Allocating Scarce Resources using Scoring Rules: Modelling and Optimization

Sam Gilmour, Patrick Jaillet, and Nikos Trichakis

Operations Research Center, MIT, Cambridge, MA 02139, USA

{sgilmour,jaillet,ntrichakis}@mit.edu

Abstract. This paper considers the problem faced by an authority tasked with allocating scarce resources amongst a set of recipients. Allocations must be constructed according to a specific process: the authority selects a *scoring rule* to compute scores for each recipient-resource pair based on some observable properties that relate the pair, and an allocation mechanism then determines the allocation using only the scores. We develop a fluid approximation for this setting with heterogeneous recipient and resource types, and study the resulting optimization problems faced by an authority under two specific allocation mechanisms. The first mechanism allocates each resource type to the top-scoring recipient type; we show the corresponding optimization problem is NP-hard when the authority is restricted to linear scoring rules and present a mixed-integer program (MIP) formulation that can be solved to provable optimality within 5 hours for up to 30 recipient and resource types. We present two heuristics for the problem and establish bounds on their performance relative to optimality before showing their usefulness in practice. We also show how a scoring rule of high quality in the type-based model can fail in a setting where individual recipients and resources exhibit within-type variation in properties, and provide an approach that finds better scoring rules for allocating these individuals. Where possible, the steps in this analysis are repeated for a second, lottery-based allocation mechanism which allocates resources randomly in proportion with the scores of recipients.

1. Introduction

The question of how to allocate scarce resources amongst a set of recipients is relevant in countless settings around the globe. Cadaver organs must be allocated to patients, public school spaces to students, and public housing to residents who require it. The COVID-19 pandemic has almost universally forced hospitals to decide how to allocate limited supplies of medicine and ICU beds amongst their patients (Supady et al., 2021). Each of these settings share the fundamental characteristic that, according to the relevant laws which have been designed to make the system fair and equitable for recipients who participate, resources *cannot* be allocated on the basis of monetary transfers. Without a traditional market to decide which recipient receives each resource, it is left up to an authority to design and implement an allocation mechanism that operates to this end.

One type of mechanism that is frequently implemented in practice is a *priority mechanism*. In this type of mechanism, for each resource, the authority assigns each recipient a priority and then allocates the resources to recipients in the order defined by the associated priorities. Though priorities may be chosen arbitrarily, it is more common to compute them using a *scoring rule*: a function that acts upon observable properties relating the recipient-resource pair to produce a number (a score). When a system is implemented with a scoring rule that can be easily interpreted, this type of system has the attractive property that its allocations are easily understood by the participants.

Priority systems implemented with scoring rules also satisfy some fundamental principles of equity identified by Young (1994). Even so, there remains no universally agreed-upon definition of equity in the context of allocating scarce resources and therefore priority mechanisms are not always an appropriate choice for every setting. *Lottery* mechanisms are a different but still frequently-encountered approach to allocation that use randomization to appeal to an intuitive human notion of fairness. In this paper, we study both priority and lottery mechanisms that are based upon scoring rules.

The main question we will focus on is the following: given knowledge of the allocation mechanism, how can the authority select a scoring rule to obtain a desirable allocation? There have been surprisingly few studies that aim to answer this question. Even though scoring rule systems are widely implemented in practice and backed by sound economic principles, guidance for implementing these systems has therefore been sparse. This paper addresses the deficit by providing models and formulations that can be used by an authority as they attempt to design an effective scoring-rule-based allocation mechanism.

1.1. Contributions

In Section 2, we introduce the setting for allocating scarce resources with a scoring rule and develop a fluid approximation model based on heterogeneous recipient and resource types. We use the model to define the optimization problem faced by the authority given an arbitrary allocation mechanism.

In Section 3 we introduce the priority allocation mechanism, instantiate the optimization problem faced by the authority, and prove it to be NP-hard. We present a MIP formulation with a simple structure and show that the formulation scales to a size that is useful for practical modelling purposes. Section 4 provides two heuristics for solving this optimization problem approximately as well as bounds on their performance relative to optimality. Section 5 explains how scoring rules of high quality in the type-based model can fail when individual recipients and resources exhibit within-type variation in their properties, and provides an approach to deal with this situation.

Section 6 repeats most of these steps for the lottery allocation mechanism: we define the optimization problem faced by the authority, formulate it as a nonconvex quadratic problem, and present and test a heuristic and approach for allocating individuals. Section 7 concludes with a brief discussion on fairness in the context of the two allocation mechanisms presented in the paper.

1.2. Related Literature

As previously mentioned, little effort has been dedicated to the problem of selecting a scoring rule that maximizes welfare under a given allocation mechanism. Of the few studies that do, all focus on the priority allocation mechanism where a resource is allocated to the recipient with the highest score. They are generally motivated by applications in organ allocation systems.

Bertsimas, Farias, and Trichakis (2013) study the cadaver kidney allocation system with a data-driven optimization approach. They develop a heuristic for finding a scoring rule that produces an efficient allocation while approximately satisfying ex-post fairness constraints under the priority allocation mechanism. Their procedure first solves a maximum-weighted bipartite matching problem with fairness constraints and uses the optimal dual variables to compute *fairness-adjusted* reward coefficients for each patient-organ pair. They then solve a regression problem to describe these coefficients using a scoring rule. Though they also optimize for the scoring rule, their heuristic does not attempt to model the mechanics of the allocation mechanism as we do in this paper.

Ding, McCormick, and Nagarajan (2021) use tools from queuing theory to develop their type-based model of resources and recipients where the scoring rule is the sum of two terms: a function of the types of a recipient-resource pair, and an increasing function of the waiting time within a recipient queue. All resources of a particular type are allocated to the recipient queue with the highest score. They derive an approximation to the steady-state behaviour of the system and find a scoring rule that maximizes a specific objective function that trades off efficiency with fairness.

With a discrete stochastic model, David and Yechiali (1995) build on Righter (1989) to consider the problem of allocating a finite set of organs amongst a finite set of patients. In their setting the reward obtained by matching an organ to a patient depends on only a single attribute. While their characterization of the optimal policy allows it to be interpreted as a scoring rule, our model is concerned with types that are identified by multiple attributes rather than just one.

It is also worth highlighting some related work which either entirely sweeps away the role of the scoring rule or minimizes its importance. Amongst this group is a growing body of recent work that studies how techniques from optimization can be applied to allocation systems. Shi (2019) recognizes that priorities are often optimization variables for the authority to choose. He proposes and studies a model that allocates a continuum of customers (split into market segments) to resources using a priority mechanism, while optimizing over the priorities. The model is general enough to encompass allocation mechanisms such as the Gale-Shapley deferred acceptance, top trading cycles, and serial dictatorship algorithms. Su and Zenios (2006) provide a model for allocating cadaver kidneys that allows patients to choose a queue at the time they enrol on the waitlist based on the type of kidney they would accept. The optimization variables are the proportions of each kidney type to allocate to each queue. Ashlagi and Shi (2016) both Bodoh-Creed (2020) consider the allocation of heterogeneous resources to customers who have both a publicly known type and privately known vector of utilities. They optimize the allocation as a function for each customer type that maps a utility vector onto a set of allocation proportions over the resources.

In the realm of queueing theory, Su and Zenios (2004) note that in the early 2000s an explosion of patients enrolling on organ transplant waitlists combined with a continued shortage of organs to leave the priority-based system resembling a first come, first served (FCFS) queue, and then study the effect of patient choice in a FCFS setting. Afeche, Caldentey, and Gupta (2019) consider a queuing system with a set of heterogeneous customer classes processed by a set of heterogeneous servers and seek to find the matching topologies (which are not selected based on scores) in this bipartite graph which define the Pareto frontier for system reward and customer delay. Sisselman and Whitt (2007) and Mehrotra, Ross, Ryder, and Zhou (2012) both model allocation of calls in a call center and aim to maximize match-specific rewards.

While there is a lack of technical research on the role of scoring rules in allocation systems, real-world implementations have been studied extensively from the perspective of their operations: the methods used to choose scoring rules, debates on the ethics of the priority allocation mechanism, and the outcomes observed when systems are simulated or implemented in practice. Sparrow (1951) reviews a historical example from the 1940s with the demobilization of United States (US) army troops after the Second World War, where individual soldiers posted overseas were removed from duty according to points assigned by a scoring rule that took into account factors such as the number of dependents of the soldier and their time spent in combat. Been, O'Regan, Waldinger, and Center (2018) and Thakral (2016) study the public housing allocation system in the US (for which some regions use a points-based system) and Greely (1977) notes that jobs in the US civil service have historically been assigned on the basis of a scoring rule. Edwards (1999) describes waitlist systems that operate within the National Health Service (NHS) in the United Kingdom (UK) and the scoring systems that allocate medical services to patients on these lists. Zenios, Wein, and Chertow (1999) use a simulation model and data taken from the US to compare various mechanisms for cadaver kidney allocation, amongst which they include the scoring system used at that time.

2. Allocating Resources with a Scoring Rule

This section describes the resource allocation setting considered in this paper. We briefly describe the role of scoring rules in many real-world allocation systems and then develop a fluid model that leads to tractable (though NP-hard) optimization problems. This section does *not* introduce any specific procedures for allocating resources. It is left to Section 3 and Section 6 for describing the two allocation mechanisms we study: the *priority* and *lottery* mechanisms, respectively.

Our motivation is a setting where an authority is tasked with allocating resources amongst recipients. In much of the literature on allocation systems, the authority is allowed complete freedom in choosing which recipient to allocate each resource. Our setting is different: when a resource arrives, the authority computes a score for each recipient using a function called a *scoring rule* which acts on a vector of properties that relates the recipient-resource pair. An *allocation mechanism* then allocates the resource to a recipient only on the basis of these scores. This approach is used frequently in practice – for instance, the kidney allocation system in the US uses a scoring rule that assigns points to patients for properties such as the number of years they have spent on the waitlist, and then allocates the next kidney to the patient with the highest score (Israni et al., 2014).

Throughout the paper, we specifically consider the setting where resources are scarce. This statement will be made precise when the fluid model is introduced in Section 2.1, but in essence it means that new recipients arrive at a much larger rate than the resources. Many real allocation systems have this characteristic: for instance, in 2020, patients waiting to receive a kidney transplant in the US outnumbered available kidneys by 5 to 1 (UNOS, 2021). In 2021, low-income families outnumbered available affordable housing units in some urban centers within the US by up to 6 to 1 (NLIHC, 2021).

When resources are scarce, it is important that they are allocated in line with certain axiomatic principles of equity. Young (1994) notes that scoring rule systems provide *impartiality*: no distinctions are made between recipients except for differences in their properties. Two recipients who have the same properties will necessarily have the same scores and subsequently the same chance of receiving any resource. Young (1994) identify two further principles that are satisfied specifically by the priority mechanism; we will return to these in Section 3, after describing our fluid model for allocation.

2.1. Fluid Model

Our model is based on I heterogeneous recipient types and J heterogeneous resource types. Figure 1 illustrates the setup, which operates as follows. Recipients of type i arrive into the system at a rate of $\lambda_i \in \mathbb{R}_+$ and enter a queue. Resources of type $j \in [J]$ arrive at a rate of $\mu_j \in \mathbb{R}_+$ and are fractionally allocated across the recipient queues. Let $x_{ij} \in [0,1]$ be the fraction of resource type j arrivals that are allocated to the recipient queue of type i, meaning that resources are allocated to queue i at a rate $\sum_{j=1}^{J} \mu_j x_{ij}$. Assume that $\bar{\lambda}_i := \lambda_i - \sum_{j=1}^{J} \mu_j x_{ij} \geq 0$ (we will return to the significance of this assumption soon).

The allocation fractions in $\mathbf{x}$ must belong to a set, $\mathcal{X}$, so that no more than the total resources arriving are allocated amongst the recipient queues:

$$\mathcal{X} := \left\{ \mathbf{x} \in [0,1]^{I \times J} : \sum_{i=1}^{I} x_{ij} \leq 1, \ \forall j \in [J] \right\} \tag{1}$$

Let $L_i(t)$ be the length of queue i at some time t and $\delta > 0$ be the next small unit of time. Within this unit of time, $\delta\lambda_i$ individuals join the queue and $\delta \sum_{j=1}^{J} \mu_j x_{ij}$ individuals leave the queue after receiving a resource. Each remaining individual in the queue decides to leave on their own with probability δq_i, which depletes the queue by $\delta q_i \left(L(t) + \delta\bar{\lambda}_i \right)$. The parameter q_i models recipient attrition seen in systems such as for organ allocation (where patients can die or become too sick to receive a transplant).

These dynamics allow us to derive an expression for the rate of change of the length of the queue with time:

$$L_i(t+\delta) - L_i(t) = \delta\bar{\lambda}_i - \delta q_i \left(L_i(t) + \delta\bar{\lambda}_i \right) \tag{2}$$

$$\lim_{\delta \to 0} \left[\frac{L_i(t+\delta) - L_i(t)}{\delta} \right] = \bar{\lambda}_i - q_i L_i(t) \tag{3}$$

Solving the resulting differential equation for $L_i(t)$ and taking the limit as $t \to \infty$ leaves an expression for the steady-state length of the queue:

$$L_i = \frac{\bar{\lambda}_i}{q_i} \tag{4}$$

Finally, let $R_{ij} \in \mathbb{R}_+$ be the reward obtained by allocating a resource of type j to a recipient of type i, and define $r_{ij} := \mu_j R_{ij}$. The rate of reward earned by the authority is:

$$\sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij} x_{ij} \tag{5}$$

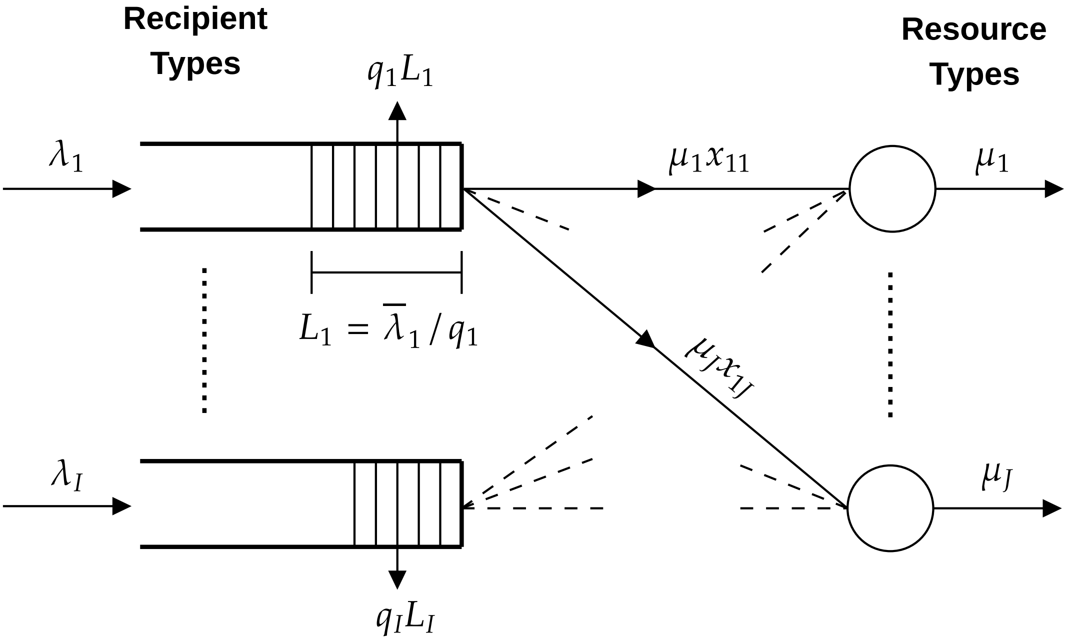

Figure 1: The setting for the fluid approximation model, where recipients arrive and form queues of types before being served by the arriving resources (split across the queues).

2.2. Optimizing Allocation Fractions

So far, we have described a model where fixed allocation fractions give rise to queues with lengths and an overall rate of earning reward. But our objective is to model a system that operates with a scoring rule – therefore, allocation fractions should be computed based on the scores of each recipient-resource pair, and scores should be computed based on properties that relate these pairs.

With $K \in \mathbb{N}$ being this number of properties, let $\mathbf{f}_{ij} \in \mathbb{R}^K$ be the values relating recipient type i with resource type j. These vectors act as the inputs for the scoring rule $s : \mathbb{R}^K \to \mathbb{R}$. As a shorthand we use $s_{ij} = s(\mathbf{f}_{ij})$ to refer to the score of recipient type i for resource type j. An *allocation function* is simply a function that maps scores onto allocation fractions:

Definition 1 (Allocation Function)**.** *An allocation function, defined on a set of scoring matrices $\mathcal{S} \subseteq \mathbb{R}^{I \times J}$, has the signature $\mathbf{x} : \mathcal{S} \mapsto \mathcal{X}$. It maps a matrix of scores onto a matrix of allocation fractions, and must be scale-free by satisfying the property:*

$$\mathbf{x}(\theta \mathbf{s}) = \mathbf{x}(\mathbf{s}) \qquad \forall \theta > 0,\ \forall \mathbf{s} \in \mathcal{S}$$

The two functions we define and study specifically are the *priority* and *lottery* allocation functions, introduced in Section 3 and Section 6 respectively.

An allocation function may be defined on all score matrices in $\mathbb{R}^{I \times J}$, but the authority is the one who actually generates the scores by choosing a scoring rule from within a restricted class to act on the properties. This decision will be described in more detail in Section 2.3, but for now let $\bar{\mathcal{S}} \subseteq \mathcal{S}$ be the set of scores that can be produced by the authority.

We return to the assumption that $\bar{\lambda}_i \geq 0$ for each queue i. When resources are scarce it can be expected that the recipient queues in our model do not deplete regardless of the scores chosen by the authority. This observation motivates the precise definition of scarcity used in the rest of this paper:

Assumption 1 (Scarce Resources)**.** *For some allocation function,* $\mathbf{x}$, *resources are assumed to be scarce in the sense that, for every recipient type i, the following holds:*

$$\lambda_i - \sum_{j=1}^{J} \mu_j x_{ij}(\mathbf{s}) \geq 0 \qquad \forall \mathbf{s} \in \bar{\mathcal{S}}$$

With all these definitions in place, we now define the central optimization problem faced by an authority who aims to maximize the rate of reward earned under a particular allocation function, $\mathbf{x}$, and set of possible scores, $\bar{\mathcal{S}}$:

Problem 1 (Fluid Model Optimization)**.**

$$\max_{\mathbf{s} \in \bar{\mathcal{S}}} \qquad \sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij} x_{ij}(\mathbf{s}) \tag{6}$$

2.3. Scoring Rule Considerations

Recall that $\bar{\mathcal{S}}$ is produced by giving the authority a set of candidate scoring rules to choose from. We consider linear rules as the broad set of candidates, but since the priority and lottery allocation functions are defined on different sets of scores, some extra restrictions must be placed on these rules depending on the allocation function. These will be introduced in the relevant sections.

Above all else, a linear scoring rule is *interpretable*. Choosing a linear rule simply means specifying a weight vector $\mathbf{w} \in \mathbb{R}^K$ so that $s(\mathbf{f}_{ij}) = \mathbf{w}^\intercal \mathbf{f}_{ij}$ with the interpretation that

w_k is the marginal score assigned for a unit increase in the value of the kth property in $\mathbf{f}_{ij}$. This information is easily digested by the recipients in an allocation system and it contributes to explaining why linear scoring rules have been so widely adopted in real-world systems. Regardless, if we wish to assign scores based on a nonlinear function of the properties then an extra component that computes this nonlinear function can simply be appended to $\mathbf{f}_{ij}$.

3. Priority Allocation

This section introduces the *priority* allocation function, which is designed to model the allocation procedure in real-world allocation systems where each resource is allocated to the top-scoring recipient. We will define the allocation function, analyze the optimization problem faced by the authority, and present some numerical results.

First, though, it is worth noting two observations made by Young (1994). As well as the principle of impartiality that has already been described, allocating resources using a priority mechanism satisfies the principle of *consistency*. This means that two recipient types i and i' will always be allocated a particular resource type in the same way regardless of the other recipients present in the system – which is certainly a desirable property for an allocation system to have.

Young (1994) also notes that when linear scoring rules are used in a priority mechanism, the resulting system is *separable*: scores do not exhibit complementary effects between properties. To be more precise – fix a resource type j and suppose we have two arbitrary pairs of recipient types: $a, b \in [I]$ and $c, d \in [I]$. Each pair is identical in properties 3 through K (though possibly different between the pairs). Now let a match c in the first two properties ($f_{aj1} = f_{cj1}$ and $f_{aj2} = f_{cj2}$) and the same for b and d. A scoring mechanism is separable if $s_{aj} \geq s_{bj} \iff s_{cj} \geq s_{dj}$. Though some allocation settings may indeed be better suited to inseparable mechanisms, it is important to note that such complementary effects can still be modelled by amending $\mathbf{f}_{ab}$ with extra properties that are computed based on some function of the original properties.

The priority allocation function, retaining notation as $\mathbf{x}$, can be applied to *any* matrix of scores. Accordingly, we set $\mathcal{S} = \mathbb{R}^{I \times J}$. Letting $\mathcal{I}_j = \arg\max_{i \in [I]}(s_{ij})$ be the top-ranked recipient types for resource type j, the priority allocation function is:

$$x_{ij}(\mathbf{s}) = \begin{cases} 1 & i \in \arg\max_{k \in \mathcal{I}_j}(r_{kj}) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$x_{ij}(\mathbf{s})$ is equal to 1 whenever recipient type i is top-scoring for resource type j and also wins a tiebreaker by having the highest reward of all top-scoring queues. Note that this function is not well-defined when there are two top-scoring queues that are tied in the reward tiebreaker. We leave it up to implementation on how to deal with these secondary ties given that any choice of policy does not impact the reward earned. In the absence of secondary ties, $\mathbf{x}$ certainly satisfies the required signature $\mathbf{x} : \mathcal{S} \mapsto \mathcal{X}$ whilst being scale-free, and is therefore a valid allocation function according to Definition 1.

In the fluid model this allocation function effectively assigns all resources to their top-ranked recipient queue. It is therefore useful for modelling systems such as the previously-described organ allocation systems under the assumption of resource scarcity.

Finally, a restriction needs to be placed on the candidate scoring rules from which the authority can choose. Since setting $\mathbf{w} = \mathbf{0}$ results in a zero-score matrix and $\mathbf{x}(\mathbf{0})$ simply picks the recipient type with the highest reward for each resource type, we ignore the trivial $\mathbf{w} = \mathbf{0}$ rule. The set of scores available to the authority is therefore:

$$\bar{\mathcal{S}} = \left[\mathbf{w}^\intercal \mathbf{f}_{ij} : \mathbf{w} \neq \mathbf{0}\right]$$

It is possible that some problem instances admit other rules that produce the zero-score matrix (or some other constant matrix) – we comment on how to deal with these in Section 3.2, but eliminating $\mathbf{w} = \mathbf{0}$ means this situation does not occur in every instance and therefore allows us to establish some complexity results.

3.1. Complexity of Optimization

This definition of $\mathbf{x}$ and $\bar{\mathcal{S}}$ combined with Problem 1 gives rise to a well-defined optimization problem. This subsection characterizes the complexity of the problem both intuitively (with an example) and formally (with a result showing it is NP-hard).

First, though, some new notation is required. The priority allocation function ensures that each $x_{ij}(\mathbf{s}) \in \{0, 1\}$ and it is therefore convenient to introduce $\mathbf{y} = (i_1, \ldots, i_J)$ when referring to an allocation where $x_{i_j j} = 1$ for each resource type j. This notation allows a *feasible allocation* to be easily defined:

Definition 2. *An allocation $\mathbf{y} = (i_1, \ldots, i_J)$ is feasible under the priority allocation function if:*

$$\exists \mathbf{s} \in \bar{\mathcal{S}} : x_{i_j j}(\mathbf{s}) = 1,\ \forall j \in [J]$$

Much of the difficulty in solving the optimization problem is due to the fact that only a subset of all allocations are feasible for any given instance of the problem. In fact, Proposition 1 provides a basic characterization of the feasible allocations in terms of the polyhedral structure of the property vectors. A proof is included in Appendix A.

Proposition 1 (Characterization of Feasibility)**.** *Let the polytope $\mathcal{P}_j = conv(\{\mathbf{f}_{1j}, \ldots, \mathbf{f}_{Ij}\})$ be defined for each resource type $j \in [J]$. Let also $\mathbf{y} = (i_1, \ldots, i_J)$ be some allocation. $\mathbf{y}$ is a feasible allocation if and only if $\sum_{j=1}^{J} \mathbf{f}_{i_j j}$ is an extreme point of $\sum_{j=1}^{J} \mathcal{P}_j$.*

Unfortunately, this restricted set of feasible allocations can contain local optima in the sense that a feasible allocation may have an objective value strictly greater than all its neighbours in $\sum_{j=1}^{J} \mathcal{P}_j$ without being the global optimum. Example 1 gives an illustration of such local optima – which, at an intuitive level, are the main sources of complexity in the problem. When combined with the observation that the number of extreme points in the sum of J polytopes may increase exponentially in J (Delos & Teissandier, 2014), we are left with a complex search space of exponential size that contains local optima.

Proposition 2 establishes that the problem is, in fact, NP-hard. A proof using a reduction from the Maximum Feasible Linear System (MAX-FLS) problem studied in Amaldi and Kann (1995) is included in Appendix B.

Proposition 2 (NP-Hardness of Optimization)**.** *Problem 2 is NP-hard when its input size is measured in the number of resource types J.*

Amaldi and Kann (1995) provide a simple algorithm that approximates MAX-FLS within a constant factor of 2 and also show that the problem *cannot* be approximated arbitrarily well. Unfortunately, these results do not easily extend to our problem. In Section 4.1 we describe two heuristics which provide good practical performance and establish some elementary performance bounds – but results with similar strength as those available for the MAX-FLS problem remain elusive.

3.2. Optimizing with a MIP

Even though the problem is NP-hard, this subsection shows that it admits formulation as a linear MIP. It is useful to first introduce a group of sets that divide the set of linear scoring rules the authority may choose from:

$$\mathcal{W}_{ij} = \left\{ \mathbf{w} \in \mathbb{R}^K : \mathbf{w}^\intercal (\mathbf{f}_{ij} - \mathbf{f}_{i'j}) \geq 0,\ \forall i' \neq i \right\} \setminus \{\mathbf{0}\} \tag{8}$$

$\mathcal{W}_{ij}$ is the set of scoring rules for which recipient type i is top-scoring for resource type j. Our formulation in Problem 2 makes use of the binary variables $x_{ij} \in \{0, 1\}$ for $i \in [I]$

Example 1. Consider a case where $I = 6$, $J = 2$, $K = 2$. The problem data takes the structure shown in Figure 2. It is immediately clear that, for example, the allocation $\mathbf{y} = (3, 5)$ is infeasible: if recipient type 3 is selected for the first resource type, recipient type 5 cannot be selected for the second resource type. This allocation does not correspond to an extreme point of $\mathcal{P}_1 + \mathcal{P}_2$.

A greedy algorithm that begins by choosing either recipient type 3 for resource type 1 or recipient type 5 for resource type 2 will find itself stuck in a local optimum within $\mathcal{P}_1 + \mathcal{P}_2$, because these choices prevent any reward from being obtained for the other resource type. The optimal decision is to pick $\mathbf{w} = (-1, 0)$ which leads to an allocation $\mathbf{y} = (1, 1)$ and reward of $2 + 2\delta$.

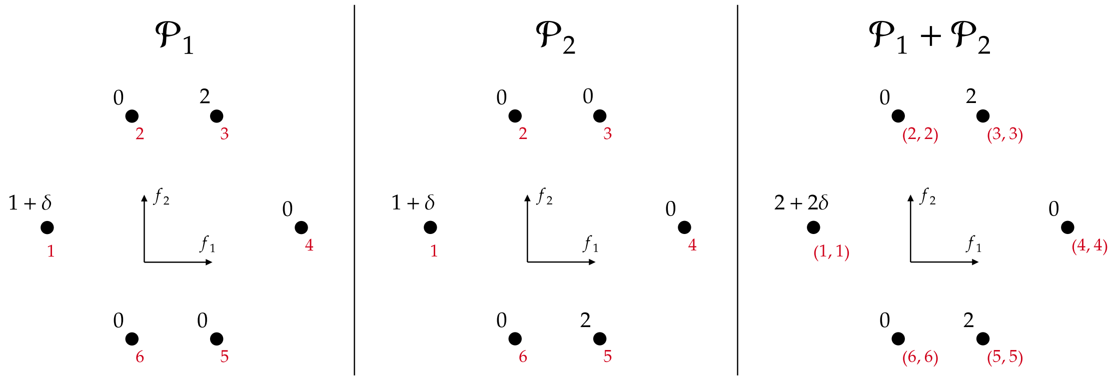

Figure 2: Example demonstrating a local optimum in the optimization problem with the priority allocation function where $I = 6$, $J = 2$, $K = 2$. The left-hand and center groups of points correspond to properties for the first and second resource types respectively, and the right-hand group of points is the set of extreme points of the sum $\mathcal{P}_1 + \mathcal{P}_2$. Red labels are the recipient type indices and black labels are the reward coefficients for the pair. $\delta > 0$ is a small positive constant.

and $j \in [J]$ and constrains them so that $x_{ij} = 1$ if and only if the chosen linear scoring rule $\mathbf{w}$ is contained in $\mathcal{W}_{ij}$.

Problem 2 (Priority Allocation Optimization)**.**

$$\max_{\mathbf{w}\in\mathbb{R}^K,\, \mathbf{x}\in\{0,1\}^{I\times J}} \quad \sum_{i=1}^{I}\sum_{j=1}^{J} r_{ij}x_{ij} \tag{9a}$$

$$\text{subject to} \quad \mathbf{e}^\intercal \mathbf{w} = 1 \tag{9b}$$

$$s_{ij} = \mathbf{w}^\intercal \mathbf{f}_{ij} \qquad \forall i \in [I],\ j \in [J] \tag{9c}$$

$$s_{ij} - s_{i'j} \geq M(x_{ij} - 1) \qquad \forall i \in [I],\ j \in [J],\ i' \neq i \tag{9d}$$

$$\sum_{i=1}^{I} x_{ij} \leq 1 \qquad \forall j \in [J] \tag{9e}$$

To confirm that this formulation is correct, suppose we have computed scores using a scoring rule $\mathbf{w} \in \mathcal{W}_{ij}$. It follows that $s_{ij} - s_{i'j} \geq 0$ for all $i' \neq i$, allowing the constraint in Equation (9d) to be satisfied when setting $x_{ij} = 1$. On the other hand, if $\mathbf{w} \notin \mathcal{W}_{ij}$ then there is some i' for which $s_{ij} - s_{i'j} < 0$ and setting $x_{ij} = 0$ is required to ensure the constraint holds. Note also that Equation (9e) ensures at most one recipient type is matched with each resource type in case there is a tie for the top-ranked queue, and the objective ensures that this tie is broken by selecting the match with maximum reward.

It is impossible to exactly model the requirement that $\mathbf{w} \neq \mathbf{0}$ in a MIP, but we include Equation (9b) to make this solution infeasible. Since the allocation function is scale-free the search space effectively covers all scoring rules with $\mathbf{e}^\intercal\mathbf{w} > 0$ but eliminates those with $\mathbf{e}^\intercal\mathbf{w} \leq 0$. The MIP can be solved a second time with $\mathbf{e}^\intercal\mathbf{w} = -1$ to cover $\mathbf{e}^\intercal\mathbf{w} < 0$, and covering $\mathbf{e}^\intercal\mathbf{w} = 0$ can be achieved by solving with a small perturbation of the properties.

An instance of this problem can clearly have multiple optimal solutions. Suppose that $\mathbf{w}^*$ is an optimal scoring rule to Problem 2 which produces an allocation $\mathbf{y}^* = (i_1^*, \ldots, i_J^*)$. If $\mathbf{y}^*$ is the unique optimal allocation, then the set of optimal scoring rules is given exactly by Equation (10):

$$\mathcal{W}^* = \left(\bigcap_{j=1}^{J} \mathcal{W}_{i_j^* j} \right) \tag{10}$$

and if $\mathbf{y}^*$ is not the unique optimal allocation, then $\mathcal{W}^*$ is a subset of the optimal rules.

Finally, a remark on the structure of the properties. Let **F** refer to the matrix formed by stacking values of corresponding properties across all recipient and resource types:

$$\mathbf{F} := \begin{bmatrix} f_{111} & \cdots & f_{11K} \\ \vdots & \ddots & \vdots \\ f_{I11} & \cdots & f_{I1K} \\ f_{121} & \cdots & f_{12K} \\ \vdots & \ddots & \vdots \\ f_{IJ1} & \cdots & f_{IJK} \end{bmatrix} \tag{11}$$

so that $\mathbf{Fw} \in \mathbb{R}^{IJ}$ gives the scores for each recipient-resource type pair in vector form. If the columns of **F** are linearly dependent, then there exists some $\mathbf{w} \neq \mathbf{0}$ so that $\mathbf{Fw} = \mathbf{0}$. It is also possible that there is some **w** which results in all scores being identical and nonzero. Both cases are unlikely to happen in practice if it is assumed that $K << IJ$. But if Problem 2 is solved and all scores are returned identical, a small $\epsilon > 0$ can be added to Equation (9d) to require that the top-scoring recipient type has strictly the highest score.

3.3. Computational Experiments

The scalability of the formulation was tested on randomly generated data. We used $I = J$ (the same number of recipient and resource types) with $K \in \{5, 10\}$. All elements in **F** and **r** were generated independently and uniformly at random on the unit interval $[0, 1]$. For each configuration of parameters, we generated 5 independent problem instances and solved the formulation in Problem 2 with a time limit of 5 hours. All tests were conducted on an Intel Xeon 2.1 GHz quad-core CPU with 32 GB of RAM and solved with Gurobi 9.1.1.

Figure 3 shows how the range of solution times changed as the number of recipient and resource types in the instance increase. Problems with up to 35 types for $K = 5$ and up to 25 types for $K = 10$ could be solved to provable optimality within the time limit.

Figure 4 shows how the range of optimality gaps changed as the number of recipient and resource types varied. Optimality gaps grew quickly once the problem could not be solved within 5 hours. When $K = 10$ and the number of types grew to 50, optimality gaps were between 35% and 50%.

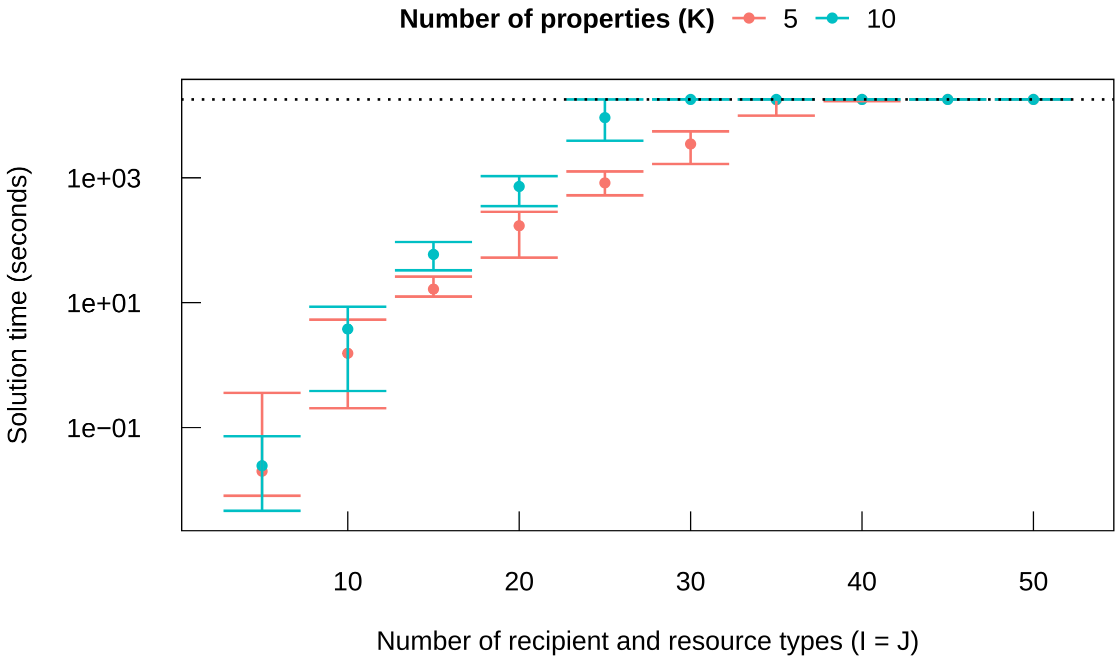

Figure 3: Ranges of solution times when instances were solved to provable optimality. The dashed line indicates the 5 hour time limit imposed on the solver.

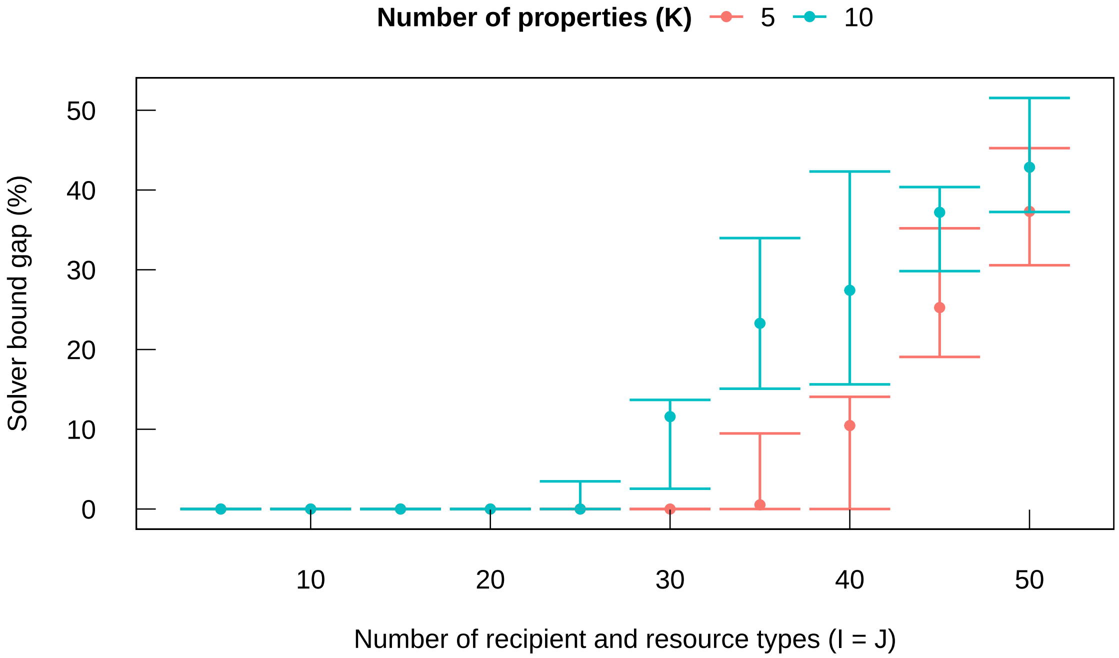

Figure 4: Ranges of the solver optimality gap after the 5 hour time limit had elapsed.

4. Heuristics for Priority Allocation

Though Problem 2 can be solved to provable optimality for moderately-sized instances with a reasonable number of properties, it is still useful to have some heuristics that provide good solutions when larger instances must be solved. This section describes two, and provides some performance bounds and numerical experiments that highlight when each should be applied.

4.1. Projection Heuristic

The first heuristic finds a scoring rule that produces priority scores which are *close* to the rewards by projecting $\mathbf{r}$ onto $\bar{\mathcal{S}}$. It is similar to the approach used by Bertsimas et al. (2013), who solve a least-squares problem to project their fairness-adjusted rewards onto the set of feasible scores that can be produced by the authority in their setup.

In our setting, the motivation for this heuristic is the following: suppose there is a scoring rule $\hat{\mathbf{w}} \in \mathbb{R}^K$ for which $\mathbf{f}_{ij}^\intercal \hat{\mathbf{w}} = r_{ij}$ for all $i \in [I]$ and $j \in [J]$. Then, since it guarantees the top-ranking recipient type has the highest reward for each resource type, this scoring rule is optimal. If we instead find scores that are close to the rewards, they may be good (rather than optimal).

In this section it is more convenient to represent the rewards and properties by stacking their components. Recall the previous notation used for the properties, $\mathbf{F}$, and write the rewards as:

$$\mathbf{r} := (r_{11}, \dots, r_{I1}, r_{12}, \dots, r_{IJ}) \tag{12}$$

The *projection solution* for our setup is defined when $\mathbf{F}$ has linearly independent columns (recall, a mild assumption) and is given in Definition 3:

Definition 3 (Projection Solution). *The projection solution $\hat{\mathbf{w}}(\mathbf{F}, \mathbf{r}) \in \mathbb{R}^K \setminus \{\mathbf{0}\}$ is obtained by solving:*

$$\hat{\mathbf{w}}(\mathbf{F}, \mathbf{r}) := \arg\min_{\mathbf{w} \in \mathbb{R}^K} \frac{1}{2} ||\mathbf{F}\mathbf{w} - \mathbf{r}||_2^2 = (\mathbf{F}^\intercal \mathbf{F})^{-1} \mathbf{F}^\intercal \mathbf{r}$$

Note that the projection solution is not defined when $\mathbf{F}^\intercal \mathbf{r} = \mathbf{0}$ or equivalently $\mathbf{r} \perp \text{colspace}(\mathbf{F})$.

4.1.1. Analysis

In general the projection solution can be arbitrarily far from optimality. Figure 5 illustrates a problem instance where this is the case for $I = 2, J = 1, K = 1$.

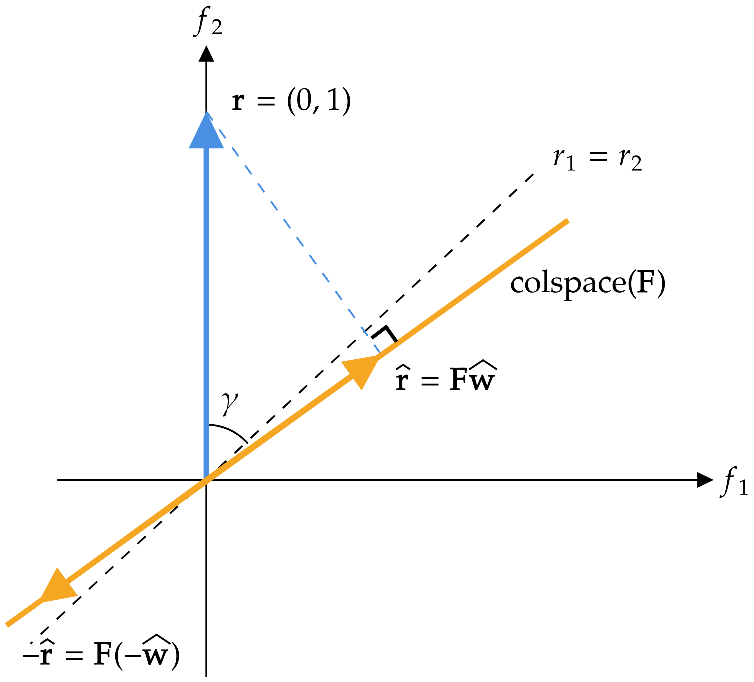
f_2
$\mathbf{r} = (0, 1)$
$r_1 = r_2$
colspace($\mathbf{F}$)
$\hat{\mathbf{r}} = \mathbf{F}\widehat{\mathbf{w}}$
γ
f_1
$-\hat{\mathbf{r}} = \mathbf{F}(-\widehat{\mathbf{w}})$

Figure 5: In this example $\hat{r}_1 > \hat{r}_2$, so the projection solution generates reward $r_1 = 0$. However, there exists a scoring rule $-\hat{\mathbf{w}}$ with $-\hat{r}_1 < -\hat{r}_2$ and so it is possible for the authority to earn $r_2 = 1$.

In Figure 5, the angle γ between $\mathbf{r}$ and $\mathcal{S}$ is large and contributes to how poorly the projection solution performs. γ is a measure of how well the properties *describe* the rewards. If $\gamma = 0$ then the properties can perfectly reproduce the rewards, whereas if $\gamma = \frac{\pi}{2}$ then the properties provide no information about the rewards. We bound the performance of the projection heuristic in terms of γ, which is defined for the angle between a vector $\mathbf{r}$ and score set as:

$$\gamma = \cos^{-1}\left(\max_{\mathbf{s}\in\mathcal{S}} \frac{\mathbf{r}^\intercal \mathbf{s}}{||\mathbf{r}|| \cdot ||\mathbf{s}||}\right) \tag{13}$$

We consider a single resource type ($J = 1$) and drop the corresponding subscript. The bound in Proposition 3 (with a proof in Appendix C) extends easily to the case $J > 1$ when the result is interpreted as a *per-resource-type* bound.

Proposition 3. *Let $J = 1$, and the projected reward coefficients be given by $\hat{\mathbf{r}} \in \mathbb{R}^I$ with $\hat{r}_1 \geq$*

$\hat{r}_2 \geq \ldots \geq \hat{r}_I$. Let γ be the angle between $\mathbf{r}$ and $\mathcal{S}$ as in Equation (13).

Let z^* be the optimal objective value of Problem 2, and let z be the objective value of the projection solution. The following bound holds for any angle $0 \leq \gamma < \pi/2$:

$$\frac{z^* - z}{||\mathbf{r}||} \leq \max\left(0, \frac{\hat{r}_2 - \hat{r}_1}{||\hat{\mathbf{r}}||} + \sqrt{2}\sin(\gamma)\right) \tag{14}$$

A second proposition writes the RHS in terms of $\mathbf{r}$ only:

Proposition 4. *Let $J = 1$, and the projected reward coefficients be given by $\hat{\mathbf{r}} \in \mathbb{R}^I$ with $\hat{r}_1 \geq \hat{r}_2 \geq \ldots \geq \hat{r}_I$. Let γ be the angle between $\mathbf{r}$ and $\mathcal{S}$ as in Equation (13).*

Let z^* be the optimal objective value of Problem 2, and let z be the objective value of the projection solution. The following bound holds for any angle $0 \leq \gamma < \pi/2$:

$$\frac{z^* - z}{||\mathbf{r}||} \leq \max\left(0, \frac{r_2 - r_1}{||\mathbf{r}||} + 2\sqrt{2}\sin(\gamma)\right) \tag{15}$$

The left-hand side of the bound does not have the most intuitive interpretation. Note that, ideally, we would like to be able to produce a bound on $(z^* - z)/z^*$ that measures the relative optimality gap. But in the absence of such a result, $||\mathbf{r}||$ takes the place of the denominator and acts as a *proxy* for the optimal value of Problem 2. The key observation to make is that the right-hand side increases with a known function of γ, and decreases when there is larger separation in the scores of the two top-ranked queues.

The bound we have derived is tight for any fixed $0 \leq \gamma \leq \pi/4$, in the sense that an example can be constructed with property vectors $\mathbf{F}$ so that the left-hand side is arbitrarily close to the right-hand side in Proposition 3. Example 2 illustrates this observation. For $\gamma > \pi/4$, the bound is not necessarily tight.

4.2. Lookahead Heuristic

The projection heuristic is a *geometric* approach to solving the problem approximately. It is also possible to frame the problem as a sequence of individual allocations and make use of a *combinatorial* approach. This section describes such an approach and calls it the *lookahead* heuristic. It requires some new notation and a definition of the problem as a sequence of decisions – ultimately leading to formulation with a Bellman equation.

Note that a complete allocation can be constructed in J steps, where at each step a resource is selected and allocated to a recipient. At the conclusion of this process we

Example 2. For a fixed $\gamma > 0$, a single resource type ($J = 1$), four recipient types ($I = 2$) and one property ($K = 1$), consider a property vector and $\hat{\mathbf{r}}$ given by:

$$\hat{\mathbf{r}}^{\intercal} = \mathbf{F}^{\intercal} = \left(\frac{1}{\sqrt{2}} + \epsilon, \frac{1}{\sqrt{2}} \right)$$

where $\epsilon > 0$ is a small positive number. Note that a basis for the subspace orthogonal to $\mathcal{S}$ is:

$$\mathbf{G} \approx \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

The projection solution $\hat{\mathbf{r}}$ ranks recipient type 1 highest, so the reward it obtains is given by r_1. On the other hand, there is a scoring rule ($\mathbf{w} = (0, 1)$) under which type 2 is ranked highest. So for any $\mathbf{r}$, its absolute suboptimality is $\max(0, r_2 - r_1)$.

Now consider an adversary who constructs $\mathbf{r}$ to make $\hat{\mathbf{r}}$ perform as poorly as possible, and with the angle between $\mathbf{r}$ and $\hat{\mathbf{r}}$ no more than γ. Therefore they aim to solve:

$$\max_{\delta \leq \tan(\gamma)} \quad \left(\hat{r}_1 - \delta \frac{1}{\sqrt{2}} \right) + \left(\hat{r}_2 + \delta \frac{1}{\sqrt{2}} \right)$$

For $\gamma \leq \frac{\pi}{2}$, they will choose $\delta = 1$ and $\mathbf{r} \approx (0, \sqrt{2})$. The absolute suboptimality is $r_2 - r_1 = \sqrt{2}$, and we have:

$$\frac{z^* - z}{||\mathbf{r}||} \approx \frac{\sqrt{2}}{\sqrt{2}} \approx \max \left(0, \frac{\hat{r}_2 - \hat{r}_1}{||\hat{\mathbf{r}}||} + \sqrt{2} \sin(\gamma) \right)$$

must be left with a feasible allocation, which can be ensured by maintaining feasibility of the partial allocation as it is built up.

To this end, we let a partial allocation that has been produced in D steps be denoted by $\mathbf{z}^D := \{(i_1, j_1), \ldots (i_D, j_D)\}$. Here, j_d is the d^{th} resource to be allocated and i_d is the recipient which was allocated this resource. The set of scoring rules which are consistent with this partial allocation is:

$$\mathcal{W}^D = \bigcap_{d=1}^{D} \mathcal{W}_{i_d j_d} \tag{16}$$

and the set of recipients (say $\bar{\mathcal{I}}_j$) which can therefore be selected for some $j \in \bar{\mathcal{J}}$ to extend the partial allocation and maintain feasibility is given by:

$$\bar{\mathcal{I}}_j = \{i : \mathcal{W}_{ij} \cap \mathcal{W}^D \neq \emptyset\} \tag{17}$$

Now let the value function $f(\mathbf{z}^D)$ be the remaining reward that can be generated when resources $\bar{\mathcal{J}} := [J] \setminus \{j_1, \ldots, j_D\}$ are still to be allocated to complete the partial allocation $\mathbf{z}^D$. The Bellman equation that describes the optimal structure of the problem is:

$$f(\mathbf{z}^D) = \max_{j \in \bar{\mathcal{J}},\, i \in \bar{\mathcal{I}}_j} \left[r_{ij} + f(\mathbf{z}^D \cup \{(i,j)\}) \right] \tag{18}$$

This equation says the following: at each step, the authority must choose a resource that has not yet been allocated ($j \in \bar{\mathcal{J}}$) and a recipient who can be matched to this resource when restricted to scoring rules that are consistent with the partial allocation ($i \in \bar{\mathcal{I}}_j$). This should maximize the sum of the reward obtained due to the pair and the remaining reward that can be obtained by extending the partial allocation in this way.

While $f(\mathbf{z}^D)$ cannot be computed directly, it is possible to compute an upper bound that helps to define the lookahead heuristic. This upper bound, denoted by $\bar{f}(\mathbf{z}^D)$, is simple: the sum of the maximum rewards that may be obtained for each remaining resource in $\bar{\mathcal{J}}$ while choosing only from scoring rules that are consistent with the partial allocation $\mathbf{z}^D$:

$$\bar{f}(\mathbf{z}^D) = \sum_{j \in \bar{\mathcal{J}}} \max_{i \in \bar{\mathcal{I}}_j} r_{ij} \tag{19}$$

The lookahead heuristic uses this upper bound to repeatedly extend a partial solution into a full solution by selecting, at each step, the pair which maximimizes the following expression:

$$\max_{j \in \bar{\mathcal{J}},\, i \in \bar{\mathcal{I}}_j} \left[r_{ij} + \bar{f}(\mathbf{z}^D \cup \{(i,j)\}) \right] \tag{20}$$

4.2.1. Analysis

Proposition 5. *The lookahead heuristic approximates the optimal solution within a factor of* $J - 1$. *This bound is (arbitrarily) tight.*

A proof for the first part of Proposition 5 is included in Appendix D. Example 3 provides a tight example for $J = 3$ (which is easily extended for $J \geq 4$) where the lookahead heuristic approximates the optimal solution within a factor arbitrarily close to 2.

It is also worth noting that the lookahead heuristic finds the optimal solution when the data is generated according to the process described in Section 4.1 with $\gamma = 0$. The argument is straightforward: when $\gamma = 0$, $\bar{f}(\mathbf{z}^D) = f(\mathbf{z}^D)$ for any partial solution $\mathbf{z}^D$, so the lookahead heuristic will pick pairs in decreasing order of their reward. It is difficult to find an bound on suboptimality for the lookahead heuristic in terms of γ but this observation is the reason for testing the two heuristics on problem instances parameterized by γ in the next section.

4.3. Numerical Results

We tested the performance of both heuristics on randomly generated data. We used $I = J = 10$ and $K = 5$ so that the MIP formulation could be solved to optimality and the exact optimality gap of each heuristic could be measured. All elements in $\mathbf{F}$ were generated independently and uniformly at random on the unit interval $[0, 1]$. A random point $\hat{\mathbf{r}} \in \bar{\mathcal{S}}$ was generated along with $\delta \in \bar{\mathcal{S}}^{\perp}$ such that $||\delta||_2 = 1$. Then, γ was selected uniformly at random from the interval $[0, \frac{\pi}{2}]$ and the reward coefficients were set to be $\mathbf{r} = \hat{\mathbf{r}} + ||\hat{\mathbf{r}}|| \tan(\gamma)\delta$ so that the angle between $\hat{\mathbf{r}}$ and $\mathbf{r}$ was γ. We generated 500 of these problem instances and solved the formulation in Problem 2 to optimality before comparing with the projection solution.

Figure 7 shows the optimality gap achieved by the two heuristics on each of the problem instances. When $\gamma \approx 0$, both heuristics are optimal. When $\gamma = \frac{\pi}{8} = 22.5°$, the optimality gaps within a single standard error of a Loess regression model are between 2% and 13% for the lookahead heuristic and between 12% and 30% for the projection heuristic. At $\gamma = \frac{\pi}{4} = 45°$ they are between 10% and 33%, and 45% and 75% respectively. Though a bound on performance in terms of γ could not be found for the lookahead heuristic, it is the better performer in these experiments.

Example 3. Suppose $I = 4$ and $J = 3$. Figure 6 provides an alternative representation of the problem data: instead of being in terms of the property vectors, $\mathbf{f}_{ij}$, it shows the sets $\mathcal{W}_{ij}$ where recipient i is the top-scoring type for resource j. The corresponding pair (i, j) is denoted by the reward coefficient for this match.

The lookahead heuristic begins with $\mathbf{z}^0 = \emptyset$. The unique choice for the first match is $\mathbf{z}^1 = \{(1, 1)\}$ since this produces $r_{11} + \bar{f}(\{(1, 1)\}) = 2$ and it can be readily observed that no other selection is able to match this value.

But the partial solution $\mathbf{z}^1$ can only gain a reward of one unit (from either $(1, 2)$ or $(4, 3)$) when completed into a full solution. Compared with the optimal solution $\mathbf{z} = \{(3, 1), (3, 2), (3, 3)\}$ with an objective value of $2 - 2\delta$, the approximatoin ratio of the lookahead solution is $1/(2 - 2\delta)$.

Essentially, this effect occurs when the first selection made by the lookahead solution is maximally naive. The example can be easily extended for any $J \geq 4$ to obtain an optimality bound arbitrarily close to $1/(J - 1)$.

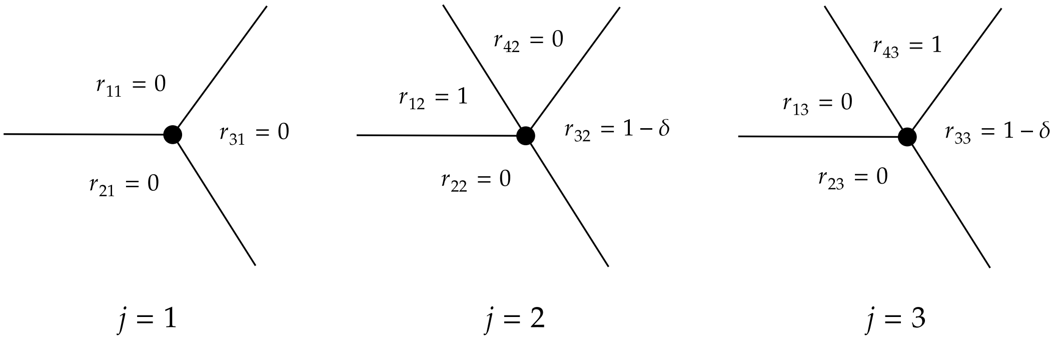

Figure 6: Problem data for the example. Each separate diagram shows the reward coefficients and structure of the $\mathcal{W}_{ij}$ sets for a resource type j. For example, the cone of scoring rules which corresponds to r_{11} in the diagram is $\mathcal{W}_{11}$.

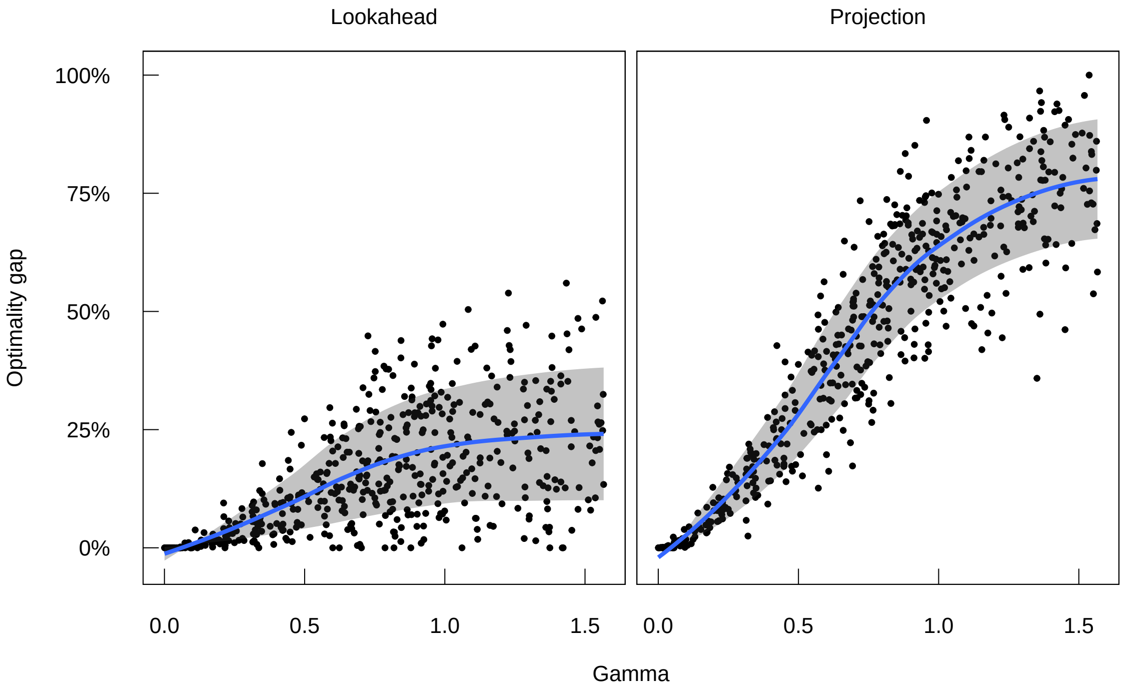

Figure 7: Optimality gap for the two heuristics as the value of γ is varied when generating the data. The fitted line is from a Loess regression model and the shaded band is the single standard error range.

5. Modifying a Scoring Rule for Individual Allocation

In practice, scoring rule systems allocate individual resources to individual recipients. Our type-based model is an accurate one in systems where every individual can be assigned to one of a relatively small number of types and individuals with the same types share identical properties. On the other hand, there are systems where this is not possible and individuals which share a type only have *similar* properties. The kidney allocation system in the US includes time spent on the waiting list as a property, and since this variable is continuous, individuals cannot be divided into a small set of types which share identical properties. This section explains how Problem 2 or the heuristics in Section 4 may fail when designing for this type of system and suggests an approach to find a more appropriate scoring rule.

5.1. A Model for Allocating Individuals

We first establish a model which allocates individual resources to individual recipients and use it to make the idea of a discrepancy relative to the previous type-based model more precise. The model is very simple, allocating a finite set of resources to a finite set of recipients, but the ideas it illustrates are relevant even when the model is extended to a more complicated setting with stochasticity and infinite streams of recipient and resource arrivals.

Individual resources are indexed by $\mathcal{B}$ and individual recipients by $\mathcal{A}$. The properties relating a pair of individuals, $(a, b) \in \mathcal{A} \times \mathcal{B}$, are denoted $\bar{\mathbf{f}}_{ab} \in \mathbb{R}^K$. For some fixed scoring rule $s : \mathbb{R}^K \to \mathbb{R}$, the score relating two individuals is $\bar{s}_{ab} = s(\bar{\mathbf{f}}_{ab})$ and the reward obtained by a match is $\bar{r}_{ab} \in \mathbb{R}_+$.

The model retains the notion of types: I recipient types and J resource types. Let the type of recipient a be $\alpha_a \in [I]$ and the type of resource b be $\beta_b \in [J]$. Assume that it is possible to find a *representative* property vector that relates a pair of types, $(i, j) \in [I] \times [J]$, and call this $\mathbf{f}_{ij} \in \mathbb{R}^K$. In keeping with this notation, a score relating two types is $s_{ab} = s(\mathbf{f}_{ab})$ and a representative reward for matching two types is $r_{ij} \in \mathbb{R}_+$. Let the rate λ_i be the fraction of recipients in $\mathcal{A}$ who are type i (and let μ_j be computed similarly for the resources).

The procedure for constructing an allocation of individuals is a natural one. Let $\mathcal{M} \subseteq$

$\mathcal{A}$ be the set of recipients who have already been matched, starting with $\mathcal{M} = \emptyset$. The resources arrive in order. When $b \in \mathcal{B}$ arrives, we compute all scores for the recipients who remain, $\mathcal{A} \setminus \mathcal{M}$, and apply an *allocation mechanism* to these scores to select a recipient a and update $\mathcal{M} \leftarrow \mathcal{M} \cup \{a\}$. The allocation mechanism is a set of functions:

Definition 4 (Allocation Mechanism). *An allocation mechanism is a family of (possibly randomized) functions, $\{g_m\}_{m=1}^{|\mathcal{A}|}$. Each of the functions has the signature:*

$$g_m : \mathbb{R}^m \mapsto [m] \tag{21}$$

g_m takes takes as input the vector of m scores from the recipients who are waiting for a resource, and selects one of these recipients to allocate the next resource.

Let $\bar{x}_{ij}(\mathbf{w})$ be the fraction of all individual type j resources allocated to individual type i individual recipients, and note that we have parameterized the input with the scoring rule rather than scores, since a scoring rule generates an allocation in both the individual and type-based models. Now, by specifying some allocation function $x_{ij}(\mathbf{w})$ in the type-based model and using the properties $\mathbf{F}$, rewards $\mathbf{r}$, and rates $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ previously defined, we obtain a corresponding type-based model. The discrepancy maps a scoring rule onto a measure of difference between the individual allocations and the underlying type-based allocations:

$$d_p(\mathbf{w}) = ||\bar{\mathbf{x}}(\mathbf{w}) - \mathbf{x}(\mathbf{w})||_p \tag{22}$$

5.2. Where Do Discrepancies Arise?

Under the priority allocation mechanism, a large discrepancy occurs when within-type variation in properties (and therefore scores) causes the type of top-ranked individual recipients to differ from the type-based model. Example 4 shows that even arbitrarily small within-type variation can lead to a scoring rule that is optimal for the underlying type-based model performing poorly when used to allocate individuals.

If a scoring rule leads to a large discrepancy, we must accept that the type-based model is not a good approximation of the individual allocation under this scoring rule. On the other hand, if it leads to a small discrepancy then the approximation is likely a good one. If it is known that the scoring rule is optimal in the type-based model and that the approximation is good, then we can hope it also performs well when used to allocate

Example 4. This example preserves notation from Section 2 and uses a set of recipients $\mathcal{A} = \{1, 2, 3\}$. The individual data is:

$$\bar{\mathbf{f}}_1 = (0, 1),\ \bar{r}_1 = 2 \qquad \bar{\mathbf{f}}_2 = (1 - 3\delta, 0),\ \bar{r}_2 = 1 - \delta \qquad \bar{\mathbf{f}}_3 = (1 + \delta, 0),\ \bar{r}_3 = 1 + \delta$$

There are two recipient types, with $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = 2$. The representative data are taken to be the within-type averages:

$$\mathbf{f}_1 = (0, 1),\ r_1 = 2 \qquad \mathbf{f}_2 = (1 - \delta, 0),\ r_2 = 1$$

The scoring rule $\mathbf{w}_1 = (1, 1)$ is optimal when Problem 2 is solved on the approximate dataset because it ranks recipient 1 (with the maximum reward) highest. But, if $\mathbf{w}_1$ is used to allocate resources in the original dataset, recipient 3 has the highest score and will be allocated the resource (though they do not have the highest reward).

A better scoring rule is $\mathbf{w}_2 = (0, 1)$. This choice places more weight on the property that differentiates the two recipient types in the fluid model and separates the scores of individuals from each type further. Even when the properties of type 2 recipients vary about $\mathbf{f}_2$, the resource will be allocated to a recipient of type 1.

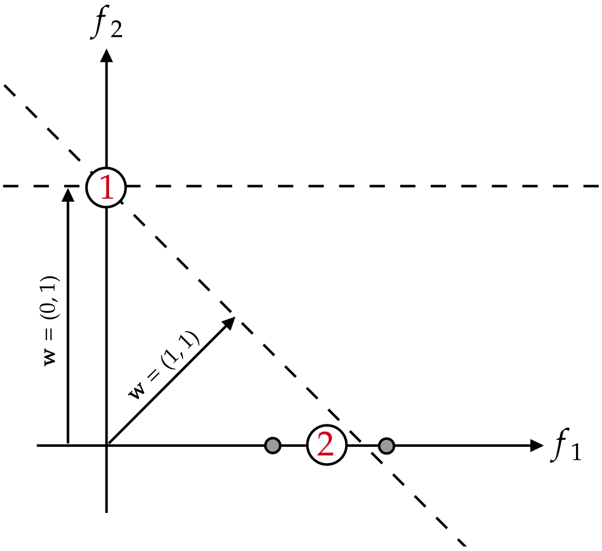

Figure 8: An illustration of the setup in Example 4. Red labels indicate property vectors in the approximate dataset, and the small grey points indicate variation of individuals. The reward for the recipient type 1 is $r_1 = 2$, and for recipient type 2 it is $r_2 = 1$.

individuals. This chain of reasoning is the basis for the heuristic presented in the next subsection.

5.3. Minimizing Discrepancy

It is quite natural to form an optimization problem that balances maximizing the objective value of the type-based model with minimizing discrepancy between the individual model and type-based model. By letting $\eta \geq 0$ be the parameter in this tradeoff, we obtain:

$$\max_{\mathbf{w}} \quad \sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij} x_{ij}(\mathbf{w}) - \eta d_p(\mathbf{w}) \tag{23}$$

Clearly, when $\eta = 0$ the problem reduces to Problem 2 and there is a tractable MIP formulation available. For any other value $\eta > 0$, the problem is harder to solve given that $\bar{\mathbf{x}}(\mathbf{w})$ is a complex function that depends on potentially *many* individual allocations. In fact, if we had a tractable representation we could simply optimize $\sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij} \bar{x}_{ij}(\mathbf{w})$.

A tractable problem can be obtained by replacing the discrepancy term with a surrogate which measures the minimum difference between the top-ranked and second-ranked scores of the recipient-resource type pairs. Maximizing this quantity is likely to reduce the size of the discrepancy (by our previous discussion). More precisely, define:

$$\bar{d}(\mathbf{w}) = \min_{j \in [J]} \left(\mathbf{s}_j(\mathbf{w})^{(1)} - \mathbf{s}_j(\mathbf{w})^{(2)} \right)$$

where $\mathbf{s}_j(\mathbf{w})^{(i)}$ is the ith highest-ranked recipient type for resource j. Our new problem becomes:

$$\max_{\mathbf{w}} \quad \sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij} x_{ij}(\mathbf{w}) + \eta \bar{d}(\mathbf{w}) \tag{24}$$

$\bar{d}(\mathbf{w})$ remains a nontrivial function to model – so we are not out of the woods. The remainder of this section shows how to solve the problem efficiently for the regime where $\eta \to 0$ in which case we obtain a linear program. Appendix E provides a tractable linear MIP formulation for the general case where $\eta > 0$.

When $\eta \to 0$, the problem becomes to maximize $\bar{d}(\mathbf{s})$ over the set of optimal solutions to Problem 2, denoted $\mathcal{W}^*$ and obtained from Equation (10). A formulation is presented in Problem 3.

Problem 3 (Margin Formulation)**.**

$$\max_{\mathbf{w}\in\mathcal{W}^*,\,\mathbf{s},\,\gamma,\,z} \quad z \tag{25a}$$

$$\text{subject to} \quad z \leq s_{i_j^* j} - \gamma_j \qquad \forall j \in [J] \tag{25b}$$

$$\gamma_j \geq s_{ij} \qquad \forall j \in [J],\ i \neq i_j^* \tag{25c}$$

$$s_{ij} = \mathbf{f}_{ij}^\mathsf{T}\mathbf{w} \qquad \forall i \in [I],\ j \in [J] \tag{25d}$$

Since the objective is to maximize z, the γ_j terms in Equation (25b) are set to be as small as the constraints allow. Equation (25c) therefore ensures that γ_j takes on the score of the second-ranked recipient type for resource type j, and so each right-hand side term in Equation (25b) measures a score difference between the optimal recipient type and the second-ranked recipient type for that resource type j. The objective maximizes the minimum of these differences.

Problem 3 does not explicitly take into account the individual variation of recipients and resources around the approximate properties of the types – it only aims to maximize the differences of the approximate properties themselves. A more useful approach would take into account the details of the variation whilst also remaining tractable to solve.

Problem 4 is a formulation which addresses this issue. We first define two well-known functions (for fixed values of L_{ij}), where $\mathcal{D}_{ij} = \{(a,b) \in \mathcal{A} \times \mathcal{B} : \sigma(a) = i,\ \sigma(b) = j\}$ is the set of pairs of individual recipients and resources with the corresponding types.

Definition 5 (Mean Top-L Function)**.**

$$\begin{aligned} g_{ij}(\mathbf{s}) = \quad & \max_{\mathbf{t}} \quad \frac{1}{L_{ij}} \sum_{(a,b)\in\mathcal{D}_{ij}} s_{ab} t_{ab} \\ & \text{subject to} \quad \mathbf{e}^\mathsf{T}\mathbf{t} = L_{ij} \\ & \qquad\qquad\quad \mathbf{t} \in \{0,1\}^{|\mathcal{D}_{ij}|} \end{aligned}$$

Definition 6 (Mean Bottom-L Function)**.**

$$\begin{aligned} h_{ij}(\mathbf{s}) = \quad & \min_{\mathbf{t}} \quad \frac{1}{L_{ij}} \sum_{(a,b)\in\mathcal{D}_{ij}} s_{ab} t_{ab} \\ & \text{subject to} \quad \mathbf{e}^\mathsf{T}\mathbf{t} = L_{ij} \\ & \qquad\qquad\quad \mathbf{t} \in \{0,1\}^{|\mathcal{D}_{ij}|} \end{aligned}$$

These functions represent the average of the L_{ij} largest and smallest scores respectively, over all property vectors in the original instance associated with a particular recipient type i and resource type j.

g_{ij} and h_{ij} are convex and concave functions respectively (Boyd & Vandenberghe, 2004). We use them to define the convex optimization in Problem 4, for some values of L_{ij} to be chosen. Note that when $L_{ij} = |\mathcal{D}_{ij}|$ is chosen, the functions return the mean score of the pairing with recipient type i and resource type j. If the mean function had been used to compute the approximate instance, then we recover the formulation in Problem 3.

Problem 4 (Top/Bottom-L Formulation)**.**

$$\begin{aligned} \max_{\mathbf{w}\in\mathcal{W}^*,\, \mathbf{s},\, \gamma,\, z} \quad & z && && \text{(26a)} \\ \text{subject to} \quad & z \leq h_{i_j^* n}(\mathbf{s}) - \gamma_j && \forall j \in [J] && \text{(26b)} \\ & \gamma_j \geq g_{ij}(\mathbf{s}) && \forall j \in [J],\ i \neq i_j^* && \text{(26c)} \\ & s_{ij} = \mathbf{f}_{ij}^{\mathsf{T}} \mathbf{w} && \forall i \in [I],\ j \in [J] && \text{(26d)} \end{aligned}$$

This formulation modifies Problem 3 to take into account only the most *influential* individuals for a particular choice of the scoring rule $\mathbf{w}$. The individuals from suboptimal recipient types who have the highest scores and the individuals from optimal recipient types who have the lowest scores are the most likely candidates for misclassifications – and it is these which the formulation aims to separate. Since g_{ij} and h_{ij} are piecewise linear functions, the problem may be easily reformulated as a linear program.

5.4. Numerical Experiments

We tested this approach on synthetic data for $I = J = 10$ and $K = 5$. Property vectors for each recipient-resource type pairing were first selected independently and uniformly at random on $[0,1]^K$. Then, the dataset of individuals was generated by adding zero-mean Gaussian noise with a diagonal covariance matrix whose entries were generated independently and uniformly at random on $[0,\eta]^K$ for some *variability* parameter $\eta > 0$. The rewards were generated as if they were the $(K+1)$-th property.

A baseline scoring rule was obtained by solving Problem 2 for the underlying type-based instance, and this rule was compared to the rule returned by the solution to Problem 4. Comparisons were made by using both rules to simulate allocations on the original data instance with individuals under the priority selection procedure. The values of L_{ij} varied according to a percentage of $|\mathcal{D}_{ij}|$.

We tracked the discrepancy in allocation $d_1(\mathbf{w})$ under both the baseline and modified scoring rules and report the ratio of these values across data instances in Figure 9. The modified solution gives consistently smaller discrepancies than the baseline solution, but as variability in the data increases this effect becomes less pronounced. At a variability of 0.02 the median reduction in discrepancy is 40% and at a variability of 0.1 the median reduction is 10%.

We see a similar, though smaller, effect on reward. The modified solution leads to an improved reward for all variabilities, though this becomes less as variability increases. The median improvement in reward earned is 5% for a variability of 0.02 and 2% for a variability of 0.1. On both metrics, there was little to distinguish the effect of the different fractions used to compute each L_{ij}.

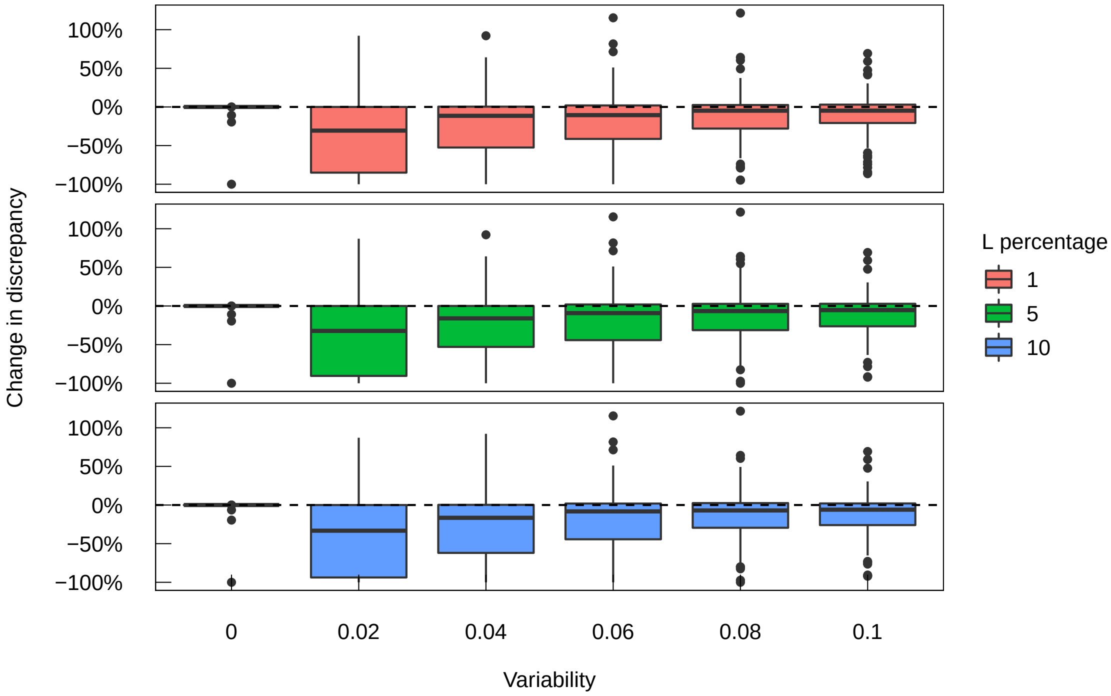

Figure 9: Misclassification ratios of the modified rule relative to the optimal rule, as the variability in the synthetic data increases.

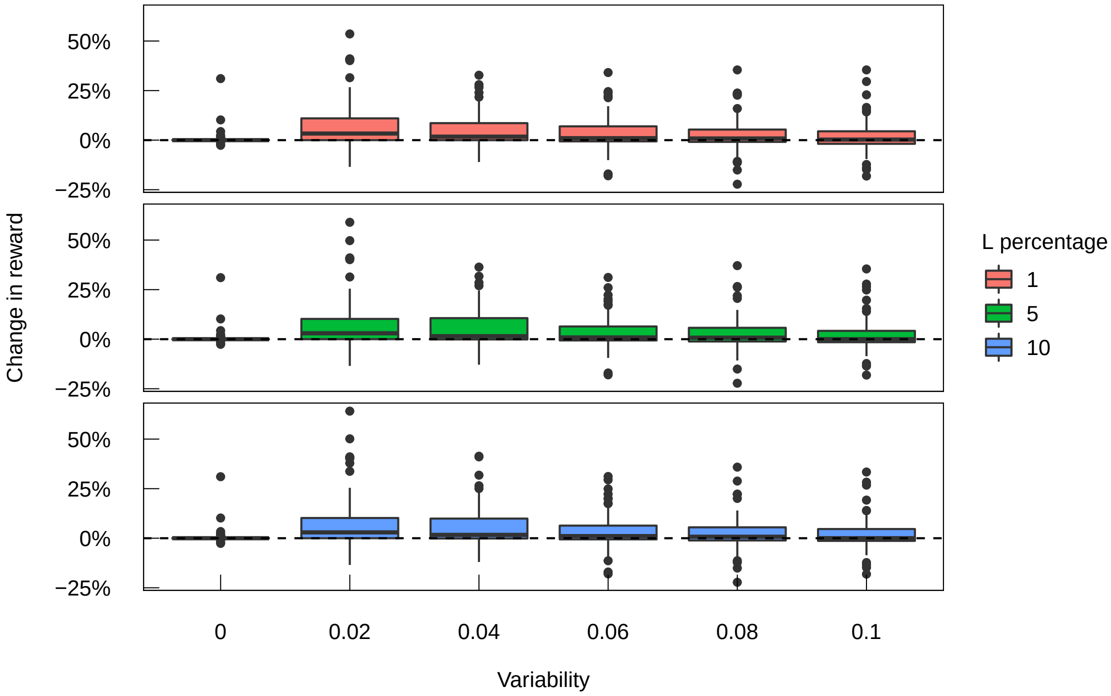

Figure 10: Objective ratio of the modified rule relative to the optimal rule, again as data variability increases.

6. Lottery Allocation

In Section 1 it was noted that the priority allocation mechanism may not always be the appropriate choice. It has the fundamental property that when two recipients have nearly identical scores, the one with the larger score receives the resource ahead of the other – but it is easy to imagine a system where two recipients with nearly identical scores have a similar random *chance* of receiving the resource. Systems which implement a random procedure such as this are commonly referred to as *lottery* systems. Leaving the outcome of an allocation up to chance appeals to an intuitive human notion of fairness which helps to explain why these systems are proposed time and again – including, most topically, for distributing scarce medical resources in the midst of the COVID-19 pandemic (Emanuel et al., 2020; Persad, Wertheimer, & Emanuel, 2009).

We will focus our attention on a variation of a lottery system called a *weighted lottery* (Saunders, 2009) in which the probabilities of receiving a resource may be different for different recipients. More precisely, the lottery mechanism we study computes scores for each waiting recipient and then randomly allocates a resource amongst them with probabilities that are proportional to their scores: if $s_{ab} = 1$ and $s_{a'b} = 2$, then recipient a' should have twice the chance of receiving resource b as recipient a.

There are many real-world examples of lottery systems that make use of scoring rules. One system was recently implemented in Philadelphia and used to distribute scarce medications for treating COVID-19 patients (White & Angus, 2020; Iyer, Hendriks, & Rid, 2020; Jansen & Wall, 2021). Another example is the Dutch system for allocating admissions to medical schools, which was run until 2017 as a weighted lottery with student scores equal to their high-school GPAs (Ten Cate, 2021) and is proposed to be reintroduced in 2023.

6.1. Defining the Allocation Function

The first step in modelling this lottery allocation mechanism is to define the corresponding allocation function in the fluid model and the scores on which it is defined. We start with the scores, which must simply be nonnegative and nonzero (so that the probabilities of allocation can be proportional to them):

$$\mathcal{S} = \left\{ \mathbf{s} \in \mathbb{R}^{I \times J} : \mathbf{s} \geq \mathbf{0} \right\} \setminus \{\mathbf{0}\} \tag{27}$$

The scores chosen by the authority must also satisfy this property:

$$\bar{\mathcal{S}} = \{\mathbf{Fw} \geq \mathbf{0} : \mathbf{w} \neq 0\} \setminus \{\mathbf{0}\} \tag{28}$$

Next we fix some some scores, $\mathbf{s} \in \mathcal{S}$, and attempt to define the allocation function $x_{ij}(\mathbf{s})$ so that it represents the probability that a resource of type j is allocated to a recipient of type i. Note that the total *weight* held collectively by recipient type i is $s_{ij}L_{ij} = \frac{s_{ij}}{q_i}\left(\lambda_i - \sum_{k=1}^{J} \mu_k x_{ik}\right)$. The allocation fractions must therefore satisfy the following expression for each recipient type i and resource type j (which is written in terms of the operator T_{ij}):

$$x_{ij} = \frac{s_{ij}L_{ij}}{\sum_{l=1}^{I} s_{lj}L_{lj}} = \frac{\frac{s_{ij}}{q_i}\left(\lambda_i - \sum_{k=1}^{J} \mu_k x_{ik}\right)}{\sum_{l=1}^{I} \frac{s_{lj}}{q_l}\left(\lambda_l - \sum_{k=1}^{J} \mu_k x_{lk}\right)} := T_{ij}(\mathbf{x}) \tag{29}$$

This expression does not give us the allocation fractions in closed form. In fact, computing the fractions amounts to finding a fixed point of the operator $T(\mathbf{x}) = [T_{ij}(\mathbf{x})] \in \mathcal{X}$. It is not immediately clear this operator has a unique fixed point or indeed that it has any fixed point at all. But if the fractions $\mathbf{x}$ are to represent a valid allocation *function* from the set of scores onto $\mathcal{X}$, then a fixed point must both exist and be unique.

Let us first establish the *existence* of a fixed point. It is clear that $T(\mathbf{x}) \in \mathcal{X}$ when $\mathbf{x} \in \mathcal{X}$ due to the scarcity assumption (the numerators in $T_{ij}(\mathbf{x})$ are nonnegative) and the normalization terms (which ensure $\sum_{i=1}^{I} T_{ij}(\mathbf{x}) = 1$). Since $\mathcal{X}$ is compact and convex, and $T(\mathbf{x})$ is continuous, the Brouwer fixed-point theorem applies and ensures that T has at least one fixed point.

Next we turn to *uniqueness* of the fixed point, though a proof remains elusive. T is not a contraction mapping and therefore the Banach fixed-point theorem cannot be applied. On the other hand, Appendix F provides the results of extensive numerical testing showing that Banach-Picard iterations converge to a unique fixed point in all instances and with generally very few iterations. We strongly suspect that T has a unique fixed point, and note that this point is easy to compute. In any case, our optimization formulation does not rely on the existence of a unique fixed point.

6.2. Optimization Formulation

Though an allocation function satisfying the ratios in Equation (29) is likely to exist, we do not have this function in closed form. This subsection formulates the optimization problem faced by the authority as a nonconvex bilinear program. This class of problems are known to be NP-hard (and generally hard to solve in practice) but modern solvers make use of a MIP reformulation that can solve instances of reasonable size either to provable optimality or with a well-quantified optimality gap. We include numerical results of our formulation on the same synthetic data that was used in Section 3.

The key to the formulation is to introduce variables $\boldsymbol{\sigma}$ and constrain them to represent the normalization terms appearing in Equation (29). The formulation is in Problem 5 and followed with a justification of its correctness:

Problem 5 (Lottery Selection Optimization)**.**

$$\max_{\mathbf{w},\, \mathbf{s},\, \mathbf{x},\, \boldsymbol{\sigma}} \quad \sum_{j=1}^{J} \sum_{i=1}^{I} \bar{r}_{ij} x_{ij} \tag{30a}$$

$$\text{subject to} \quad \sigma_j x_{ij} = \frac{s_{ij}}{q_i} \left(\lambda_i - \sum_{j'=1}^{J} \mu_{j'} x_{ij'} \right) \quad \forall i \in [I],\ j \in [J] \tag{30b}$$

$$\sum_{i=1}^{I} x_{ij} = 1 \quad \forall j \in [J] \tag{30c}$$

$$s_{ij} = \mathbf{w}^{\mathsf{T}} \mathbf{f}_{ij} \quad \forall i \in [I],\ j \in [J] \tag{30d}$$

$$\mathbf{e}^{\mathsf{T}} \mathbf{w} = 1 \tag{30e}$$

$$\mathbf{s} \geq \mathbf{0} \tag{30f}$$

$$\mathbf{x} \geq \mathbf{0} \tag{30g}$$

$$\boldsymbol{\sigma} \geq \mathbf{0} \tag{30h}$$

Equation (30b) ensures that the allocation fractions are constrained correctly provided that each σ_j takes on the correct value of the normalization term. To confirm that the σ_j variables are indeed defined correctly, note that for each $j \in [J]$:

$$\sigma_j = \left(\sum_{i=1}^{I} x_{ij} \right) \sigma_j = \sum_{i=1}^{I} x_{ij} \sigma_j = \sum_{i=1}^{I} \frac{s_{ij}}{q_i} \left(\lambda_i - \sum_{j'=1}^{J} \mu_{j'} x_{ij'} \right) \tag{31}$$

where the first equality follows from Equation (30e) and the third from Equation (30b).

The normalization constraint in Equation (30e) is again included to ensure that the resulting scores lie in $\mathcal{S}$. The same comment as in Problem 2 applies: the problem can be re-solved for $\mathbf{e}^\intercal \mathbf{w} = -1$ and perturbations of the properties to cover the entire search space of $\mathbf{w}$.

We tested the scalability of this formulation on data generated randomly in the same way as described in Section 3.3, with the addition of random $\boldsymbol{\mu}$ values from $[0, 1]$ and random $\boldsymbol{\lambda}$ values from $[0, 1]$ which were then scaled to satisfy Assumption 1. All tests were conducted with a time limit of 5 hours on an Intel Xeon 2.1 GHz quad-core CPU with 32 GB of RAM and solved with the nonconvex bilinear procedure in Gurobi 9.1.1.

Figure 11 shows how the range of solution times changed as the number of customer and resource classes in the instance increased. There was little to distinguish between $K = 5$ and $K = 10$, and problems with up to approximately 15 types could be solved to provable optimality within the time limit. While this is fewer than the 35 types that could be solved to optimality for the priority allocation procedure with $K = 5$, it is still a problem size that is useful for modelling purposes.

Figure 12 shows how the range of optimality gaps changed as the number of types was varied. Once again, changing $K = 5$ to $K = 10$ did not change the optimality gaps appreciably. However, these gaps grew quickly once the problem could not be solved within 5 hours. When $K = 5$ and the number of types was greater than 20, the optimality gaps were between 75% and 100%.

6.3. Heuristic Performance

Problem 5 is difficult to solve, and it is therefore useful to study heuristics for solving it. While the lookahead heuristic from Section 4 cannot be easily re-purposed for this problem, the projection heuristic can.

The motivation for applying the projection heuristic to the lottery allocation function is similar, though not identical, to the previous case. Suppose that we are able to reproduce the reward coefficients with a scoring rule $\hat{\mathbf{w}} \in \mathcal{W}$. Then, for any given resource type j, a recipient type that earns a high reward will tend to receive a larger share of the resource and a recipient type that earns a low reward will tend to receive a smaller share (which should also likely be true in the optimal solution).

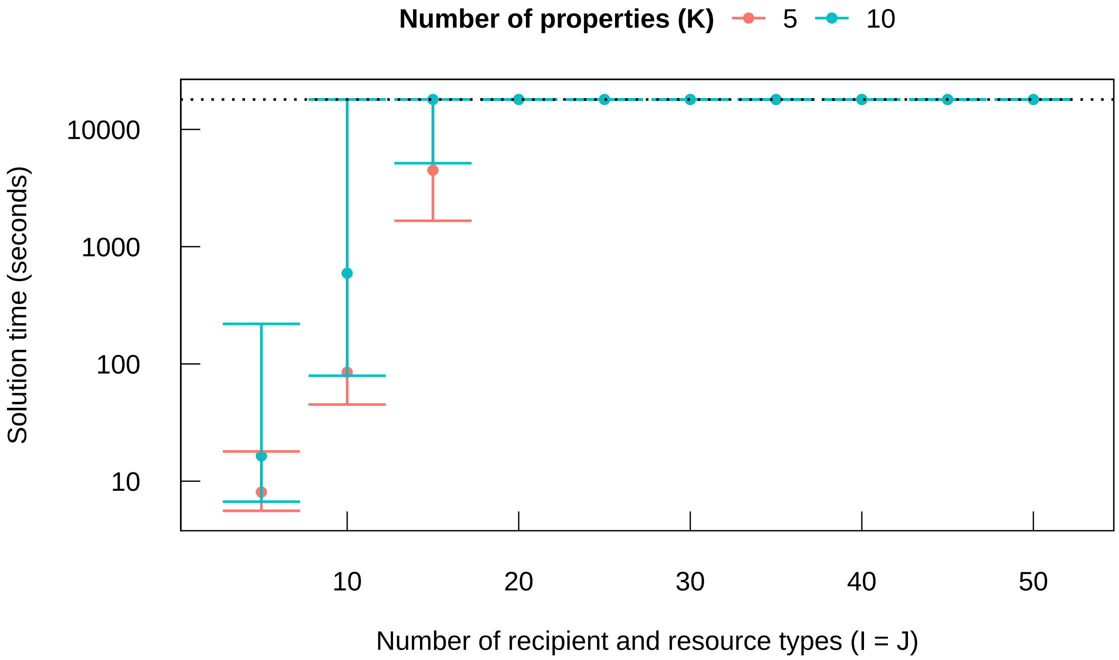

Figure 11: Ranges of solution times when instances were solved to provable optimality. The dashed line indicates the 5 hour time limit imposed on the solver.

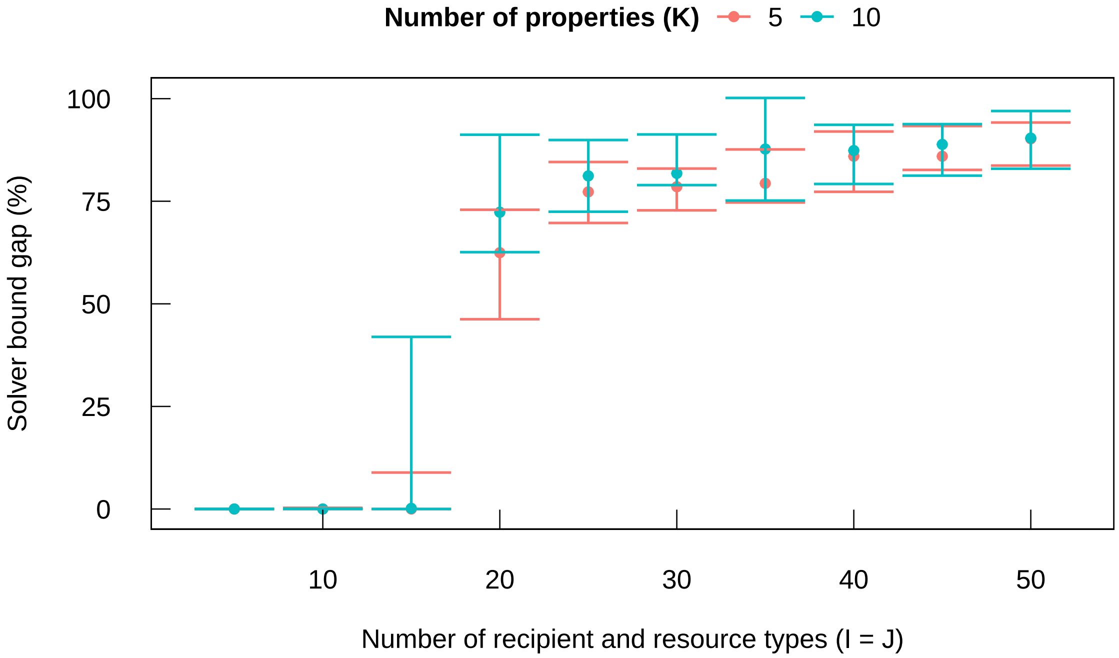

Figure 12: Ranges of the solver optimality gap after the 5 hour time limit had expired.

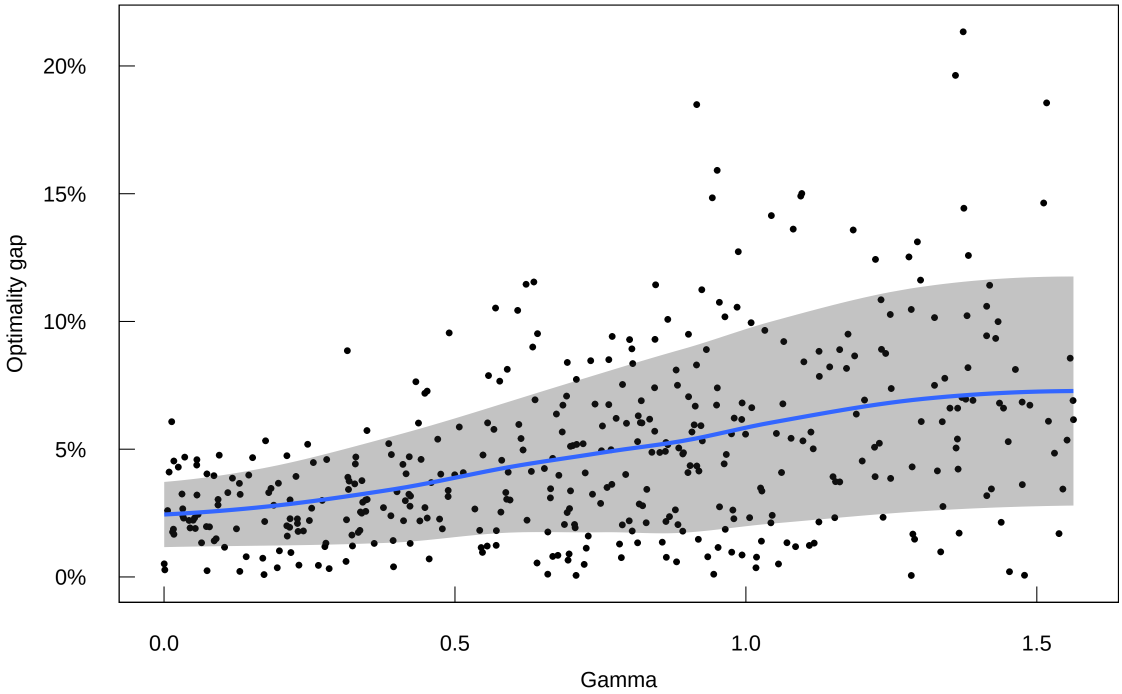

Figure 13: Optimality gap achieved by the projection solution under the lottery mechanism.

The technique used to bound performance in Section 4 does not apply to this problem, and we turn instead to the computational analysis in Figure 13. Results are shown for the same instances that were generated in Section 4.3. When $\gamma = \frac{\pi}{8} = 22.5°$ the solutions obtained from the projection heuristic have optimality gaps within a single standard error of between 2% and 6%. When $\gamma = \frac{\pi}{4} = 45°$ they are between 2% and 8%.

6.4. Modifying for Individual Allocation

In this section we consider how to approach the problem of allocating individuals under the lottery system. Equation (24) is a good starting point, but the surrogate discrepancy function we included in the objective for the priority system no longer applies.

For the lottery system, discrepancy is minimized when the scores of individual (a, b) pairs are similar to the scores of their corresponding type pair (α_a, β_b). This observation motivates us to select the surrogate discrepancy function as a least-squares term that compares the scores of the individuals and their representative types, in order to get the problem:

$$\max_{\mathbf{w}} \quad \sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij} x_{ij}(\mathbf{w}) - \eta \sum_{a \in A} \sum_{b \in \mathcal{B}} \left(\bar{s}_{ab}(\mathbf{w}) - s_{\alpha_a \beta_b}(\mathbf{w}) \right)^2 \tag{32}$$

For general $\eta > 0$ this is a mixed-integer second order cone program (SOCP) and is solvable by several leading commercial solvers, including Gurobi. We again conduct numerical tests on the regime $\eta \to 0$ where the problem reduces to a SOCP. We use the same data instances that were used when testing the priority allocation function in Section 5. Figure 14 shows ratios of the discrepancy under the baseline solution and the modified solution, and Figure 15 shows similar for the reward. The heuristic did not improve either discrepancy or reward earned.

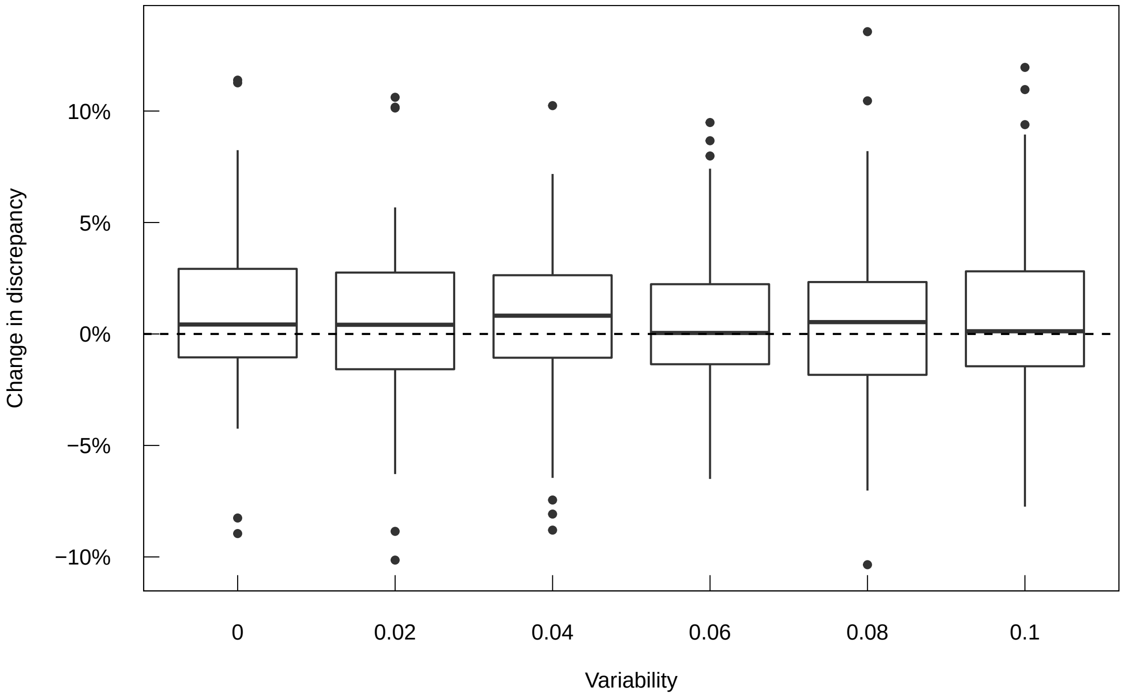

Figure 14: Ratios of the discrepancy under the modified rule relative to the baseline rule from the lottery model, as variability in the synthetic data increases.

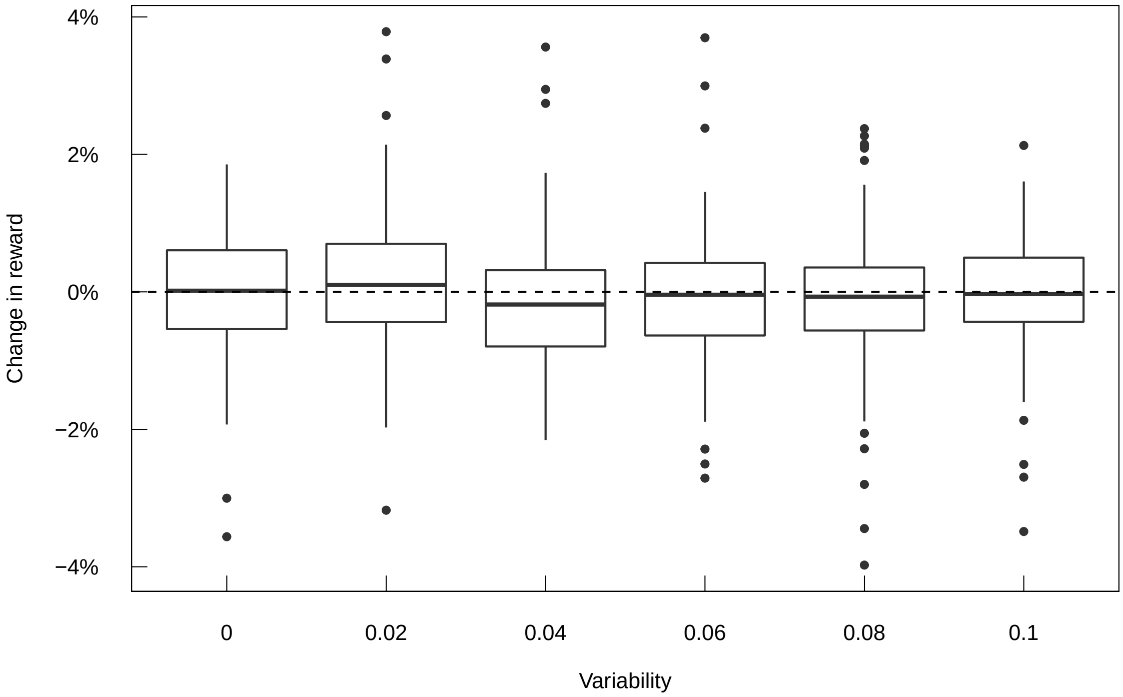

Figure 15: Objective ratio of the modified rule relative to the optimal rule for the lottery model, as data variability increases.

7. Ex-Post Fairness in Allocations

At various points in the paper we have highlighted some desirable aspects of a system for allocating scarce resources – for example, the three principles of equity identified by Young (1994) and the role of chance in a fair mechanism. Each of these properties relate to the *process* followed by the system rather than the outcome of the allocation. It is common for designers to seek mechanisms that also provide some guarantees on the outcome of the allocation: this is *ex-post* fairness. This section provides some comments on modifying our approaches to achieve ex-post fairness.

In our context, an ex-post fairness constraint requires the allocation fractions to satisfy some property. For example: the authority may wish to constrain the proportion of recipients who receive a resource and have some attribute to lie within a certain range. If the first and second recipient types are the only ones with this attribute, then lower and upper bounds can be enforced with a simple linear constraint on the allocation variables:

$$L \leq \sum_{j=1}^{J} \frac{1}{J}(x_{1j} + x_{2j}) \leq U \tag{33}$$

The class of linear constraints is large and covers several flavors of bound on the proportion or average value of an attribute in both the recipient types which are allocated resources and those which are not. These constraints are also interpretable and intuitive – Bertsimas et al. (2013) note that it is a type favoured by policymakers in practice.

To be more precise, we may associate an additional set of N attributes with each recipient type. Let $a_{in} \in \mathbb{R}$ be the nth attribute associated with the ith recipient type. These attributes may be distinct from the properties in each $\mathbf{f}_{ij}$ or they may overlap. The key distinction is that the properties in $\mathbf{f}_{ij}$ are *unprotected* – the system may compute priority scores on their basis. Some of the attributes in each $\mathbf{a}_i$ may be *protected*, meaning that although they can be used to define fairness constraints, they cannot be used as a basis for computing priority scores. Race and gender are two examples of attributes generally considered to be protected in most practical systems.

Enforcing linear ex-post fairness constraints within our framework simply amounts to adding linear constraints on the allocation fractions, $\mathbf{x}$, in Problem 1. This does not alter the problem class of the optimization formulations derived from it – the priority allocation function still gives rise to a linear MIP and the lottery allocation function still gives rise to a nonconvex bilinear program – though the additional constraints presumably increase the time required to solve instances to optimality. However, it is important to note that satisfying even a single ex-post fairness constraint is as hard as the complete optimization problem. This makes enforcing such a constraint in either of the two heuristics we have discussed impossible.

A approximate way of dealing with this trouble is provided by Bertsimas et al. (2013) (where it was used successfully). Here, an offline matching problem with ex-post fairness constraints is solved to obtain optimal dual variables. The reward coefficients are then modified with these dual variables to obtain *fairness-adjusted* coefficients under which the optimal solution to a standard bipartite matching problem satisfies the constraints. Using these fairness-adjusted coefficients as inputs into the projection and lookahead heuristics may provide a rough method for obtaining allocations that approximately satisfy the fairness constraints.

8. Discussion and Conclusions

In this paper, we introduced a simple model for allocating a set of heterogeneous resource types amongst a set of heterogeneous recipient types on the basis of scores computed from observable properties. We used the model to define a general optimization problem faced by a benevolent authority tasked with choosing the scores from a restricted set, and analysed specific instances of the optimization problem associated with two real-world allocation mechanisms: priority allocation and lottery allocation. Our models are attractive for their focus on describing the underlying allocation mechanism – in contrast to other black-box methods that are based on simulation.

Both optimization problems are hard to solve, but tractable with modern optimization techniques and a moderate number of recipient and resource types. The priority model was solved to optimality within five hours with 30 resource and recipient types, and the lottery model with 15 resource and recipient types (though both periods also depended on the number of properties in the scoring rule). Our formulations also allow for fairness constraints to be included without increasing the overall complexity of the models.

For problems larger than these or in a setting where a solution must be obtained quickly, we presented some heuristics. The projection heuristic is instantaneous and approximates the optimal solution to the priority model well when the reward coefficients are *well-explained* by the properties. It also approximates the optimal solution to the lottery model even when reward coefficients are poorly explained by the properties, rarely achieving an optimality gap less than 10%. Our lookahead heuristic was more effective at solving the priority model than the projection heuristic when the reward coefficients are poorly explained by the properties, with a maximum optimality gap around 25%.

We also studied the setting where properties of individual recipients and resources vary within their assigned types, and suggest some strategies for choosing a scoring rule. The optimization problem becomes more complex for both priority and lottery models, but we demonstrate that introducing a surrogate objective function leads to tractable problems. We defined the notion of discrepancy, which measures how different the individual allocation is from the underlying type-based model. On the lottery model, we showed experimentally that the techniques had no benefit; however,the techniques significantly reduced discrepancy and improved the reward earned in the priority model.

A. Characterization of Feasible Allocations

Here we include a proof of Proposition 1:

Proof. First suppose $y = (a, b)$ is a feasible allocation. This means:

$$\mathbf{w}^\intercal f_{a1} > \mathbf{w}^\intercal \bar{\mathbf{f}}_1,\ \forall \bar{\mathbf{f}}_1 \in \mathcal{P}_1 \setminus \{\mathbf{f}_{a1}\} \quad \text{and} \quad \mathbf{w}^\intercal f_{b2} > \mathbf{w}^\intercal \bar{\mathbf{f}}_2,\ \forall \bar{\mathbf{f}}_2 \in \mathcal{P}_1 \setminus \{\mathbf{f}_{b2}\}$$

It follows that $\mathbf{w}^\intercal \left(\mathbf{f}_{a1} + \mathbf{f}_{b2}\right) > \mathbf{w}^\intercal \left(\bar{\mathbf{f}}_1 + \bar{\mathbf{f}}_2\right), \forall \bar{\mathbf{f}}_1 \in \mathcal{P}_1 \setminus \{\mathbf{f}_{a1}\}, \bar{\mathbf{f}}_2 \in \mathcal{P}_2 \setminus \{\mathbf{f}_{b2}\}$. Therefore, $\mathbf{f}_{a1} + \mathbf{f}_{b2}$ is an extreme point of $\mathcal{P}_1 + \mathcal{P}_2$.

Now suppose that $\mathbf{f}_{a1} + \mathbf{f}_{b2}$ is an extreme point of $\mathcal{P}_1 + \mathcal{P}_2$. By definition there is some $\mathbf{w}$ for which:

$$\left(\mathbf{f}_{a1} + \mathbf{f}_{b2}\right)^\intercal \mathbf{w} > \left(\mathbf{g}_1 + \mathbf{g}_2\right)^\intercal \mathbf{w}$$

for any $\mathbf{g}_1 \in \mathcal{P}_1$ and $\mathbf{g}_2 \in \mathcal{P}_2$ such that $\mathbf{g}_1 + \mathbf{g}_2 \neq \mathbf{f}_{a1} + \mathbf{f}_{b2}$.

If there is some $i \neq a$ for which $\mathbf{f}_{i1}^\intercal \mathbf{w} \geq \mathbf{f}_{a1}^\intercal \mathbf{w}$ (meaning that customer class i scores at least as highly as a on resource class 1) we would have:

$$\left(\mathbf{f}_{i1} + \mathbf{f}_{b2}\right)^\intercal \mathbf{w} \geq \left(\mathbf{f}_{a1} + \mathbf{f}_{b2}\right)^\intercal \mathbf{w}$$

which violates the previous observation given that $\mathbf{f}_{i1} + \mathbf{f}_{b2} \neq \mathbf{f}_{a1} + \mathbf{f}_{b2}$ and both $\mathbf{f}_{i1} \in \mathcal{P}_1$ and $\mathbf{f}_{b2} \in \mathcal{P}_2$.

So $\mathbf{f}_{i1}^\intercal \mathbf{w} < \mathbf{f}_{a1}^\intercal \mathbf{w}$ for any $i \neq a$ and customer class a is the unique top-scoring customer class for resource class 1. Similar reasoning applies to b on resource class 2. □

B. NP-Hardness of Priority Scoring Rule Optimization

This section provides a proof that the optimization in Problem 2 is NP-hard. Notation is preserved from Section 2.

We reduce from an instance of the MAX-FLS (feasible linear subsystem) problem in Amaldi and Kann (1995). This problem deals with finding the largest subset of relations that can be satisfied simultaneously in a system of linear inequalities. An instance is defined by a set of J halfspaces given by $H_j = \{\mathbf{w} \in \mathbb{R}^K : \mathbf{a}_j^\intercal \mathbf{w} \geq b_j\}$.

Amaldi and Kann (1995) show that this problem remains NP-hard when the inequalities are homogeneous ($b_j = 0$) and the trivial solution $\mathbf{w} = \mathbf{0}$ is excluded. This is the version of the problem we reduce from.

To see the reduction, set $I = 2$, and then $\mathbf{f}_{1j} = \mathbf{a}_j$, $\mathbf{f}_{2j} = \mathbf{0}$, $r_{1j} = 1$ and $r_{2j} = 0$ for all j. If we are able to find some $\mathbf{w} \neq \mathbf{0}$ so that $\mathbf{a}_j^\intercal \mathbf{w} \geq 0$ then we also have $(\mathbf{f}_{1j} - \mathbf{f}_{2j})^\intercal \mathbf{w} \geq \mathbf{0}$ and therefore claim the reward $r_{1j} = 1$ on the jth resource type. The objective values in both problems is clearly identical and the reduction is complete.

C. Projection Optimality Bound

Our approach to deriving the bound is to fix $\hat{\mathbf{r}}$, and let an adversary pick a reward vector $\mathbf{r}$ which forms an angle no greater than γ in order to maximize the suboptimality of the projection solution. The adversary is only allowed to perturb the reward vector perpendicular to $\bar{\mathcal{S}}$ in order to ensure that $\hat{\mathbf{r}}$ remains the projected scores.

Note that any vector of reward coefficients can be represented in terms of the perturbation away from $\hat{\mathbf{r}}$ as $\mathbf{r} = \hat{\mathbf{r}} + \delta$, where $\delta \perp \bar{\mathcal{S}}$ is a perturbation orthogonal to $\bar{\mathcal{S}}$. The relationship between γ and $||\delta||$ is:

$$\tan(\gamma) = \frac{||\delta||}{||\hat{\mathbf{r}}||}$$

We wish to define the set $\mathbf{\Delta}$ to be all perpendicular perturbations with length less than or equal to some value $\theta := ||\hat{\mathbf{r}}|| \tan(\gamma) > 0$. Let a collection of orthonormal vectors spanning the space orthogonal to $\bar{\mathcal{S}}$ be arranged in columns to get $\mathbf{G} \in \mathbb{R}^{IJ\times(IJ-K)}$. Then $\mathbf{\Delta}$ can be defined as a degenerate ball with radius θ in terms of $\mathbf{G}$:

$$\mathbf{\Delta} = \{\mathbf{G}\mathbf{u} : ||\mathbf{u}||_2 \leq \theta\}$$

We start by establishing the following lemma:

Lemma 1. *Let $J = 1$, and the projected reward coefficients be given by $\hat{\mathbf{r}} \in \mathbb{R}^I$ with $\hat{r}_1 \geq \hat{r}_2 \geq \ldots \geq \hat{r}_I$. Let γ be the angle between $\mathbf{r}$ and $\hat{\mathcal{S}}$ as in Equation (13).*

Let z^* be the optimal objective value of Problem 2, and let z be the objective value of the projection solution. Let $\hat{\mathbf{x}}$ be any set of allocation fractions. The following bound holds for any angle $0 \leq \gamma < \pi/2$:

$$\frac{z^* - z}{||\hat{\mathbf{r}}||} \leq \max\left(\frac{\hat{r}_1}{||\hat{\mathbf{r}}||} + ||\mathbf{e}_1 - \hat{\mathbf{x}}||_2 \tan(\gamma), \ldots, \frac{\hat{r}_I}{||\hat{\mathbf{r}}||} + ||\mathbf{e}_I - \hat{\mathbf{x}}||_2 \tan(\gamma)\right) - \frac{\hat{\mathbf{r}}^\mathsf{T}\hat{\mathbf{x}}}{||\hat{\mathbf{r}}||}$$

Proof. We look for the largest suboptimality that can be induced by the adversary across all feasible perturbations, referring to this quantity as α. Since z^* and z depend on the perturbation we write them as $z^*(\delta)$ and $z(\delta)$. When the adversary perturbs, they are restricted by the fact that $\mathbf{r} \geq \mathbf{0}$, which means that by dropping this restriction we get an upper bound on α as $\alpha \leq \max_{\delta\in\mathbf{\Delta}}(z^*(\delta) - z(\delta))$.

An upper bound on $z^*(\delta)$ is $\max(r_1, \ldots, r_I) = \max(\hat{r}_1 + \delta_1, \ldots, \hat{r}_I + \delta_I)$. Also note that, since the allocation fractions of the projection solution do not change as the adversary perturbs the reward coefficients, we have $z(\delta) = (\hat{\mathbf{r}} + \delta)^\intercal \hat{\mathbf{x}}$. We can write:

$$\begin{aligned}
\alpha &\leq \max_{\delta \in \boldsymbol{\Delta}} \left(z^*(\delta) - (\hat{\mathbf{r}} + \delta)^\intercal \hat{\mathbf{x}}\right) \\
&\leq \max_{\delta \in \boldsymbol{\Delta}} \left(\max(\hat{r}_1 + \delta_1, \ldots, \hat{r}_I + \delta_I) - (\hat{\mathbf{r}} + \delta)^\intercal \hat{\mathbf{x}}\right) \\
&= \max_{\delta \in \boldsymbol{\Delta}} \left(\max\left(\hat{r}_1 + \delta_1 - \delta^\intercal \hat{\mathbf{x}}, \ldots, \hat{r}_I + \delta_I - \delta^\intercal \hat{\mathbf{x}}\right) - \hat{\mathbf{r}}^\intercal \hat{\mathbf{x}}\right)
\end{aligned}$$

The RHS optimization is nonconvex (as the maximization of a piecewise linear convex function) but we can write it as:

$$\alpha \leq \max\left(\hat{r}_1 + \max_{\delta \in \boldsymbol{\Delta}} \left(\delta_1 - \delta^\intercal \hat{\mathbf{x}}\right), \ldots, \hat{r}_I + \max_{\delta \in \boldsymbol{\Delta}} \left(\delta_I - \delta^\intercal \hat{\mathbf{x}}\right)\right) - \hat{\mathbf{r}}^\intercal \hat{\mathbf{x}}$$

We let $v_i^* = \max_{\delta \in \Delta} \left(\delta_i - \delta^\intercal \hat{\mathbf{x}}\right)$. A closed form solution for each of these values can be obtained by optimizing over the ellipsoid $\boldsymbol{\Delta}$ (parameterized by θ for this part of the argument):

$$v_i^* = \theta \sqrt{(\mathbf{e}_i - \hat{\mathbf{x}})^\intercal \mathbf{G}\mathbf{G}^\intercal (\mathbf{e}_i - \hat{\mathbf{x}})}$$

Since the columns of the nonsquare matrix $\mathbf{G}$ are orthonormal, we can extend them into an orthonormal square matrix $\bar{\mathbf{G}}$ to derive a bound on each v_i^*:

$$v_i^* \leq \theta \sqrt{(\mathbf{e}_i - \hat{\mathbf{x}})^\intercal \bar{\mathbf{G}}\bar{\mathbf{G}}^\intercal (\mathbf{e}_i - \hat{\mathbf{x}})} \leq \theta ||\mathbf{e}_i - \hat{\mathbf{x}}||_2$$

and use this to return to the bound on α:

$$\alpha \leq \max\left(\hat{r}_1 + \theta ||\mathbf{e}_1 - \hat{\mathbf{x}}||_2, \ldots, \hat{r}_I + \theta ||\mathbf{e}_I - \hat{\mathbf{x}}||_2\right) - \hat{\mathbf{r}}^\intercal \hat{\mathbf{x}}$$

Substituting in the relationship $\tan(\gamma) = \theta / ||\hat{\mathbf{r}}||$ leaves us with the result we intended to show:

$$\frac{\alpha}{||\hat{\mathbf{r}}||} \leq \max\left(\frac{\hat{r}_1}{||\hat{\mathbf{r}}||} + ||\mathbf{e}_1 - \hat{\mathbf{x}}||_2 \tan(\gamma), \ldots, \frac{\hat{r}_I}{||\hat{\mathbf{r}}||} + ||\mathbf{e}_I - \hat{\mathbf{x}}||_2 \tan(\gamma)\right) - \frac{\hat{\mathbf{r}}^\intercal \hat{\mathbf{x}}}{||\hat{\mathbf{r}}||}$$

□

Next we refine Lemma 1 to prove Proposition 3 for the specific case of the priority selection procedure.

Proof. In this selection procedure, the allocation is determined by the highest-ranking projected reward coefficient (this is the recipient type with the highest score), so $\hat{\mathbf{x}} = \mathbf{e}_1$.

Substituting into the bound obtained in Lemma 1 gives:

$$\begin{aligned}
\frac{z^* - z}{||\hat{\mathbf{r}}||} &\leq \max\left(\frac{\hat{r}_1}{||\hat{\mathbf{r}}||} + 0, \ldots, \frac{\hat{r}_I}{||\hat{\mathbf{r}}||} + ||\mathbf{e}_I - \mathbf{e}_1||_2 \tan(\gamma)\right) - \frac{\hat{r}_1}{||\hat{\mathbf{r}}||} \\
&= \max\left(0, \frac{\hat{r}_2 - \hat{r}_1}{||\hat{\mathbf{r}}||} + \sqrt{2}\tan(\gamma), \ldots, \frac{\hat{r}_I - \hat{r}_1}{||\hat{\mathbf{r}}||} + \sqrt{2}\tan(\gamma)\right) \\
&= \max\left(0, \frac{\hat{r}_2 - \hat{r}_1}{||\hat{\mathbf{r}}||} + \sqrt{2}\tan(\gamma)\right)
\end{aligned}$$

Finally, by using the relationship $||\hat{\mathbf{r}}|| = ||\mathbf{r}|| \cos(\gamma)$ we get:

$$\frac{z^* - z}{||\mathbf{r}||} \leq \max\left(0, \frac{\hat{r}_2 - \hat{r}_1}{||\hat{\mathbf{r}}||} + \sqrt{2}\sin(\gamma)\right)$$

□

D. Lookahead Optimality Bound

The proof in this section establishes the result for $J = 3$, but the same approach can easily be extended to any $J > 3$.

Proof. Let $(i_1, j_1), (i_2, j_2), (i_3, j_3)$ be the sequence of matches selected by the lookahead heuristic, and let $(i'_2, j_2), (i'_3, j_3)$ be the matches appearing in the upper bound used by the heuristic when it selects its first match.

Next, let $(i^*_1, j_1), (i^*_2, j_2), (i^*_3, j_3)$ be the matches in the optimal solution. The algorithm followed by the heuristic ensures that the following two inequalities hold:

1. $r_{i_2 j_2} + r_{i_3 j_3} \geq \max(r_{i'_2 j_2}, r_{i'_3 j_3})$.
2. $r_{i_1 j_1} + r_{i'_2 j_2} + r_{i'_3 j_3} \geq r_{i^*_1 j_1} + r_{i^*_2 j_2} + r_{i^*_3 j_3}$.

If the first inequality did not hold, then the heuristic would have selected (i'_2, j_2) or (i'_3, j_3) before (i_2, j_2). If the second did not hold, then it would have selected (i^*_1, j_1) before (i_1, j_1).

Piecing these observations together gives the ratio:

$$\frac{z}{z^*} = \frac{r_{i_1 j_1} + r_{i_2 j_2} + r_{i_3 j_3}}{r_{i^*_1 j_1} + r_{i^*_2 j_2} + r_{i^*_3 j_3}} \tag{34}$$

$$\geq \frac{r_{i_1 j_1} + \max(r_{i'_2 j_2}, r_{i'_3 j_3})}{r_{i_1 j_1} + r_{i'_2 j_2} + r_{i'_3 j_3}} \tag{35}$$

This ratio is at its minimum when $r_{i_1 j_1} = 0$, and once this restriction is enforced it becomes clear that the minimum value of the ratio is $1/2$. □

Note that, in the final ratio constructed in the proof, replacing $\max(r_{i'_2 j_2}, r_{i'_3 j_3})$ with $\max(r_{i'_2 j_2}, \dots, r_{i'_J j_J})$ in the numerator and $r_{i'_2 j_2} + r_{i'_3 j_3}$ with $r_{i'_2 j_2} + \cdots + r_{i'_J j_J}$ in the denominator leads to a minimum ratio of $1/(J-1)$. The remaining logic all holds for $J > 3$, and so does the result.

E. Margin Formulation with $\eta > 0$

In the general case when $\eta > 0$, Equation (24) can be formulated as a linear MIP rather than a linear program. Problem 6 provides the full formulation:

Problem 6 (General Margin Formulation)**.**

$$
\begin{aligned}
\max_{\mathbf{w},\, \mathbf{s},\, \mathbf{x} \in \{0,1\}^{I \times J},\, \gamma,\, z} \quad & \sum_{i=1}^{I} \sum_{j=1}^{J} r_{ij} x_{ij} + \eta z && && \text{(36a)} \\
\text{subject to} \quad & \mathbf{e}^\mathsf{T} \mathbf{w} = 1 && && \text{(36b)} \\
& s_{ij} = \mathbf{w}^\mathsf{T} \mathbf{f}_{ij} && \forall i \in [I], j \in [J] && \text{(36c)} \\
& s_{ij} - s_{i'j} \geq M(x_{ij} - 1) && \forall i \in [I],\ j \in [J],\ i' \neq i && \text{(36d)} \\
& \sum_{i=1}^{I} x_{ij} \leq 1 && \forall j \in [J] && \text{(36e)} \\
& z \leq \sum_{i=1}^{I} x_{ij} s_{ij} - \gamma_j && \forall j \in [J] && \text{(36f)} \\
& \gamma_j \geq s_{ij} - M x_{ij} && \forall j \in [J] && \text{(36g)}
\end{aligned}
$$

The $\sum_{i=1}^{I} x_{ij} s_{ij}$ term in Equation (36f) picks out the top score for resource type j, and the constraints in Equation (36g) are only switched on for recipient types which are not top-ranked. The product terms each involve a binary and continuous variable and can be linearized using standard techniques.

F. Fixed Point Uniqueness in the Lottery Mechanism

We numerically verified that Banach-Picard iteration using the operator $T(\mathbf{x})$ (that was defined in Section 6) converge to a unique fixed-point. The computation was conducted as follows: for each $I, J \in \{2, 3, 5, 10, 20, 50\}$ (representing the number of recipient and resource types, respectively), 1000 instances of data were randomly generated to satisfy the scarcity assumption. For each instance, 1000 random vectors in $\mathcal{X}$ were selected and used as starting points for the Banach-Picard iteration scheme.

In *all* instances and starting points, iterations converged to a unique fixed point. Figures 16 and 17 show the number of iterations and time required to find the fixed points:

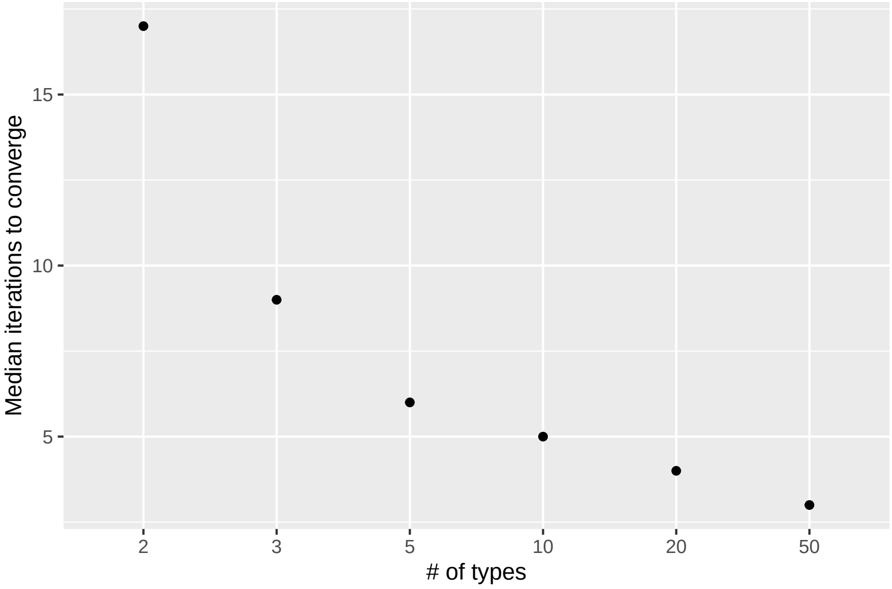

Figure 16: Median number of iterations (across data instances and starting points) required for convergence to a fixed point.

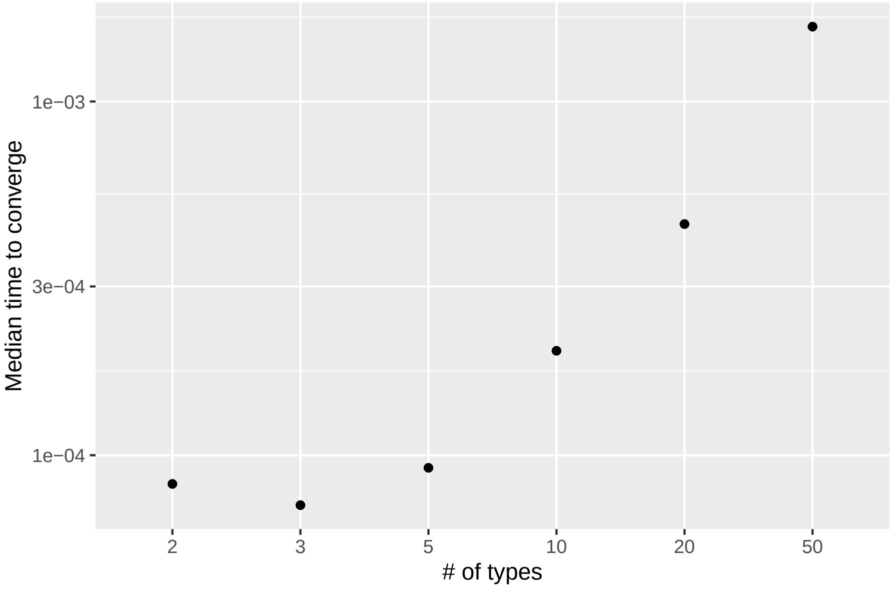

Figure 17: Median number of seconds (across data instances and starting points) required for convergence to a fixed point.

References

Afeche, P., Caldentey, R., & Gupta, V. (2019). On the optimal design of a bipartite matching queueing system. *Available at SSRN 3345302*.

Amaldi, E., & Kann, V. (1995). The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical computer science*, *147*(1-2), 181–210.

Ashlagi, I., & Shi, P. (2016). Optimal allocation without money: An engineering approach. *Management Science*, *62*(4), 1078–1097.

Been, V., O'Regan, K., Waldinger, D., & Center, N. F. (2018). Allocation of the limited subsidies for affordable housing. *New York Times*.

Bertsimas, D., Farias, V. F., & Trichakis, N. (2013). Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research*, *61*(1), 73–87.

Bodoh-Creed, A. L. (2020). Optimizing for distributional goals in school choice problems. *Management Science*, *66*(8), 3657–3676.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

David, I., & Yechiali, U. (1995). One-attribute sequential assignment match processes in discrete time. *Operations research*, *43*(5), 879–884.

Delos, V., & Teissandier, D. (2014). Minkowski sum of polytopes defined by their vertices. *arXiv preprint arXiv:1412.2564*.

Ding, Y., McCormick, S. T., & Nagarajan, M. (2021). A fluid model for one-sided bipartite matching queues with match-dependent rewards. *Operations Research*, *69*(4), 1256–1281.

Edwards, R. T. (1999). *Points for pain: waiting list priority scoring systems: May be the way forward, but we need to learn more about their effects.* British Medical Journal Publishing Group.

Emanuel, E. J., Persad, G., Upshur, R., Thome, B., Parker, M., Glickman, A., … Phillips, J. P. (2020). *Fair allocation of scarce medical resources in the time of covid-19* (Vol. 382) (No. 21). Mass Medical Soc.

Greely, H. (1977). Equality of allocation by lot, the. *Harv. CR-CLL Rev.*, *12*, 113.

Israni, A. K., Salkowski, N., Gustafson, S., Snyder, J. J., Friedewald, J. J., Formica, R. N., … others (2014). New national allocation policy for deceased donor kidneys in the united states and possible effect on patient outcomes. *Journal of the American Society* *of Nephrology*, *25*(8), 1842–1848.

Iyer, A. A., Hendriks, S., & Rid, A. (2020). Advantages of using lotteries to select participants for high-demand covid-19 treatment trials. *Ethics & Human Research*, *42*(4), 35–40.

Jansen, L. A., & Wall, S. (2021). Weighted lotteries and the allocation of scarce medications for covid-19. *Hastings Center Report*, *51*(1), 39–46.

Mehrotra, V., Ross, K., Ryder, G., & Zhou, Y.-P. (2012). Routing to manage resolution and waiting time in call centers with heterogeneous servers. *Manufacturing & service operations management*, *14*(1), 66–81.

NLIHC. (2021). *The gap — NLIHC.* https://reports.nlihc.org/gap. (Last accessed on 2021-8-23)

Persad, G., Wertheimer, A., & Emanuel, E. J. (2009). Principles for allocation of scarce medical interventions. *The Lancet*, *373*(9661), 423–431.

Righter, R. (1989). A resource allocation problem in a random environment. *Operations Research*, *37*(2), 329–338.

Saunders, B. (2009). A defence of weighted lotteries in life saving cases. *Ethical Theory and Moral Practice*, *12*(3), 279–290.

Shi, P. (2019). Optimal priority-based allocation mechanisms. *Available at SSRN 3425348*.

Sisselman, M. E., & Whitt, W. (2007). Value-based routing and preference-based routing in customer contact centers. *Production and Operations Management*, *16*(3), 277–291.

Sparrow, J. C. (1951). *History of personnel demobilization in the united states army* (No. 20). Office of the Chief of Military History, Department of the Army.

Su, X., & Zenios, S. (2004). Patient choice in kidney allocation: The role of the queueing discipline. *Manufacturing & Service Operations Management*, *6*(4), 280–301.

Su, X., & Zenios, S. A. (2006). Recipient choice can address the efficiency-equity trade-off in kidney transplantation: A mechanism design model. *Management science*, *52*(11), 1647–1660.

Supady, A., Curtis, J. R., Abrams, D., Lorusso, R., Bein, T., Boldt, J., … Brodie, D. (2021). Allocating scarce intensive care resources during the covid-19 pandemic: practical challenges to theoretical frameworks. *The Lancet Respiratory Medicine*.

Ten Cate, O. (2021). *Rationales for a lottery among the qualified to select medical trainees: Decades of dutch experience* (Vol. 13) (No. 5). The Accreditation Council for Graduate Medical Education.

Thakral, N. (2016). *The public-housing allocation problem* (Tech. Rep.). Technical report, Harvard University.

UNOS. (2021). *Transplant trends – UNOS.* https://unos.org/data/transplant-trends/. (Last accessed on 2021-8-23)

White, D. B., & Angus, D. C. (2020). A proposed lottery system to allocate scarce covid-19 medications: promoting fairness and generating knowledge. *Jama*, *324*(4), 329–330.

Young, H. P. (1994). *Equity: In theory and practice.* Princeton University Press. Retrieved from http://www.jstor.org/stable/j.ctv10crfx7

Zenios, S. A., Wein, L. M., & Chertow, G. M. (1999). Evidence-based organ allocation. *The American journal of medicine*, *107*(1), 52–61.