Achieving Rawlsian Justice in Food Rescue

GERDUS BENADÈ, Boston University, USA
AYDIN ALPTEKINOGLU, Pennsylvania State University, USA

We study a problem faced by a national food rescue platform which matches donations 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 these potentially inequitable outcomes. We show that an n-stage priority list, which controls the announcement time for every agent individually, can achieve any desired fractional allocation, hence any fairness target. Two-stage priority lists, which announce a donation in only two waves, are simpler to implement and administer but offer less fine-grained control over fairness outcomes. We give polynomial-time algorithms to find the 2-stage and n-stage priority lists that maximize a class of Rawlsian objective functions. Computational experiments confirm that even simple, 2-stage priority lists lead to significantly more balanced allocations than the existing first-come first-served allocation rule.

ACM Reference Format:
Gerdus Benadè and Aydin Alptekinoglu. 2023. Achieving Rawlsian Justice in Food Rescue. 1, 1 (September 2023), 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

We partner with a food rescue platform (FRP) which matches donations to recipients in 50 states. The organization maintains a database of recipients in each area (typically food pantries and emergency relief organizations) who sign up to receive notifications when donations are available. When the FRP is notified of a donation, which can range from left-over already-prepared meals at a catered event to 20,000 gallons of dairy, they announce the donation, including a description and pickup location, to those recipients who are able collect the donation from the donor location (for smaller donations this is usually recipients within a 15-30 mile radius, for larger donations this may include recipients in other states). The first recipient to respond that they want the donation 'claims' it and is placed into contact with the donor to arrange pickup.

By acting as matchmaker in this way,[1] the FRP enables donations to reach those who have active and current need for it with minimal intervention and without the informational and logistical overhead of eliciting each recipient's value for an item, which may be necessary for any centralized allocation algorithm. Of course, the first-come first-serve (FCFS) nature of the matching system also has a very real downside. Due to differences in size and staffing levels, some organizations are able to receive and respond to notifications much more quickly than others. As a result, in a three year sample of data from several counties in Florida, 5% of recipients recieved 50% of donations (and 82% of the donated pounds of food). This concentration of donations is common across the industry even at organizations who do not make

[1] The FRP has recently experimented with providing delivery services themselves and offering the option of splitting a large donation into multiple smaller ones. However, in the majority of states they still function only as a matching platform — we study this aspect and leave the interesting logistical questions around volunteer delivery services and when to break bulk for future work.

Authors' addresses: Gerdus Benadè, Boston University, USA, benade@bu.edu; Aydin Alptekinoglu, Pennsylvania State University, USA, aydin@psu.edu.

allocations on an FCFS basis; for example, Lee et al. [7] report that (prior to intervention) 412 Food Rescue[2] allocated 70% of their donations to 20% of recipients.

Our goal is to develop a framework for the more equitable distribution of donations, given the constraints of the FRP's existing IT infrastructure and without increasing the elicitation or reporting burden on any party.

Intuitively, we may agree that allocating 50% of donations to 5% of recipients is not a desirable outcome, but this does not answer the question of what objective function to optimize. From discussions with the FRP, we learn that they value allocating donations to recipients proportional to the recipient's current demand or, if demand information is not readily available,[3] closer to uniformly. After deliberation, the FRP set a Rawlsian objective of maximizing the *value* for the worst-off recipient. We consider three possible meanings of 'value': the number of donations received, the total weight (in pounds) of the donations received, and the fraction of the recipient's demand fulfilled by the donations.

Next, we narrow down the space of potential allocation mechanisms. The FRP wants to minimize any disruption caused to existing recipients by a new mechanism. This excludes market-based mechanisms (as employed at Feeding America [8]) that require eliciting every recipient's value for every item. A promising class of allocation mechanisms are *priority lists*, which retains all of the existing FCFS infrastructure but gives the FRP control over one additional parameter: exactly when a donation is announced to each recipient. Ideally, by delaying announcing an item to the fastest responders, others will have more opportunity to claim donations, eventually leading to more equitable outcomes. Priority lists offer several other benefits. Compared to the FRP unilaterally deciding where to assign each item, it continues to involve recipients in the allocation process and avoids situations where the FRP makes inappropriate or wasteful allocations due to incomplete information about the current operational constraints (capacity, need etc.) at the recipient. Priority lists are easy to explain to recipients, and require only minor changes to the current IT infrastructure. From the recipient's perspective this intervention is minimally disruptive — the claiming process under a priority list works exactly as before, in fact, it may not even be possible for a recipient to tell when a priority list is used.

1.1 Our contribution

We study whether priority lists lead to more a more equitable distribution of donations and show how to construct priority lists that optimize Rawlsian objectives under exponentially distributed response rates.

In Section 3, we study n-stage priority lists, which take the form of an ordering of the recipients and, for each recipient, the time after which the donation is announced to them (if still unclaimed). We characterize the expected (fractional) allocations that are achievable for each ordering of the recipients. Next, we observe that equal fractional allocations can be achieved by a priority list which orders recipients from slowest to fastest. We show the flexibility of n-stage priority lists and generalize this idea to achieve any desired fractional allocation: order recipients in decreasing order of desired fractional share divided by response rate. This observation, together with a simple water-filling algorithm, is enough to maximize the value of the worst-off agent.

Next, in Section 4, we study a simpler form of priority lists we call *binary* or 2-*stage priority lists*, in which a donation is first announced to some subset of recipients and, after some time, announced to the rest. In other words, a donation is announced to everyone after only two stages, rather than individually in n stages. Binary priority lists are less flexible than n-stage priority lists and can not achieve every fractional allocation. However, we show that there are always optimal binary priority lists where the set of priority recipients have a very particular structure, specifically,

[2]A food rescue service operating out of Pittsburgh, PA.

[3]There are good reasons why recipients may not want to publicize or share how many individuals they serve, for example, if the organization works exclusively with undocumented immigrants.

sort recipients by the ratio of their desired fractional allocation to their response rate and include them in the priority set in this order (until some cut-off). This enables an efficient algorithm for finding the optimal binary priority list.

Finally, we test our ideas on real data in Section 5. We use a sample of data from Florida to calibrate our response rates and item sizes and simulate the existing FCFS mechanism as well as some variants on the optimal binary priority lists. We find that priority lists lead to a significantly more equitable distribution of donations across all metrics. For example, under FCFS roughly 20% of recipients receive no items and 75% of items are concentrated among only 20% of recipients, in contrast, all recipients receive at least one item under the optimal binary priority list and the best-off 20% of recipients receive only 30% of all items. The simulations also reveal a drawback of priority lists, which is that they slow down allocation times, suggesting that they should be deployed carefully when allocating perishable goods. We leave further investigation of this for future study.

1.2 Related work

We briefly remark on some of the most closely related literature. On the theoretical side, Kawase and Sumita[6] 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 drawn from a distribution. In contrast, we ignore agent values and focus on the effect of response rates in an FCFS setting where the allocation mechanism can control the (fractional) allocation only indirectly through the construction of an appropriate priority list.

Several papers study or propose allocation mechanisms in real-world food rescue organizations. Prendergast [8] analyzes a marketplace rolled out at Feeding America which lets each recipient bid some artificial currency on roughly 30 truckloads of 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. [7], in collaboration with 412 Food Rescue, develop an algorithmic framework that lets dispatchers train a model which recommends who should receive a donation and how it should be delivered to them. Shi et al. [9], working with the same organization, propose a machine learning algorithm to recommend which recipients should be contacted first for a given donation. Aleksandrov et al. [1] study simple mechanisms where an agent only indicates whether they like an item or not, and give results on envy-freeness (which requires that every agent prefers their own bundle over every other) and strategyproofness.

Several papers study the fair allocation of indivisible resources arriving online with an eye to minimizing envy [5] and simultaneously maintaining efficiency [4, 10, 11], even in cases where the allocation algorithm has only limited information about recipients' preferences [3]. Closer to our objective, Banerjee et al. [2] aim for proportional allocations instead of envy-freeness and ask if having predictions of recipients' values for goods help. In contrast, our notion of fairness is maximizing the welfare of the worst-off recipient rather than proportional allocation or minimizing envy and we do not evaluate the outcome with regards to underlying values.

2 MODEL

We develop a simplified model of the decision problem faced by the food rescue service. Consider a set $\mathcal{N}$ of n agents, or recipients. Let $[n] = \{1, \ldots, n\}$. For set $S \subset \mathcal{N}$, let $\bar{S} = \mathcal{N} \setminus S$. A sequence of m *items* or *donations* arrive, one after the other, to be allocated immediately and irrevocably. A item typically consists of a quantity of food from one or more categories (for example, produce, dairy or prepared meals). We let s_t denote the size of donation t, typically measured in pounds. Donation t is within the pickup radius of recipients $R_t \subseteq \mathcal{N}$, we take R_t as known since recipients specify a pickup radius when signing up. Each item will be announced to (and competed for by) agents in R_t. The food rescue platform's existing allocation system does not elicit any signals of preferences or interest. However, if this information was available we could incorporate it in a similar way, for example, if I_t is the set of agents interested in item t the item would be announced to $R_t \cap I_t$.

Let Δ_n denote the simplex $\Delta_n = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1\}$. Denote with $x^t \in \Delta^n$ the allocation of the t-th item, where x_i^t is the fraction of item t received by i. Let $X^t \in \mathbb{R}^{n\times t}$ be the matrix with x^k as its k-th column, for $k = 1, \ldots, t$. Let $X = X^m$, the final allocation. For logistical reasons, donations are allocated in their entirety to a single recipient, so eventually $x^t \in \{0, 1\}^n$. However, when the item arrives and before it is assigned, it will be useful to think of x^t as fractional with x_i^t representing the probability that i receives item i under a particular allocation mechanism.

2.1 Valuation functions and Rawlsian justice criteria

Given realized assignments of the m items, X^m, the food rescue platform keeps track of several metrics in its efforts to ensure a reasonable distribution of the donations among recipients. Principal among these are

- The number of items recipient i received, with $v_i^{\mathrm{N}}(X) = \sum_{j=1}^m X_{ij}$.
- The total poundage received by recipient i, with $v_i^{\mathrm{LB}}(X) = \sum_{j=1}^m X_{ij} s_j$.
- The fraction of recipient i's demand that is fulfilled $v_i^{\mathrm{D}}(X) = \sum_{j=1}^m X_{ij} s_j / d_i$.

We focus on these valuation functions but note that our results apply to any valuation function linearly increasing in each of the X_{ij}. At intermediate points of the process, say after the arrival of j items, x^j may be fractional (representing the expected allocation) and $x^k = 0$ for those $j < k \leq m$ items that are yet to arrive.

The food rescue organization is interested in Rawlsian welfare, as measured by the welfare of the worst-off recipient. If all items were known in advance, we could formulate this problem as $\max_X \min_{i\in\mathcal{N}} v_i(X)$. Given the online nature of the problem and the fact that previous allocations are irreversible, we instead consider a sequence of single stage decision problems: upon the arrival of the t-th item we wish to solve $\max_{x^t\in\Delta_n} \min_{i\in R_t} v_i(X^{t-1}, x^t)$. Notice the minimization is over only those agents eligible to receive item t.

2.2 Exponential response rates and FCFS allocations

The food rescue platform currently allocates donations on a first come first served basis. Arriving items are announced to recipients in R_t and allocated to the first recipient who claims it.

We model this by associating each recipient i with *response rate*, modelled with an exponential distribution with parameter $\lambda_i > 0$. Let $\tau_{i,t}$ be the time that passes between i hearing of item t and moving to claim it. We assume $\tau_{i,t} \sim \mathrm{Exp}(\lambda_i)$ when $i \in R_t$ (i wants donation t and it is within i's pickup radius), and set $\tau_{i,t} = \infty$ otherwise. When $\tau_{i,t} \sim \mathrm{Exp}(\lambda_i)$, it is known that the expected response time $\mathbb{E}[\tau_i] = 1/\lambda_i$. For a subset of participants $S \subseteq \mathcal{N}$ with exponential response rates, let $\lambda_S = \sum_{i\in S} \lambda_i$. A property of the exponential distribution is that, for any such subset, $\tau_S = \min_{i\in S}\{\tau_i\} \sim \mathrm{Exp}(\lambda_S)$, so $\mathbb{E}[\min_{i\in S} \tau_i] = 1/\lambda_S$. Furthermore, the probability that any $i \in S$ responds first and receives an item is λ_i/λ_S. In particular, in the first-come first-served regime outlined above where each item is announced to recipients in R_t, $x_i^t = \lambda_i/\lambda_{R_t}$.

Example 1. *Consider three recipients with response rates $\lambda_i = i$, for $i = 1, 2, 3$, all able to pick up the donation. In the basic FCFS setting, with exponentially distributed response times, $x_i^{FCFS} = \lambda_i/(\lambda_1 + \lambda_2 + \lambda_3)$, as a result, $x^{FCFS} = (\frac{1}{6}, \frac{2}{6}, \frac{3}{6})$.*

2.3 Priority lists

One approach available to the food rescue platform is to solve $\max_{x^t\in\Delta_n} \min_{i\in R_t} v_i(X^{t-1}, x^t)$ for the fractional allocation that maximizes the Rawlsian objective at the arrival of every time t, and then randomize accordingly to decide who receives the item. We call the ideal (fractional) assignment of the current item x^* and the resulting objective function value z^*. In the remainder of this section we consider a generic item within the pick-up radius of $R \subseteq \mathcal{N}$. Such explicit randomization may easily appear unfair, and this approach also ignores any real-time constraints or demands from the recipient orgnaizations.

Instead, the FRP wishes to augment their existing FCFS system with a priority list. Before defining priority lists it is necessary to fix notation to speak about permutations. For permutations $\pi : [n] \to [n]$ over the recipients, $\pi(1)$ is the position of recipient 1 and $\pi^{-1}(1)$ is the recipient in position 1 — the one with highest priority. We use $\overline{\pi} \triangleq \pi^{-1}$ and write permutations in word representation (omitting braces as conventional), so $\pi = 231$ means that recipient 3 is placed first on the priority list. It is often more convenient to refer to the inverse permutation $\overline{\pi} = 312$, since recipients appear here in order of priority, with the highest priority recipient appearing first. For vector $v \in \mathbb{R}^n$ and permutation π, let $\pi * v = (v_{\pi(1)}, \dots, v_{\pi(n)})$, in particular, the components of $\overline{\pi} * v$ appear in the order of priority.

An n-stage priority list is defined as $(\pi, \vec{t} = (t_1, \dots, t_n))$, where π is a permutation of the recipients based on priority and $\vec{t}$ a sequence of time intervals. An n-stage priority list $(\pi, \vec{t})$ announces the available donation to one agent at a time in order of $\overline{\pi}$. Let t_k be the time between the donation being announced to $\overline{\pi}(k)$ and $\overline{\pi}(k+1)$, with $t_0 = 0$ and $t_n = \infty$. Call the time interval $[\sum_{t=0}^{i-1} t_i, \sum_{t=0}^{i} t_i]$ period i, so that $\overline{\pi}(i)$ and all $\overline{\pi}(i'), i' < i$ are aware of the item in period i. To account for the radius R we further constrain π so that $\overline{\pi}(i) < \overline{\pi}(j)$ for all $i \in R$, $j \in \mathcal{N} \setminus R$ and set $t_{|R|} = \infty$, so recipients able to pick the donation up is given highest priority and the item is allocated to one of them with probability 1. When the ordering π is clear from the context, we assume recipients are relabeled according to π, so that the item is announced to i at the beginning of period i.

Perhaps the simplest implementation of a priority list scheme is a 2-stage (or binary) priority list (L, T_L) defined by a subset of recipients $L \subseteq R$ and a time T_L. Upon arrival, the item is announced to the set of recipients L. If it remains unclaimed at time T_L, the remainder of the recipients in $R \setminus L$ are also notified of the item.

Recall Δ_n denotes the simplex. Employing a (n-stage or 2-stage) priority list results in an expected (fractional) allocation $x^t \in \Delta_n$ where x_i^t is the probability that i receives the donation. Using a priority list appears to limit the amount of control that the food rescue platform has over x^t. However, by carefully controlling when each recipient finds out about the available item, there is still hope that the resulting expected allocation x^t is close to x^* and $z = \min_{i \in R} v_i(X^{t-1}, x^t)$ is close to z^*.

3 GENERAL n-STAGE PRIORITY LISTS

Consider the problem facing the food rescue platform when the t-th item arrives. The first $t-1$ items are already allocated, with $X_{ij}^{t-1} = 1$ when the j-th item went to i. The food rescue platform wishes to maximize the expected value of the worst-off recipient. New item t can only be received by recipients in R_t, so $x_i^t = 0$ for $i \in \mathcal{N} \setminus R_t$ and, for the valuation functions we consider, $v_i(X^{t-1}, x^t) = v_i(X^{t-1})$. Since agents outside R_t are not eligible to receive t, let

$$z^* = \max_{x^t \in \Delta_n} \min_{i \in R_t} v_i(X^{t-1}, x^t).$$

In this single stage problem, recipients in $\mathcal{N} \setminus R_t$ are completely irrelevant: the donation is not announced or allocated to them, and they do not affect the objective function. As a result, we may restrict the problem only to recipients in R_t; for notational convenience we make the equivalent assumption that $\mathcal{N} = R_t$. We also suppress the dependence on t and write X to represent the previous allocation instead of X^{t-1} and x for the allocation of the new item instead of x^t. Now

$$z^* = \max_{x^t \in \Delta_n} \min_{i \in \mathcal{N}} v_i(X, x).$$

Practically, the food rescue platform does not get to decide x. Instead, it decides on an allocation policy, in this case an n-stage priority list $L = (\pi, \vec{t})$, which results in some expected allocation x^L. We write this as

$$z^{\mathrm{PL}} = \max_L \min_{i \in \mathcal{N}} v_i(X, x^L).$$

Our main result in this section is that, for the valuation functions we care about, the food rescue platform does not give up any flexibility when it commits to using an n-stage priority list rather than unilaterally enforcing a (randomized) assignment, in other words, $z^* = z^{\mathrm{PL}}$.

3.1 Finding the optimal fractional allocation

Given previous allocation X, let s be the quantity of the newly arrived item and x^* the optimal solution to $\max_{x \in \Delta_n} \min_{i \in \mathcal{N}} v_i(X, x)$.

We show that x^* can be found with a simple water-filling algorithm, Algorithm 1, for $v = v^{\mathrm{D}}$. The same algorithm also yields x^* for $v = v^{\mathrm{LB}}$ by setting $d_i = 1$ for all $i \in \mathcal{N}$, and for $v = v^{\mathrm{N}}$ by setting $s = 1$ for all items. The algorithm considers the agents in order from lowest value on the X to highest. First, it adds recipient 1 (the recipient with lowest value) to the set of active recipients (those that determine the objective function value) N. Then it determines how much the value of 1 must increase to equal the value of the second highest agent and, assuming it is possible, increases x_1 until the values of agents 1 and 2 are equal. Now agent 2 is added to the set of active recipients, the algorithm checks the increase required to reach the value of the third highest agent and raises agents 1 and 2 simultaneously. At some point it may not be possible to increase all the active agents to match the next value, in which case their values are increased as much as possible until $\sum_{i \in N} x_i = 1$.

Algorithm 1: Water-filling algorithm to find $x^* = \arg\max_{x \in \Delta_n} \min_{i \in \mathcal{N}} v_i^{\mathrm{D}}(X, x)$

Data: Demands d_i for $i \in \mathcal{N}$. Previous allocations X. Agents labelled so that $v_1(X) \leq v_2(X) \leq \cdots$. Item of size s.
Result: Fractional allocation $x \in \Delta_n$ maximizing $\min_{i \in \mathcal{N}} v_i^{\mathrm{D}}(X, x) = \min_{i \in \mathcal{N}} (\sum_{j=1}^{t-1} X_{ij} s_j + x_i s)/d_i$

$x \leftarrow 0$; $i \leftarrow 1$; $X_{\mathrm{remain}} = 1$;
$v_{n+1}^D(X) \leftarrow \max_{i \in \mathcal{N}} v_i^{\mathrm{D}}(X) + s/\min_{i \in \mathcal{N}} d_i$; `/* largest value achievable */`
$N \leftarrow \{1\}$; `/* set of agents with minimum value */`
while $X_{remain} > 0$ **do**
  $\delta = v_{i+1}^{\mathrm{D}}(X) - (v_1^{\mathrm{D}}(X) + \frac{x_1 s}{d_1})$; `/* gap to next lowest value */`
  if $\sum_{i \in N} \frac{\delta \cdot d_i}{s} \leq X_{remain}$ **then** `/* increase value of all agents in` N `to match` $i+1$`'s */`
    $x[N] \leftarrow x[N] + \frac{\delta}{s} \cdot d_i$
    $X_{\mathrm{remain}} \leftarrow X_{\mathrm{remain}} - \sum_{i \in N} \frac{\delta}{s} \cdot d_i$
    $i \leftarrow i + 1$
    $N \leftarrow N \cup \{i\}$
  else `/* increase values proportional to what remains */`
    $x[N] \leftarrow x[N] + \frac{X_{\mathrm{remain}}}{\sum_{i \in N} \frac{\delta \cdot d_i}{s}} \cdot \frac{\delta}{s} \cdot d_i$
    $X_{\mathrm{remain}} \leftarrow 0$

We illustrate the working of the algorithm on a small example.

Example 2. *Consider an instance with three recipients with $d_1 = 5, d_2 = d_3 = 10$ and $v_1^D(X) = 1, v_2^D(X) = 4$ and $v_3^D(X) = 10$. Suppose the arriving item has size 30. Initially $x = 0$. The algorithm starts by increasing x_1 until 1's value is equal to 2's. This happens at $x_1 = 0.5$, where $1 + 0.5 \cdot 30/5 = 4$. Now 1 and 2's values are increased simultaneously until it* *reaches* $v_3^D(X)$ *or the entire item is allocated. Here the latter happens and the algorithm terminates with* $x_1 = \frac{4}{6}$, $x_2 = \frac{2}{6}$ *and* $v_1^D(X,x) = 1 + (\frac{4}{6} \cdot 30)/5 = 5 = 4 + (\frac{2}{6} \cdot 30)/10 = v_2^D(X,x)$.

Theorem 3. *The output of Alg 1, denoted* x^**, is an optimal solution to* $\max_{x \in \Delta_n} \min_{i \in N} v_i^D(X,x)$.

Before proving Theorem 3, we establish the following fact about Algorithm 1.

Lemma 4. *At the end of any iteration of Algorithm 1,* $v_i^D(X,x) = z(x)$ *for all* $i \in N$.

Proof of Lemma 4. Call N at the end of iteration i, N_i, and similarly use x^i, X^i_{remain} to refer to x and X_{remain}, respectively, at that point. We prove by induction that at the end of iteration k, $v_i^{\text{D}}(X,x^k) = z(x^k)$ for all $i \in N_k$.

As base case for the induction, consider iteration 1 of Algorithm 1. If $N_1 = \{1\}$, it means that δ is large enough that $\frac{\delta \cdot d_1}{s} > X_{\text{remain}} = 1$. In this case $x_1 < \frac{\delta \cdot d_1}{s}$, implying that

$$v_1^{\text{D}}(X,x_1) = v_1^{\text{D}}(X) + \frac{x_1 s}{d_1} < v_1^{\text{D}}(X) + \delta = v_1^{\text{D}}(X) + (v_2^{\text{D}}(X) - (v_1^{\text{D}}(X) + \frac{0 \cdot s}{d_1})) = v_2^{\text{D}}(X).$$

Recipients are labelled in order of increasing values, so this implies $z(x^1) = v_i^{\text{D}}(X,x^1)$, as required. Suppose instead that $N_1 = \{,21\}$. Now $\frac{\delta \cdot d_1}{s} < X_{\text{remain}} = 1$ and $x_1^1 = \frac{\delta \cdot d_1}{s}$ so that

$$v_1^{\text{D}}(X,x_1) = v_1^{\text{D}}(X) + \frac{x_1 s}{d_1} = v_1^{\text{D}}(X) + \delta = v_1^{\text{D}}(X) + (v_2^{\text{D}}(X) - (v_1^{\text{D}}(X) + \frac{0 \cdot s}{d_1})) = v_2^{\text{D}}(X).$$

Since 1 had smallest value, 2 second smallest value and for all $j \neq 1$, $x_j^1 = 0$, we again obtain the required condition.

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 $j \geq k$ and (by the induction hypothesis) $v_i(X,x^{k-1}) = v_j(X,x^{k-1})$ for all $i,j \in N^{k-1}$. N^{k-1} takes two possible values: it either includes k or not depending on the result of the 'if' statement in the previous iteration.

If $k \notin N^{k-1}$, then $X_{\text{remain}} = 0$ and the algorithm terminates with final $N = N^{k-1}$ and $x = x^{k-1}$, and the induction hypothesis continues to hold. Suppose instead that $k \in N^k$. Now $v_1(X,x^{k-1}) = v_j(X,x^{k-1})$ for all $j \in N^{k-1} = \{1,\ldots,k\}$. Line 5 computes $\delta = v_{k+1}^{\text{D}}(X,x^{k-1}) - v_1^{\text{D}}(X,x^{k-1})$. If line 6 evaluates to true, there is enough of the item remaining to increase the allocation of all agents in N^{k-1} so that their values match $k+1$'s. After the value update,

$$v_j^{\text{D}}(X,x^k) = v_j^{\text{D}}(X,x^{k-1}) + \frac{(x^k - x^{k-1})s}{d_j} = v_j^{\text{D}}(X,x^{k-1}) + \frac{\delta d_j}{s}\frac{s}{d_j} = v_j^{\text{D}}(X,x^{k-1}) + v_{k+1}^{\text{D}}(X,x^{k-1}) - v_1^{\text{D}}(X,x^{k-1}),$$

for any $j \in N^{k-1}$; it follows from the induction hypothesis that $v_j^{\text{D}}(X,x^k) = v_{k+1}^{\text{D}}(X,x^{k-1})$. Values don't decrease during the remainder of the iteration, and $N^k = N^{k-1} \cup \{k+1\}$. We conclude that, since recipients are ordered by increasing values, $z(x^k) = v_j^{\text{D}}(X,x^k)$ for every $j \in N^{k-1}$.

It is also possible that line 6 evaluates to False. We perform the computation as before and see that

$$v_j^{\text{D}}(X,x^k) - v_j^{\text{D}}(X,x^{k-1}) = \delta \cdot \frac{X_{\text{remain}}}{\sum_{i \in N^{k-1}} \frac{\delta \cdot d_i}{s}},$$

which is a constant. No further updates are made to x^k or $N^k = N^{k-1}$. Since the values of all recipients in N^{k-1} start equal (by the induction hypothesis) and increase by the same constant, their values at the end of iteration k remain equal. Moreover, $v_j^{\text{D}}(X,x^k) < v_{k+1}^{\text{D}}(X,x^k)$ for any $j \in N^k$, so $z(k) = v_j^{\text{D}}(X,x^k) = v_1^{\text{D}}(X,x^k)$. □

Now we are ready to prove that Algorithm 1 returns an optimal fractional allocation.

Proof of Theorem 3. Consider output x^* with objective function value z^*. Call $N^* = \{i \in \mathcal{N} : v_i^{\mathrm{D}}(X, x^*) = z^*\}$ the set of active recipients, those that have values equal to the objective function value.

Suppose for contradiction there exists an alternative fractional allocation $x' \in \Delta_n$ with objective function value $z' > z^*$. For every $i \in N^*$, it holds that

$$\frac{\sum_{j=1}^{t-1} X_{ij} s_j + x_i' s}{d_i} = v_i^{\mathrm{D}}(X, x') \geq \min_{k \in \mathcal{N}} v_k^{\mathrm{D}}(X, x') = z' > z^* = v_i^{\mathrm{D}}(X, x^*) = \frac{\sum_{j=1}^{t-1} X_{ij} s_j + x_i^* s}{d_i},$$

in other words, $x_i' > x_i^*$ and each active recipient must receive a strictly larger fraction of the item in x' than in x^*.

Since $\sum_{i \in \mathcal{N}} x_i' = 1 = \sum_{i \in \mathcal{N}} x_i^*$, this implies there is some $i \in \mathcal{N} \setminus N^*$ for which $x_i' < x_i^*$. Since $x' \geq 0$, this requires that there is some inactive recipient $k \in \mathcal{N} \setminus N^*$ for who $v_k^{\mathrm{D}}(X, x^*) > z^*$ and $x_k^* > 0$. Alg 1 only increases the assignments of the recipients in the set N, so at some point during execution k must have been added to N.

By Lemma 4 and the fact that recipients are never removed from N, $k \in N$ at termination and thus $v_k^{\mathrm{D}}(X, x^*) = z^*$, contradicting $k \in \mathcal{N} \setminus N^*$. We conclude that x^* is optimal. □

3.2 Priority lists can achieve all fractional allocations

In the previous section we show that it is possible to find the fractional allocation which maximizes the value of the worst-off recipient. In this section we sometimes call such a fractional allocation a *target allocation* and denote it with $\hat{x}$. Because the allocation process, even under priority lists, remains inherently an FCFS procedure that depends on agent's response rates, there is no guarantee that any particular target allocation is achievable within our framework. In particular, the order of an n-stage priority list impacts which target allocations are feasible.

Example 5. *Consider the same three recipients as before with response rates $\lambda_i = i$, for $i = 1, 2, 3$. A simple fairness requirement is to ask that each recipient has the same probability to receive the current item, which has target allocation $\hat{x} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.*

Consider an n-stage priority list which orders the agents $\overline{\pi} = 231$. Recipient 1 has the largest probability of receiving the item when it is immediately announced to all three recipients, so for any vector of announcement times $\vec{t}$. $x_1^{PL\text{-}231} \leq \frac{1}{6}$. This shows $\hat{x}$ is not feasible for this permutation.

Consider, instead, $\overline{\pi} = 123$. Now $\hat{x}$ is feasible, and achieved with announcement times $\vec{t} = (0, -\ln\frac{5}{6}, -\frac{1}{3}\ln\frac{5}{6})$. We verify that this priority list results in the target allocation. Period 1 lasts $-\ln\frac{5}{6} \approx 0.182$ time units, the probability that recipient claims the item before this time is $\mathbb{P}(\tau_i \leq -\ln\frac{5}{6}) = 1 - \exp(-1 \cdot (-\ln\frac{5}{6})) = \frac{1}{6}$. At the end of the first period, the expected fractional allocation is therefore $x^{PL\text{-}123} = (\frac{1}{6}, 0, 0)$. By the memorylessness of the exponential distribution and the fact that the minimum of multiple exponentially distributed variables are again distributed exponentially, we analyze period 2 similarly, $\mathbb{P}(\min(\tau_1, \tau_2) \leq t_2) = 1 - \exp(-3 \cdot (-\frac{1}{3}\ln\frac{5}{6}) = \frac{1}{6}$. Conditioned on the item being claimed in period 2, 1 receives it with probability $\frac{\lambda_1}{\lambda_1 + \lambda_2} = 1/3$. As a result, at the end of period 2, $x^{PL\text{-}123} = (\frac{1}{6} + \frac{1}{18}, \frac{2}{18}, 0)$. Finally, the item is claimed with remaining probability $2/3$ during period 3 and, conditioned on this happening, it goes each recipient proportionally to their response rates. We conclude that $x^{PL\text{-}123} = (\frac{1}{6} + \frac{1}{18} + \frac{1}{6}\frac{2}{3}, \frac{2}{18} + \frac{2}{6}\frac{2}{3}, \frac{3}{6}\frac{2}{3}) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

In the previous example, the priority list on $\overline{\pi} = 123$ is able to assign the item with equal probability to each agent because the recipients appear in the permutation ordered from slowest response rate to fastest. This gives slower recipients a period of exclusive access to the item where they are not forced to compete with the faster recipients. Ordering recipients from slowest to fastest always works when $\hat{x} = (\frac{1}{n}, \dots, \frac{1}{n})$. This is a special case of a more general result to follow, so we state it without proof for now.

Proposition 6. *The balanced target allocation $\hat{x} = (\frac{1}{n}, \dots, \frac{1}{n})$ is feasible for a priority list with permutation π satisfying $\lambda_{\overline{\pi}(1)} \leq \lambda_{\overline{\pi}(2)} \leq \cdots \leq \lambda_{\overline{\pi}(n)}$.*

However, ordering recipients from slowest to fastest does not achieve all target allocations. For example, consider the recipients from Example 5 with $\overline{\pi} = 123$ and suppose that $\hat{x}_3 = \frac{2}{3}$. Recipient 3 has greatest probability of receiving the item when it is immediately announced to all recipients ($t_1 = t_2 = 0$), but even then $x_3^{\text{PL-123}} = \frac{1}{2}$. Achieving $\hat{x}_3 = \frac{2}{3}$ requires an ordering in which 3 does not appear last. Let X^π denote the allocations achievable with a priority list on permutation π. The following result establishes, for a particular order, which target allocations are feasible.

Theorem 7. *The feasible allocations of a general n-stage priority list on π is the convex hull of $\{a_1, \dots, a_n\}$, i.e. $X^\pi = Conv(a_1, \dots, a_n)$, where $\overline{\pi} * a_1 = (1, 0, \dots, 0) \in \mathbb{R}^n$, $\overline{\pi} * a_i = (\frac{\lambda_{\overline{\pi}(1)}}{\sum_{j=1}^{i} \lambda_{\overline{\pi}(j)}}, \dots, \frac{\lambda_{\overline{\pi}(i)}}{\sum_{j=1}^{i} \lambda_{\overline{\pi}(j)}}, 0, \dots, 0) \in \mathbb{R}^n$ and $\overline{\pi} * a_n = (\frac{\lambda_{\overline{\pi}(1)}}{\lambda_N}, \dots, \frac{\lambda_{\overline{\pi}(n)}}{\lambda_N})$.*

Proof. We first show $\text{Conv}(a_1, \dots, a_n) \subseteq X^\pi$. Consider arbitrary $\hat{x} \in \text{Conv}(a_1, \dots, a_n)$. It is possible to write $\hat{x}$ as a convex combination $\hat{x} = \sum_i c_i a_i$ with $\sum_i c_i = 1$. We may interpret c_i as the probability that the item is claimed/allocated in period i. This gives a sequence of n equations

$$\begin{aligned}
c_1 &= 1 - \exp(-\lambda_{\overline{\pi}(1)} t_1) \\
c_2 &= (1 - c_1)(1 - \exp(-(\lambda_{\overline{\pi}(1)} + \lambda_{\overline{\pi}(2)}) t_2)) \\
c_3 &= (1 - c_1)(1 - \frac{c_2}{1 - c_1})[1 - \exp(-\sum_{j=1}^{3} \lambda_{\overline{\pi}(i)} t_3)] = (1 - c_1 - c_2)(1 - \lambda_{\overline{\pi}(i)} \exp(-\sum_{j=1}^{3} \lambda_{\overline{\pi}(i)} t_3)) \\
&\vdots \\
c_n &= (1 - \sum_{i=1}^{n-1} c_i)(1 - \exp(-\lambda_N t_n)) = 1 - \sum_{i=1}^{n-1} c_i
\end{aligned}$$

with unknowns $t_1, \dots, t_n$. We can solve this system of equations to get $t_1 = (\ln(1 - c_1))/(-\lambda_{\overline{\pi}(1)})$ and, for $2 \leq k \leq n$,

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

It remains to check that these times are valid for a priority list, in particular, that $t_i \geq 0$ for all $i \in [n]$. Clearly $t_1 \geq 0$, since $0 < 1 - c_1 < 1$ so $\ln(1 - c_1) < 0$. Similarly, for $2 \leq k \leq n$, $1 - \sum_j^k c_j < 1 - \sum_j^{k-1} c_j$, from which it follows that $t_k \geq 0$. We conclude that $\hat{x}$ is achievable by a priority list on π, and $\text{Conv}(a_1, \dots, a_n) \subseteq X^\pi$.

It remains to show that $X^\pi \subseteq \text{Conv}(a_1, \dots, a_n)$. Consider arbitrary $\hat{x} \in X^\pi$, achieved by priority list on π with times $t_1, \dots, t_n$. Let c'_i be the probability that this priority list allocates the item in period i. Now we can write the probability agent $\overline{\pi}(1)$ receives the item as

$$(\overline{\pi} * \hat{x})_1 = \hat{x}_{\overline{\pi}(1)} = c'_1 + \frac{\lambda_{\overline{\pi}(1)}}{\lambda_{\overline{\pi}(1)} + \lambda_{\overline{\pi}(2)}} c'_2 + \cdots + \frac{\lambda_{\overline{\pi}(1)}}{\sum_{i=1}^{n} \lambda_{\overline{\pi}(i)}} c'_n = \sum_{i=0}^{n} (\overline{\pi} * a_i)_1 c'_i$$

and similarly for any agent

$$\hat{x}_{\overline{\pi}(k)} = \sum_{i=k}^{n} \frac{\lambda_{\overline{\pi}(k)}}{\sum_{j=1}^{i} \lambda_{\overline{\pi}(j)}} c'_i = \sum_{i=k}^{n} (a_i)_k c'_i = \sum_{i=0}^{n} (\overline{\pi} * a_i)_k c'_i,$$

where the final transition follows from $(\overline{\pi} * a_i)_k = 0$ for $i < k$. It follows that $\hat{x} \in \mathrm{Conv}(a_1, \ldots, a_n)$, and $X^\pi \subseteq \mathrm{Conv}(a_1, \ldots, a_n)$ as required. □

Continuing with the recipients in Example 5, this result implies that $X^{123} = \mathrm{Conv}\{(1,0,0), (\frac{1}{3}, \frac{2}{3}, 0), (\frac{1}{6}, \frac{2}{6}, \frac{3}{6})\}$. Clearly $\hat{x}_3 = \frac{2}{3}$ implies $\hat{x} \notin X^{123}$. On the other hand, for permutation $\overline{\pi} = 132$ which does not place 3 last in the ordering, $X^{132} = \mathrm{Conv}\{(1,0,0), (\frac{1}{4}, 0, \frac{3}{4}), (\frac{1}{6}, \frac{2}{6}, \frac{3}{6})\}$ so $\hat{x}_3 = \frac{2}{3}$ is achievable. This example shows that both a recipient's response rate and the magnitude of their target allocation must be considered when choosing a priority list permutation.

Our main result in this section is that any target allocation $\hat{x}$ is feasible for a priority list with permutation $\pi(\hat{x}, \vec{\lambda})$, where $\pi \triangleq \pi(\hat{x}, \vec{\lambda})$ prioritizes recipients in decreasing order of $\frac{\hat{x}_i}{\lambda_i}$, i.e. $\frac{\hat{x}_{\pi(1)}}{\lambda_{\pi(1)}} \geq \frac{\hat{x}_{\pi(2)}}{\lambda_{\pi(2)}} \geq \cdots$. Proposition 6 follows as a direct corollary: when $\hat{x}$ is uniform, recipients should be prioritized by decreasing $1/\lambda_i$, in other words, from slowest to fastest. To ease notation we suppress the dependence on π in the remainder of the section and instead relabel recipients by priority in π. So after relabelling recipient $\overline{\pi}(1)$ becomes recipient 1, with target allocation $\tilde{x} = \overline{\pi} * \hat{x}$ (so $\tilde{x}_i = \hat{x}_{\overline{\pi}(i)}$), response rates $\tilde{\lambda} = \overline{\pi} * \lambda$, etc. After relabelling, $\frac{\tilde{x}_1}{\tilde{\lambda}_1} \geq \frac{\tilde{x}_2}{\tilde{\lambda}_2} \geq \cdots \frac{\tilde{x}_n}{\tilde{\lambda}_n}$.

Theorem 8. *Given target fractional allocation $\hat{x} \in \Delta_n$, relabel the instance so that $\tilde{x} = \overline{\pi} * \hat{x}$, $\tilde{\lambda} = \overline{\pi} * \lambda$ and $\frac{\tilde{x}_1}{\tilde{\lambda}_1} \geq \frac{\tilde{x}_2}{\tilde{\lambda}_2} \geq \cdots \frac{\tilde{x}_n}{\tilde{\lambda}_n}$. Then $\tilde{x}$ is feasible in the relabelled instance for the priority list $(\mathbf{1}, (t_1, \ldots, t_n))$, where $\mathbf{1}$ is the identity permutation and*

$$t_k = \frac{-\ln\left(1 - \frac{c_k}{\left(1 - \sum_{j=1}^{k-1} c_j\right)}\right)}{\sum_{j=1}^{k} \tilde{\lambda}_k} \quad \textit{and} \quad c_k = \left(\sum_{j=1}^{k} \tilde{\lambda}_j\right)\left[\frac{\tilde{x}_k}{\tilde{\lambda}_k} - \frac{\tilde{x}_{k+1}}{\tilde{\lambda}_{k+1}}\right].$$

Proof. As before, let c_k be the probability that the item is allocated during period k. Let $c_{k,i}$, for $i \in \mathcal{N}$, be the probability that the item is allocated to i during period k. Consider any period k during which $N^k \subseteq \mathcal{N}$ are aware of the time. For any recipients $i, j \in N^k$,

$$c_{k,i} = \frac{\tilde{\lambda}_i}{\tilde{\lambda}_{N^k}} c_i = \frac{\tilde{\lambda}_i}{\tilde{\lambda}_j} \frac{\tilde{\lambda}_j}{\tilde{\lambda}_{N^k}} c_i = \frac{\tilde{\lambda}_i}{\tilde{\lambda}_j} c_{k,j}.$$

We use this relationship between recipients k and $k+1$ to obtain, for $1 \leq k < n$,

$$\tilde{x}_k = \frac{\tilde{\lambda}_k}{\sum_{j=1}^{k} \tilde{\lambda}_j} c_k + \frac{\tilde{\lambda}_k}{\tilde{\lambda}_{k+1}} \tilde{x}_{k+1} \Rightarrow c_k = \left(\sum_{j=1}^{k} \tilde{\lambda}_j\right)\left[\frac{\tilde{x}_k}{\tilde{\lambda}_k} - \frac{\tilde{x}_{k+1}}{\tilde{\lambda}_{k+1}},\right] \tag{1}$$

and, for recipient n, $c_n = \sum_{j=1}^{n} \tilde{\lambda}_j \frac{\tilde{x}_n}{\tilde{\lambda}_n}$. Notice that, since recipients are ordered in decreasing order of $\frac{\tilde{x}_k}{\tilde{\lambda}_k}$, $c_k \geq 0$ and $c_k = 0$ only when $\frac{\tilde{x}_k}{\tilde{\lambda}_k} = \frac{\tilde{x}_{k+1}}{\tilde{\lambda}_{k+1}}$ (implying that k and $k+1$ should receive simultaneous access to the item) or $\tilde{x}_k = 0$, implying k (and subsequent recipients) should not get access to the item. We can confirm that $\sum_{j=1}^{n} c_j = 1$.

Given c_k, we can compute t_k. Write $\mathcal{E}_k$ for the event that the item is allocated/claimed in period k. Now

$$\begin{aligned} c_k = \mathbb{P}[\mathcal{E}_k] &= \mathbb{P}[\mathcal{E}_k \text{ and none of } \mathcal{E}_1, \ldots, \mathcal{E}_{k-1} \text{ happened}] \\ &= (1 - \sum_{j=1}^{k-1} \mathbb{P}\left[\mathcal{E}_j\right]) \cdot \mathbb{P}[\mathcal{E}_k | \text{none of } \mathcal{E}_1, \ldots, \mathcal{E}_{k-1} \text{ happened}] \\ &= \left(1 - \sum_{j=1}^{k-1} c_j\right) \cdot \mathbb{P}\left[\tau \sim \mathrm{Exp}\left(\sum_{i=1}^{k} \tilde{\lambda}_i\right) \leq t_k\right] = \left(1 - \sum_{j=1}^{k-1} c_j\right)\left(1 - \exp\left(-t_k \sum_{j=1}^{k} \tilde{\lambda}_j\right)\right). \end{aligned}$$

Combining this with (1) yields

$$t_k = \frac{-\ln\left(1 - \frac{c_k}{\left(1-\sum_{j=1}^{k-1} c_j\right)}\right)}{\sum_{j=1}^{k} \tilde{\lambda}_k}.$$

Notice $t_k > 0$ when $c_k > 0$, which happens whenever $\frac{\tilde{x}_k}{\tilde{\lambda}_k} > \frac{\tilde{x}_{k+1}}{\tilde{\lambda}_{k+1}}$. When $\frac{\tilde{x}_k}{\tilde{\lambda}_k} = \frac{\tilde{x}_{k+1}}{\tilde{\lambda}_{k+1}}$, $t_k = 0$ so that k and $k+1$ get simultaneous access to the item. When $\tilde{x}_{k-1} > 0$ and $\tilde{x}_k = 0$, $t_k = \infty$, from which we conclude that the item will never be announced to k (nor $k+1, \dots, n$). Finally, when both $\tilde{x}_{k-1} = 0$ and $\tilde{x}_k = 0$, $t_k = 0$ 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 item.

We conclude that these times are valid ($t \geq 0$) and, together with π, implement an n-stage priority list. □

Another way of stating this result it that, for any $\hat{x} \in \Delta_n$, there exists an π such that $\hat{x} \in X^{\pi}$, moreover, π is easy to find. As a result, we may solve $\max_{x \in \Delta_n} \min_{i \in \mathcal{N}} v_i(X, x)$ (for any v) directly ignoring the fact that we are constrained to implementing a n-stage priority list, as we do in Algorithm 1.

4 BINARY PRIORITY LISTS

Recall that a 2-stage or binary priority list is defined by a set of recipients L and a time t. The new item is first announced to recipients in L, if it remains unclaimed by time t it is announced to the remainder of the recipients. Claims are FCFS, whoever responds first gets the item. In contrast to n-stage priority lists we immediately observe that binary priority lists do limit the space of feasible target allocations.

Example 9. *Consider the instance of Example 5 with binary priority lists. Recall there are three recipients with response rates $\lambda_i = i$, for $i = 1, 2, 3$ and the target allocation is $\hat{x} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.*

The only hope of equalizing the allocation is to announce it first to the slowest agent, 1, so we may assume $1 \in S$. Suppose $S = \{1\}$, this implies the item is simultaneously announced to 2 and 3, since 3 has a faster response rate it is impossible for their allocation to be equal. If, instead, at least one of 2 and 3 are included in S, then we can argue similarly that they will have a larger fractional allocation than 1. We conclude that $\hat{x} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ is not feasible with priority lists.

This creates a dilemma, we can find the optimal fractional allocation x^* with Algorithm 1, but there may not be a binary priority list that achieves it. And even if we could find the binary priority list with fractional allocation closest to x^* (say, by some norm), there is no guarantee this is the binary priority list which maxmimizes

$$z^*_{\text{BPL}} = \max_{L,t} \min_{i \in \mathcal{N}} v_i(X, x^{BPL(L,t)}).$$

As a result, we may be forced to search over the space of all possible 2-stage priority lists including the exponentially many subsets of recipients.

Our main result in this section is that this is not necessary: Every possible value of z^*_{BPL} is achieved by one of n binary priority lists. This means that we can find the optimal binary priority list by binary search on z^*_{BPL} (with initial lower bound equal to the current value and upper bound equal to z^*).

Let $x_i^{BPL(L,t)} = \mathbb{P}\,[i$ claims the item first, given 2-stage PL $(L, t)]$. Now $x^{BPL(L,t)}$ is the fractional allocation that results from (L, t). When the priority list is clear form the context we write simply x instead of $x^{BPL(L,t)}$.

For any set of valuation functions $v_i, \dots, v_n$, each v_i strictly increasing in x_i, existing allocation X and objective function value z, let $\hat{x}^z = (\hat{x}_1^z, \dots, \hat{x}_n^z)$ be the minimal allocation for which $v_i(X, \hat{x}_i^z) = z$ for all $i \in [n]$. In other words,

$\hat{x}_i^z$ is the fraction of the new item i needs to receive to attain valuation function value z, and globally $\hat{x}^z$ is the target allocation to reach z.

Theorem 10. *Given current allocation X and any set of valuation functions $v_i, \ldots, v_n$, each v_i strictly increasing in x_i. For any optimal objective function value $z = \max_{(L,t)} \min_i v_i(X, x_i^{BPL(L,t)})$, find the permutation π which prioritizes recipients in decreasing order of $\hat{x}^z/\lambda$, so that $\frac{\hat{x}^z_{\overline{\pi}(1)}}{\lambda_{\overline{\pi}(1)}} \geq \frac{\hat{x}^z_{\overline{\pi}(2)}}{\lambda_{\overline{\pi}(2)}} \geq \cdots$.*

There exists an optimal priority list (L, t) in which $L = \{\overline{\pi}(1), \ldots, \overline{\pi}(k)\}$ for some $k \in [n]$.

We interpret this as evidence that binary priority lists are closely related to n-stage priority lists. Specifically, given an objective function value z you can find binary priority list that achieves it (or prove one doesn't exist) by taking the permutation π of an n-stage priority list with target allocation $\hat{x}^z$ and including recipients, one at a time in order of priority in π, in L. If none of these binary priority lists achieve z, then z is not feasible for binary priority lists.

Proof. Let $z = \max_{(L,t)} \min_i v_i(X, x_i^{BPL(L,t)})$ and compute the fractional allocation allocation $\hat{x}^z$. As before, relabel recipients (and the remainder of the instance) so that $\frac{\hat{x}^z_1}{\lambda_1} \geq \frac{\hat{x}^z_2}{\lambda_2} \geq \cdots$. After relabelling the theorem statement is that $L = [k]$ for some $k \in [n]$. For any binary priority list (L, t), let $V_{(L,t)} = \{(i, j) \in [n]^2 : i \in L, j \in \bar{L}, i > j\}$ count the number of pairwise violations of the condition above. Note that $|V_{(L,t)}| = 0$ when $L = [k]$ for some $k \in \mathbb{N}$. Suppose for contradiction that no such solution exists.

Let (L, t_L) be an optimal solution to $\max_{(L,t)} \min_i v_i(X, x_i^{BPL(L,t)})$ with smallest $|V_{(L,t_L)}| > 0$. Without loss of generality we may assume that $L \subset \mathcal{N}$ and $t > 0$. Let $\ell = \max\{i \in L\}$ and $k = \min\{i \in \bar{L}\}$ and observe that $(\ell, k) \in V_{(L,t_L)}$, otherwise $L = [k]$ for some $k \in \mathbb{Z}$.

Given solution (L, t_L), we can compute the probability that each agent receives the current item, which we denote with $x \triangleq x^{BPL(L,t_k)}$. Denote with $F_L(t_L) = \mathbb{P}[\tau \sim \text{Exp}(\lambda_L) \leq t_L]$ the probability that the item is allocated before time t_L. If $i \notin L$ then, conditioned on the item being available at time t_L, i receives it with probability $\frac{\lambda_i}{\lambda_N}$. When $i \in L$, i receives the item with probability $\frac{\lambda_i}{\lambda_L}$ if it is allocated by time t_L, otherwise i receives it with probability $\frac{\lambda_i}{\lambda_N}$. Summarizing,

$$x_i = \begin{cases} \lambda_i \left[\frac{F_L(t_L)}{\lambda_L} + \frac{1 - F_L(t_L)}{\lambda_N}\right], & i \in L \\ \lambda_i \cdot \frac{1 - F_L(t_L)}{\lambda_N}, & i \in \bar{L}. \end{cases}$$

Notice that $\frac{x_i}{\lambda_i} = \frac{x_j}{\lambda_j}$ for all $i, j \in L$, and similarly for all $i, j \in \bar{L}$. Additionally, $x \geq \hat{x}^z$.

Define $A = \{i \in \mathcal{N} : v_i(X, x_i) = z\}$ to be the set of *active* agents — those who determine the objective function value. We proceed as follows: First we show that some agent in $\bar{L}$ must be active, in fact, k is active. We then construct an alternative solution in which ℓ is removed from the priority list and show that it has objective function value at least as large as (L, t_L), and fewer violating pairs, a contradiction.

Suppose $\bar{L} \cap A = \emptyset$, so no agent in $\bar{L}$ is active. Then there exists some $\epsilon > 0$ so that $v_i(X, p_i - \epsilon) > z$ for all $i \in \bar{L}$. But now we can find $\delta > 0$ so that the priority list $(L, t_L + \delta)$ is feasible and improves over z, contradicting that (L, t_L) is optimal. We conclude that $\bar{L} \cap A \neq \emptyset$.

Let $k' = \min\{i : i \in \bar{L} \cap A\}$ and suppose for contradiction that $k' > k$. Then

$$\frac{x_k}{\lambda_k} > \frac{\hat{x}^z_k}{\lambda_k} \geq \frac{\hat{x}^z_{k'}}{\lambda_{k'}} = \frac{x_{k'}}{\lambda_{k'}},$$

where the first inequality follows from k not being active, the second from $k < k'$, and finally $\hat{x}^z_{k'} = x_{k'}$ from $k' \in A$. But this contradicts the fact that $\frac{x_a}{\lambda_a} = \frac{x_b}{\lambda_b}$ for all $a, b \in \bar{L}$.

Consider the solution (L', t') with $L' = L \setminus \{\ell\}$ and t' chosen so that $F_{L'}(t') = F_L(t_L)$. Let x' be the resulting allocation probabilities. By the same reasoning as before

$$x'_i = \begin{cases} \lambda_i \left[\frac{F_{L'}(t')}{\lambda_{L'}} + \frac{1-F_{L'}(t')}{\lambda_N} \right] = \lambda_i \left[\frac{F_L(t_L)}{\lambda_{L'}} + \frac{1-F_L(t_L)}{\lambda_N} \right], & i \in L' \\ \lambda_i \cdot \frac{1-F_{L'}(t')}{\lambda_N} \qquad\qquad = \lambda_i \cdot \frac{1-F_L(t_L)}{\lambda_N} & i \in \bar{L}'. \end{cases}.$$

We show this solution has objective function value at least as good as (L, t_L).

(1) For $i \in L'$, $x'_i > x_i$, since $\lambda_{L'} < \lambda_L$. All v_i are increasing in the allocation probabilities, so $v_i(X, x'_i) > v_i(X, x_i) \geq z$.
(2) For $i \in \bar{L}' \cap \bar{L}$, $x'_i = \lambda_i \cdot \frac{1-F_{L'}(t')}{\lambda_N} = \lambda_i \cdot \frac{1-F_L(t_L)}{\lambda_N} = x_i$ by construction. It follows that $v_i(X, x'_i) = v_i(X, x_i) \geq z$.
(3) For $i = \ell$, $x'_\ell < x_\ell$. However,
$$\frac{x'_\ell}{\lambda_\ell} = \frac{x'_k}{\lambda_k} = \frac{x^z_k}{\lambda_k} \geq \frac{x^z_\ell}{\lambda_\ell},$$
since ℓ, k are both in $\bar{L}'$, $x'_k = x_k = \hat{x}^z_k$, and $\ell > k$. As a result, $x'_\ell \geq \hat{x}^z_\ell$ and $v_\ell(X, x'_\ell) \geq z$.

We conclude that (L', t') with resulting allocation x' has objective function value at least z. Finally, we compare $|V_{(L,t_L)}|$ and $|V_{(L',t')}|$: Since ℓ was the largest indexed agent in L, removing ℓ from the priority list can not introduce new violating pairs, so $V_{(L,t_L)} \subseteq V_{(L',t')}$. We also know that $(\ell, k) \in V_{(L,t_L)}$ and $(\ell, k) \notin V_{(L',t')}$, so $V_{(L,t_L)} \subset V_{(L',t')}$ and $|V_{(L,t_L)}| > |V_{(L',t')}|$. This contradicts the fact that (L, t_L) was the optimal solution with the fewest violations. □

Theorem 10 applies to fairly general valuation functions, or even a situation where different recipients have (structurally) different valuation functions, the only requirement is that $v_i(X, x)$ depends only on X_i and x_i and is strictly increasing in x_i. Theorem 10 does not say anything about how long those on the priority list should receive priority. Fortunately, given a target allocation $\hat{x}^z$ and a priority list L, finding a time t so that $x_i^{BPL(L,t)} \geq \hat{x}^z$ can be done with a simple linear program.

Let $0 \leq F_L(t) \leq 1$ be a continuous variable representing the probability that the item is claimed before it is announced to all recipients. Consider the linear program

$$\begin{aligned} \max \quad & F_L(t) \\ s.t. \quad & \hat{x}^z_i \leq \lambda_i \left[\frac{F_L(t_L)}{\lambda_L} + \frac{1 - F_L(t_L)}{\lambda_N} \right], && \forall i \in L \\ & \hat{x}^z_i \leq \lambda_i \cdot \frac{1 - F_L(t_L)}{\lambda_N}, && \forall i \in \mathcal{N} \setminus L \\ & 0 \leq F_L(t) \leq 1. \end{aligned} \tag{Time-LP(L)}$$

If Time-LP(L) is feasible with $F_L(t) = \gamma$, then it means that objective function value z is achievable with (L, t), where $F_L(t) = 1 - \exp(-\lambda_L t) = \gamma$. Linear programs are solvable in polynomial time.

Putting it all together, we are able to find the optimal 2-stage priority list.

Theorem 11. *Suppose solving Time-LP(L) takes time $LP \in O(poly(n))$. We can find the optimal 2-stage priority list by performing binary search on the objective function value, where checking the feasibility of an objective function in an iteration of the binary search takes time $O(n \cdot LP)$.*

The initial lower bound can be set to $\min_{i \in \mathcal{N}} v_i(X)$ and the upper bound to $\max_{x \in \Delta_n} \min_{i \in \mathcal{N}} v_i(X, x)$. Every iteration of the binary search procedure requires finding $\hat{x}^z$ and solving Time-LP(L) for each of the n prefixes of the permutation which orders recipients in decreasing order of $\hat{x}^z / \lambda$.

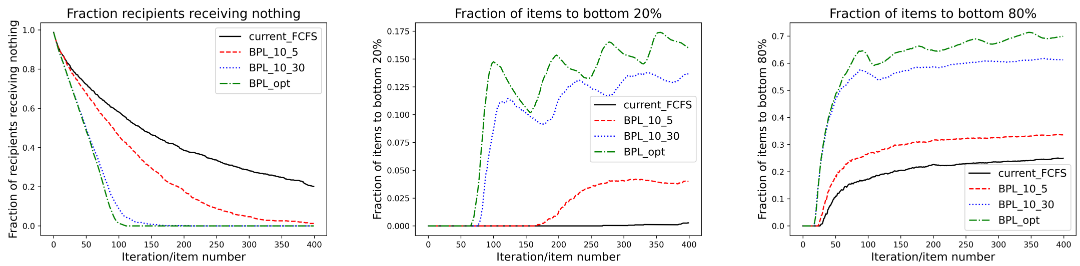

Fig. 1. Priority lists lead to more equitable allocations in terms of the fraction of recipients receiving no donations (left) and the fraction of donations that go to the worst-off 20% (center) and 80% (right) of recipients.

5 COMPUTATIONAL STUDY

We conduct simulations to compare the proposed priority lists to the existing FCFS claims system. We calibrate the model with a sample of data provided by FRP. The sample contains details on roughly 400 items, including a description and weight (in lbs), allocated to a total of 81 recipients in Florida.

We simulate 80 recipients, indexed with set N, and 400 items. The response time of agent i is distributed exponentially with rate $\hat{\lambda}_i = \hat{n}_i / \sum_{i \in N} \hat{n}_i$, where $\hat{n}_i$ is the total number of donations received by recipient i in the data.[4] Notice that $\sum_{i \in N} \hat{\lambda}_i = 1$, so the expected time until an item is claimed is 1 unit. The weight (in lbs) of each item j, w_j, is drawn uniformly with replacement from the empirical distribution of weights. Results are averaged over 10 runs.

We compare the current system and several binary priority lists.

(1) The current system (FCFS) which announces each item to all recipients and allocates to the first responder;
(2) Two simple priority lists, which give priority to the 10% recipients with lowest value for 5 and 30 units of time, respectively. Denoted BPL_10_5, BPL_10_30. Both the duration and number of recipients on the priority list remains constant for all iterations.
(3) The binary priority list which maximizes the minimum value, as described in Section 4, denoted BPL_opt. Both the number of agents on the priority list and its duration depends on the item and current allocation.

We report several aggregated metrics. First, the fraction of recipients receiving no items. Second, the fraction of items allocated to the 20% (80%) of recipients with lowest value (perfectly uniform distribution would give x% of items to the bottom x% of recipients). Finally, the time until each item is claimed.

In terms of equitability, we find that all the priority lists we test yield significant improvements over FCFS (Figure 1). The current FCFS system leads to roughly 20% of recipients receiving none of the first 400 donations and allocates only roughly 25% of the donations to the 80% worst-off recipients (or inversely, it allocates 75% of donations to only 20% of recipients). Both simple priority lists improve over this, leading to almost every recipient receiving at least one donation, 5-13% of donations going to the worst-off 20% and 30-60% going to the worst-off 80%. The optimal binary priority list from Section 4 performs even better, giving 15-17% of items to the bottom 20% of recipients and nearly 70% of donations to the bottom 80% (inversely, the top 20% of recipients get only 30% of items).

Figure 2 plots the time until donation. Unsurprisingly, delays in announcing the item to especially the those recipients with faster response rates increases the time before an item is claimed. This slowdown is fairly

[4]This makes several simplifying assumptions, including that all recipients were registered in the system for the entire period and equally interested in every donation. Reality is more complicated.

consistent for the two simple binary priority lists. Curiously, a small number of the items takes a very long time to be allocated under the optimal priority list. Closer inspection reveals that for those items the priority list consisted of only a very small fraction of the slowest recipients (1-3), and the system essentially waits however long it takes these slow recipients to respond. On average, BPL_opt allocated items roughly 20% faster than BPL_10_30, but care must be taken to avoid delaying allocation unnecessarily.

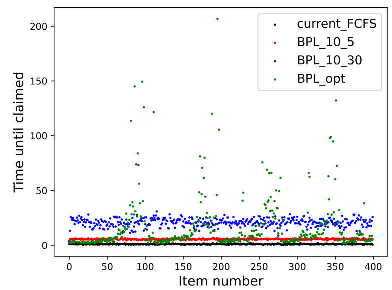

Fig. 2. Time until each item is allocated.

6 DISCUSSION AND CONCLUSION

We show theoretically that optimizing over priority lists is tractable, and via simulations that they lead to significantly more equitable outcomes than the existing FCFS allocation system. We leave open (for now) several interesting avenues of research. First, the computational results show that priority can lead to donations being allocated significantly slower, because there is a period when only the inactive or very slow recipients are aware of the item. When donations are perishable or pickup times from donors are heavily constrained this may be disastrous. Further investigation is required to come up with policies that respect item expiration dates. Second, the current model assumes every recipient is equally interested in every item which is unlikely to be the case. Richer preferences immediately lead to several questions, including how preferences are structured, whether they are Bayesian or adversarial, etc. There may also be other objectives, including weighted fairness notions, that make sense in this context.

Finally, data-sharing between food rescue platform's and recipients would enable the allocation of resources based on real-time demand, rather than proxies like average demand or organization size. This may be a bridge too far at this time, however, expanding the food rescue platform's logistical capabilities to include delivery and the splitting of large donations would likely go a long way towards more equitable outcomes.

REFERENCES

[1] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh. 2015. Online Fair Division: Analysing a Food Bank Problem. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 2540–2546.

[2] Siddhartha Banerjee, Vasilis Gkatzelis, Safwan Hossain, Billy Jin, Evi Micha, and Nisarg Shah. 2022. Proportionally Fair Online Allocation of Public Goods with Predictions. *arXiv preprint arXiv:2209.15305* (2022).

[3] Gerdus Benadè, Daniel Halpern, and Alexandros Psomas. 2022. Dynamic Fair Division with Partial Information. *Advances in Neural Information Processing Systems* 35 (2022), 3703–3715.

[4] Gerdus Benadè, Aleksandr Kazachkov, Alexandros Psomas, Ariel D Procaccia, and David Zeng. 2022. Fair and Efficient Online Allocations. Manuscript.

[5] J. G. Benadè, A. Kazachkov, C.-A. Psomas, and A. D. Procaccia. 2018. How to Make Envy Vanish Over Time. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*. 593–610.

[6] Yasushi Kawase and Hanna Sumita. 2022. Online max-min fair allocation. In *Algorithmic Game Theory: 15th International Symposium, SAGT 2022, Colchester, UK, September 12–15, 2022, Proceedings*. 526–543.

[7] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. 2019. WeBuildAI: Participatory Framework for Algorithmic Governance. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 181 (2019), 35 pages.

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

[9] Zheyuan Ryan Shi, Yiwen Yuan, Kimberly Lo, Leah Lizarondo, and Fei Fang. 2020. Improving efficiency of volunteer-based food rescue operations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13369–13375.

[10] Sean R Sinclair, Siddhartha Banerjee, and Christina Lee Yu. 2022. Sequential fair allocation: Achieving the optimal envy-efficiency tradeoff curve. *ACM SIGMETRICS Performance Evaluation Review* 50, 1 (2022), 95–96.

[11] David Zeng and Alexandros Psomas. 2020. Fairness-efficiency tradeoffs in dynamic fair division. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 911–912.