider the schedule with the same recipients eligible to claim the item as under t^*, except that all of R can claim during the period $[t_j^{\mathrm{ALG}}, t_j^*]$. Apply Lemma 2 to rearrange the periods in order of increasing response rate while keeping the per-period allocation probabilities unchanged. Call the announcement times of the resulting schedule $\underline{t}'$. Notice that $t'_\ell = t^*_\ell$ for $\ell = 1, \ldots, j$. We need to ensure the $\underline{t}'$ priority list is α-feasible and achieves z. First, increasing the number of recipients eligible to claim the donation in $[t_j^{\mathrm{ALG}}, t_j^*]$ increases the probability it is claimed in that period, and all other allocation probabilities are unaffected by the application of Lemma 2, so $x_0(\underline{t}') \leq x_0(\underline{t}^*) \leq \alpha$. Second, compared to $\underline{t}^*$, recipients $1, \ldots, j-1$ suffer from this change while $j, \ldots, r$ benefit. Since $t'_1, \ldots, t'_{j-1}$ equal $t_1^{\mathrm{ALG}}, \ldots, t_{j-1}^{\mathrm{ALG}}$, recipients $1, \ldots, j-1$ reach objective function value z at the same point in time. Moreover, after applying Lemma 2, $t'_j = t_j^{\mathrm{ALG}} = t_{j-1}^{\mathrm{ALG}} + s_{j-1,j}$ so recipient j reaches objective function value z at the same instant as $1, \ldots, j-1$. Recipient j has $x_j(\underline{t}') \geq x_j(\underline{t}^*) \geq \hat{x}_j^z$ since $\underline{t}^*$ achieved z, implying that $x_\ell(\underline{t}') \geq \hat{x}_\ell^z$ for all $\ell = 1, \ldots, j$. It follows that $\min_{i \in [r]} v_i(\underline{y}_i, x_i(\underline{t})) \geq z$. But this contradicts the choice of $\underline{t}^*$, since the the first disagreement between $\underline{t}'$ and $\underline{t}^{\mathrm{ALG}}$ must occur at some index $j' > j$.

Case 1.2 $t_j^{\mathrm{ALG}} > t_j^*$: We split into cases based on t_{j+1}^*.

a) $t_j^* < t_{j+1}^*$: Take the period $[t_j^*, \min\{t_{j+1}^*, t_j^{\mathrm{ALG}}\}]$ and apply Lemma 1 with $S_1 = [k-1]$ and $S_2 = [r]$ to decompose it into two stages, the first $[t_j^*, t'']$ during which $[k-1]$ can claim the item, the second $[t'', \min\{t_{j+1}^*, t_j^{\mathrm{ALG}}\}]$ during which $[r]$ can claim the item. Then apply Lemma 2 to sort the periods by increasing response rate while keeping the allocation probabilities unchanged; call the announcement times of the resulting schedule $\underline{t}'$. Notice $t_j^* < t_j' = t'' \le t_j^{\mathrm{ALG}}$. Lemma 1 ensures that the probability the item remains claimed during $[t_j^*, \min\{t_{j+1}^*, t_j^{\mathrm{ALG}}\}]$ remains unchanged during the initial decomposition, and by Lemma 2 sorting does not affect the per period allocation probabilities. It follows that, since $\underline{t}^*$ is waste-feasible, $\underline{t}'$ is also. Recipient j is worse-off in $\underline{t}'$ than in $\underline{t}^*$. However, $t_j' \le t_j^{\mathrm{ALG}}$, so by the time recipients $1, \ldots, j-1$ (simultaneously) reach objective function value z, which they will since they do under $\underline{t}^*$ and improve in $\underline{t}'$, j will also have $x_j(\underline{t}') \ge \hat{x}_j^z$. It follows that $\min_{i \in [r]} v_i(\underline{y}_i, x_i(\underline{t})) \ge z$. But this contradicts the choice of $\underline{t}^*$, since the the first disagreement between $\underline{t}'$ and $\underline{t}^{\mathrm{ALG}}$ occurs at index j and $|t_j^* - t_j^{\mathrm{ALG}}| > |t_j' - t_j^{\mathrm{ALG}}|$.

b) $t_j^* = t_{j'}^*$ for $j' = j+1, \ldots k < r$: Decompose $[t_j^*, \min\{t_{k+1}^*, t_j^{\mathrm{ALG}}\}]$ using Lemma 1 with $S_1 = [k-1]$ and $S_2 = [r]$ and sort using Lemma 2 to get announcement times $\underline{t}'$. The analysis is identical to the previous case except that now $t'' = t_j' = t_\ell'$ for $\ell = j+1, \ldots, k$ and recipients $j, \ldots, k$ are worse off compared to $\underline{t}^*$, but since $t'' \le t_j^{\mathrm{ALG}}$ we can still argue that $j, \ldots, k$ will reach z no later than $1, \ldots, j-1$ reach z.

c) $t_j^* = t_{j'}^*$ for $j' = j+1, \ldots r$: Set $\epsilon = (t_j^{\mathrm{ALG}} - t_j^*)/2$ and consider announcement times $\underline{t}'$ with $t_i' = t_i^*$ for $i = 1, \ldots, j-1$ and $t_i' = t_j^* + \epsilon$ for $i = j, \ldots, r$. By construction, $t_i' < t_j^{\mathrm{ALG}}$ for $i = j, \ldots, r$ so we can argue as before that $j, \ldots, r$ reach z no later than $1, \ldots, j-1$, all of which improve from $\underline{t}^*$ to $\underline{t}'$. Furthermore, $\underline{t}'$ is waste-feasible since $t_i' = t_i^{\mathrm{ALG}}$ for $i \in [j-1]$ and then the donation is announced to all remaining recipients simultaneously at $t_j^* + \epsilon < t_j^{\mathrm{ALG}} \le t_{j-1} + s_{j,\alpha}$, in other words, before the last possible moment to announce to all remaining recipients to satisfy the waste constraint. Again, this contradicts the choice of $\underline{t}^*$ since the the first disagreement between $\underline{t}'$ and $\underline{t}^{\mathrm{ALG}}$ occurs at index j and $|t_j^* - t_j^{\mathrm{ALG}}| > |t_j' - t_j^{\mathrm{ALG}}|$.

Suppose instead $t_j^{\mathrm{ALG}} = t_{j-1}^{\mathrm{ALG}} + s_{j,\alpha}$. If $t_j^* > t_j^{\mathrm{ALG}} = t_{j-1}^{\mathrm{ALG}} + s_{j,\alpha}$ then $\underline{t}^*$ can not be waste-feasible, since (given $t_1, \ldots, t_{j-1}$) $t_{j-1}^{\mathrm{ALG}} + s_{j,\alpha}$ is the last possible moment that the donation can be announced to j while satisfying the waste constraint. It follows that this can not occur. If, instead, $t_j^* < t_j^{\mathrm{ALG}}$, the analysis proceeds as in case 1.2 above and contradicts the choice of $\underline{t}^*$.

We conclude that if any waste-feasible priority list achieves z by T, then Algorithm 2 will not return infeasible. □

Using these lemmas, we prove Theorem 9.

Proof of Theorem 9. For any fixed z we can find target allocation $\hat{x}^z$. If z is achievable with a binary priority list, then by Theorem 8 it is sufficient to focus on the priority ordering satisfying $\frac{\hat{x}^z_{(1)}}{\lambda_{(1)}} \geq \frac{\hat{x}^z_{(2)}}{\lambda_{(2)}} \geq \cdots$. By Lemmas EC.2 and EC.3, Algorithm 2 can be used as feasibility check for z as part of a binary search procedure. The limit on the number of iterations follows as in Theorem 7. $\square$

EC.5. Additional computational results

Complementary figures to those reported in Section 4 for the other valuation function.

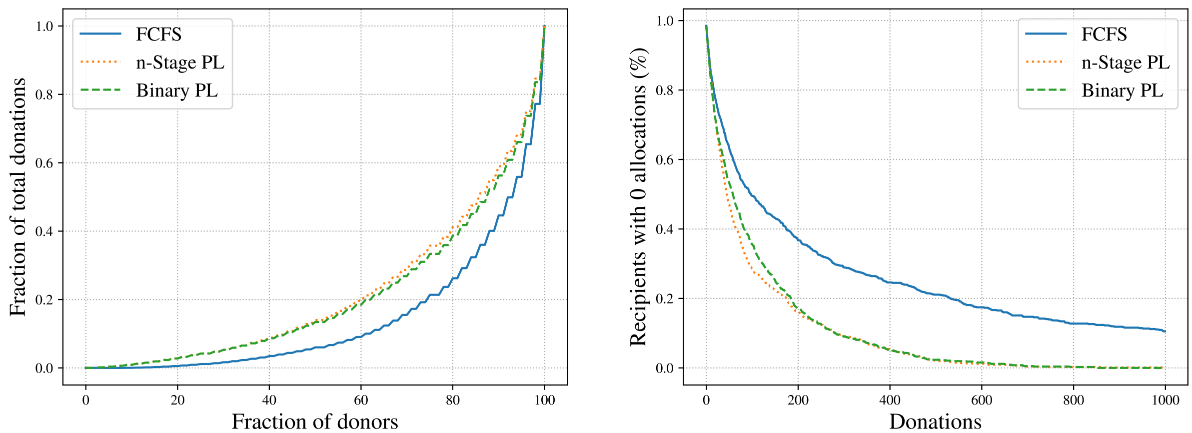

Figure EC.2 Results for v^W. Left: the Lorentz curve where (x, y) represents an x fraction of recipients receiving a y fraction of donations. Right: the fraction of recipients who have not received any allocations over time.

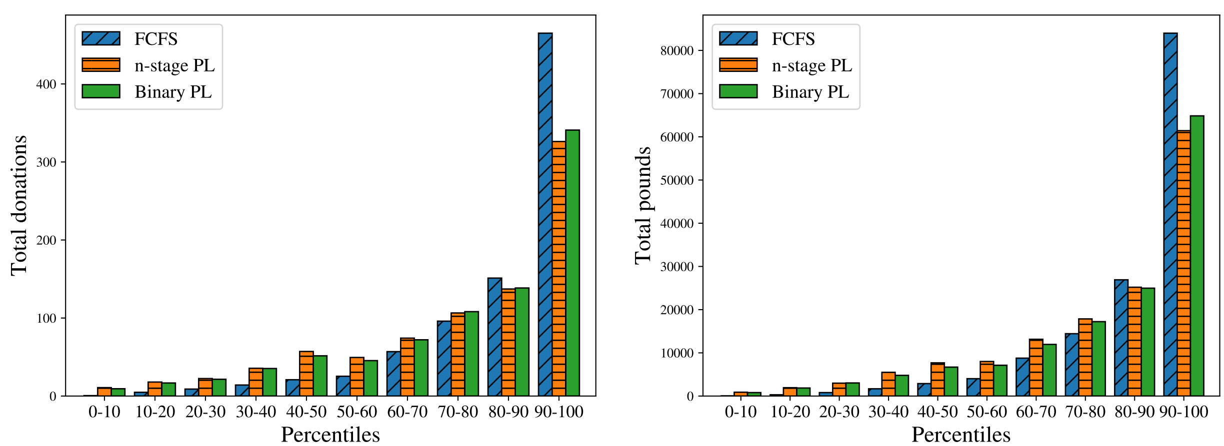

Figure EC.3 Distribution of donations per decile for v^N (number of donations; left) and v^W (pounds of donations received; right). Both binary and n-stage priority lists allocate more than FCFS to the bottom eight deciles and less to the top two.

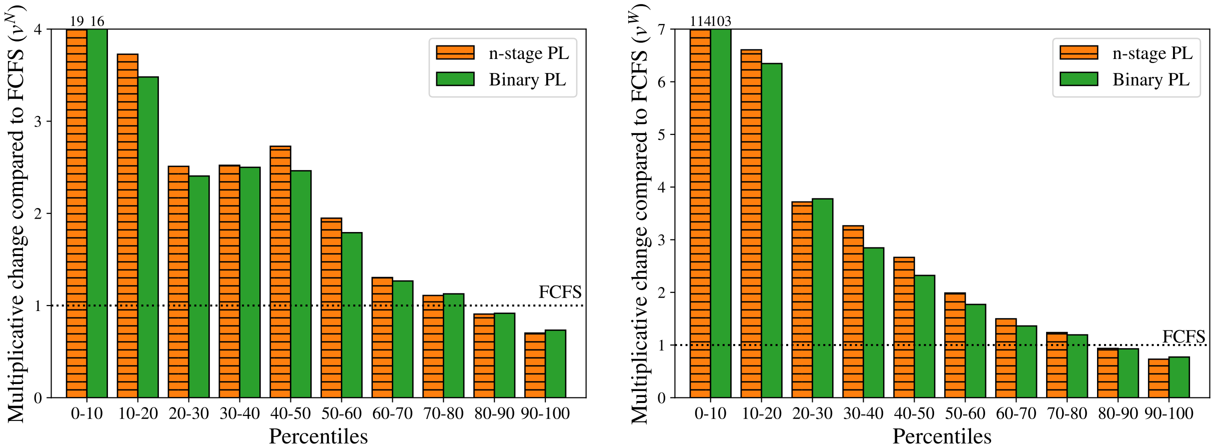

Figure EC.4 Multiplicative change in distribution of donations per decile over the FCFS allocation. Results are visualized for v^N (number of donations; left) and v^W (pounds of donations received; right). The worst-off recipients benefit more from priority lists.

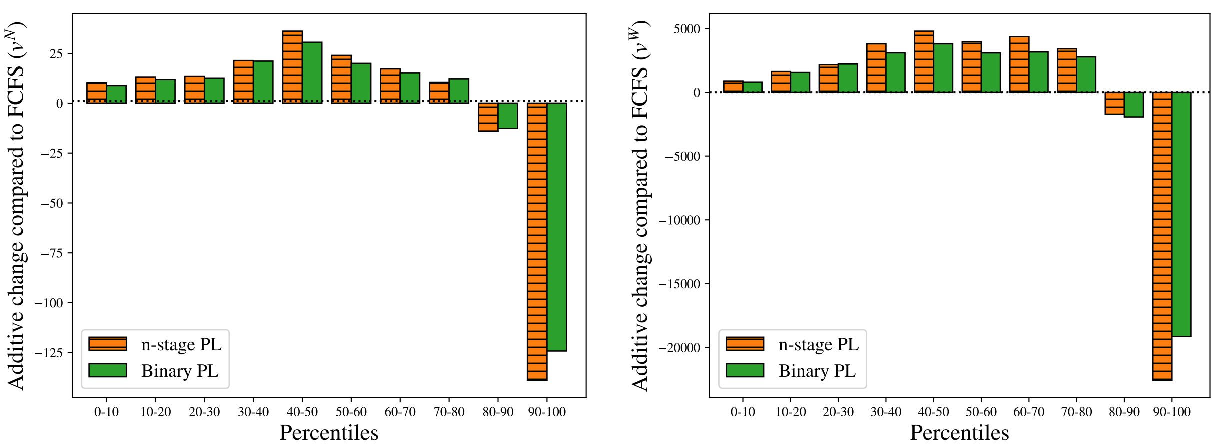

Figure EC.5 Additive change in distribution of donations per decile compared to the FCFS allocation. Results are visualized for v^N (number of donations; left) and v^W (pounds of donations received; right).