Scalable Robust Kidney Exchange

Duncan C McElfresh
Department of Mathematics
University of Maryland
College Park, MD 20742
dmcelfre@math.umd.edu

Hoda Bidkhori
Department of Industrial Engineering
University of Pittsburgh
Pittsburgh, PA 15260
bidkhori@pitt.edu

John P Dickerson
Department of Computer Science
University of Maryland
College Park, MD 20742
john@cs.umd.edu

Abstract

In barter exchanges, participants directly trade their endowed goods in a constrained economic setting without money. Transactions in barter exchanges are often facilitated via a central clearinghouse that must match participants even in the face of uncertainty—over participants, existence and quality of potential trades, and so on. Leveraging robust combinatorial optimization techniques, we address uncertainty in kidney exchange, a real-world barter market where patients swap (in)compatible paired donors. We provide two scalable robust methods to handle two distinct types of uncertainty in kidney exchange—over the *quality* and the *existence* of a potential match. The latter case directly addresses a weakness in all stochastic-optimization-based methods to the kidney exchange clearing problem, which all necessarily require explicit estimates of the probability of a transaction existing—a still-unsolved problem in this nascent market. We also propose a novel, scalable kidney exchange formulation that eliminates the need for an exponential-time constraint generation process in competing formulations, maintains provable optimality, and serves as a subsolver for our robust approach. For each type of uncertainty we demonstrate the benefits of robustness on real data from a large, fielded kidney exchange in the United States. We conclude by drawing parallels between robustness and notions of fairness in the kidney exchange setting.

1 Introduction

Real-world optimization problems face various types of uncertainty that impact both the quality and feasibility of candidate solutions. Uncertainty in combinatorial optimization is especially troublesome: if the *existence* of certain constraints or variables is uncertain, identifying a good—or even feasible—solution can be extremely difficult. Stochastic optimization approaches endeavor to maximize the *expected* objective value, under uncertainty. While sometimes successful, stochastic optimization relies heavily on a correct characterization of uncertainty; furthermore, stochastic approaches are often intractable—especially in combinatorial domains (Bertsimas *et al.* 2011a). A complementary approach is *robust optimization*, which protects against worst-case outcomes. Robust approaches can be less sensitive to the exact characterization of uncertainty, and are often far more tractable than stochastic approaches (Ben-Tal *et al.* 2009).

This paper addresses uncertainty in *kidney exchange*, a real-world barter market where patients with end-stage renal disease enter and trade their willing paired kidney donors (Rapaport 1986; Roth *et al.* 2004). Kidney exchange is a relatively new paradigm for organ allocation, but already accounts for over 10% of living kidney donations in the United States, and is growing in popularity worldwide (Biró *et al.* 2017). Modern exchanges also include *non-directed donors* (NDDs), who enter the market without a paired patient and donate their kidney without receiving one in return. Computationally, kidney exchange is a packing problem: solutions (matchings) consist of cyclic organ swaps and NDD-initiated donation chains in a directed compatibility graph, representing all participants and potential transactions. Each potential transplant is given a numerical weight by policymakers; the objective is to select cycles and chains that maximize overall matching weight. In general, this problem is NP-hard (Abraham *et al.* 2007; Biró *et al.* 2009); however, many efficient deterministic formulations exist that are fielded now and clear real exchanges (Abraham *et al.* 2007; Manlove and O'Malley 2015; Anderson *et al.* 2015; Dickerson *et al.* 2016; 2018).

Uncertainty in kidney exchange. Presently-fielded kidney exchange algorithms largely do not address uncertainty. Here, we consider two types of uncertainty in kidney exchange: over the *quality* of the transplant (weight uncertainty) and over the *existence* of potential transplants (existence uncertainty). Policymakers assign weights to potential transplants, which are (imperfect) estimates of transplant quality; weight uncertainty stems from both measurement uncertainty (e.g., medical compatibility and kidney quality) and uncertainty in the prioritization of some patients over others. Transplant existence is always uncertain: matched transplants "fail" before executing for a variety of reasons, severely impacting a planned kidney exchange. To address both cases, we propose *uncertainty sets* containing different realizations of the uncertain parameters. We then develop a scalable robust optimization approach, and demonstrate its success on data from a large fielded kidney exchange.

Robust optimization is a popular approach to optimization under uncertainty, with applications in reinforcement learning (Petrik and Subramanian 2014), regression (Xu *et al.* 2009), classification (Chen *et al.* 2017), and network optimization (Mevissen *et al.* 2013). Motivated by real-world constraints, we apply robust optimization to kidney exchange—a graph-based market clearing or resource allocation problem.

Our Contributions. To our knowledge, weight uncertainty has not been addressed in the kidney exchange literature. Our approach is similar to that of Bertsimas and Sim (2004) and Poss (2014), and uses some of their results. Several approaches have been proposed for existence uncertainty, primarily based on stochastic optimization (Dickerson *et al.* 2016; Anderson *et al.* 2015; Dickerson *et al.* 2018) or hierarchical optimization (Manlove and O'Malley 2015). The primary disadvantage of these approaches—in addition to tractability—is their reliance on, and sensitivity to, the explicit estimation of the probability of each particular potential transplant. This probability is extremely difficult to determine (Dickerson *et al.* 2018; Glorie 2012), and prevents the translation of those methods into practice. Our approach uses a simpler notion of edge existence uncertainty—an upper-bound on the number of non-existent edges—which is easier to interpret and estimate. Glorie (2014) proposed a related robust formulation that is exponentially larger than ours, and is intractable for realistically-sized exchanges.

In addition, we introduce a new scalable formulation for kidney exchange that combines concepts from two state-of-the-art formulations (Anderson *et al.* 2015; Dickerson *et al.* 2016), handles long or uncapped NDD-initiated chains without requiring expensive constraint generation, and ties into a developed literature on fairness in kidney exchange—thus addressing use cases that are becoming more common in fielded exchanges (Anderson *et al.* 2015).

2 Preliminaries

Model for kidney exchange. A kidney exchange can be represented formally by a directed compatibility graph $G = (V, E)$. Here, vertices $v \in V$ represent participants in the exchange, and are partitioned as $V = P \cup N$ into P, the set of all patient-donor pairs, and N, the set of all NDDs (Roth *et al.* 2004; 2005; Abraham *et al.* 2007). Each potential transplant from a donor at vertex u to a patient at vertex v is represented by a directed edge $e = (u, v) \in E$, which has an associated weight $w_e \in \mathbf{w}$; weights are set by policymakers, and reflect both the medical utility of the transplant, as well as ethical considerations (e.g., prioritizing patients by waiting time, age, and so on). Cycles in G correspond to cyclic trades between multiple patient-donor pairs in P; chains, correspond to donations that begin with an NDD in N and continue through multiple patient-donor pairs in P. The kidney exchange *clearing problem* is to select a feasible set of transplants (edges in E) that maximize overall weight. Let $\mathcal{M}$ be the set of all feasible *matchings* (i.e., solutions) to a kidney exchange problem; the general formulation of this problem is $\max_{\mathbf{x}\in\mathcal{M}} \mathbf{x}\cdot\mathbf{w}$, where binary decision variables $\mathbf{x}$ represent edges, or cycles and chains. This problem is NP- and APX-hard (Abraham *et al.* 2007; Biró *et al.* 2009).

Robust optimization. Robust optimization is a common approach to optimization under uncertainty, which is often more tractable and requires less accurate uncertainty information than other approaches (Bertsimas *et al.* 2011a). This approach begins by defining an *uncertainty set* $\mathcal{U}$ for the uncertain optimization parameter; $\mathcal{U}$ contains different *realizations* of this parameter. Consider the example of edge weight uncertainty: we might design an edge weight uncertainty set $\mathcal{U}_w$ that contains the *realized* (i.e. "true") edge weights $\hat{\mathbf{w}}$ with high probability, $P(\hat{\mathbf{w}} \in \mathcal{U}_w) \geq 1 - \epsilon$, for $0 < \epsilon \ll 1$. The parameter ϵ is referred to as the *protection level*, and is often used to control the number of realizations in $\mathcal{U}$.

After designing $\mathcal{U}$, the robust approach finds the best solution, assuming the *worst-case* realization within $\mathcal{U}$. For kidney exchange (a maximization problem), this corresponds to a *minimization* over $\mathcal{U}$; for example, Problem (1) is the robust formulation with uncertain edge weights.

$$\max_{\mathbf{x}\in\mathcal{M}} \min_{\hat{\mathbf{w}}\in\mathcal{U}} \quad \mathbf{x}\cdot\hat{\mathbf{w}} \tag{1}$$

The robustness of this approach depends on the proportion of possible realizations contained in $\mathcal{U}$. If $\mathcal{U}$ contains all possible realizations, the approach may be too conservative; if $\mathcal{U}$ only contains one possible realization of $\hat{\mathbf{w}}$, the solution may be too myopic. The number of realizations in $\mathcal{U}$ is often controlled by a parameter: either an *uncertainty budget* Γ, or the protection level ϵ. Next we introduce the first type of uncertainty considered in this paper: edge weight uncertainty.

3 Optimization in the Presence of Edge Weight Uncertainty

Edge weights in kidney exchange represent the medical and social utility gained by a *single* kidney transplant. Weights are determined by policymakers, and are subject to several types of uncertainty.[1] Part of this uncertainty is due to insufficient knowledge of the future: a patient or donor's health may change, raising or lowering the "true" weight of their transplant edges. Another type of uncertainty stems from disagreement between policymakers regarding the social utility of a transplant. For example, some policymakers might prioritize young patients over older patients; other policymakers might prioritize the sickest patients above all healthier patients. Policymakers aggregate these value judgments to assign a single weight to each transplant edge, but this aggregation is a contentious and imperfect process (although recent work from the AI community has begun to address this using techniques from computational social choice and machine learning (Freedman *et al.* 2018; Noothigattu *et al.* 2018)). Still, there is no way to measure the "true" social utility of a transplant, and therefore this uncertainty is not easily measured.

Interval weight uncertainty. It is beyond the scope of this work to characterize these sources of uncertainty. We simply assume that the *nominal* edge weights $\mathbf{w}$, provided by policymakers, are an uncertain estimate of the *realized* edge weights $\hat{\mathbf{w}}$, i.e., the "true" value of each transplant. Next, we formalize edge weight uncertainty and our robust approach. This section focuses on edge weights, so we write our for-

[1] The process used to set weights by the UNOS US-wide kidney exchange is published publicly (UNOS 2015).

mulations with decision variables $x_e \in \mathbf{x}$ corresponding to individual edges.

We assume that realized edge weights $\hat{\mathbf{w}}$ are random variables with a partially known symmetric distribution, centered about the nominal weights $\mathbf{w}$. This assumption implies that $E[\hat{\mathbf{w}}] = \mathbf{w}$, thus a non-robust approach that maximizes $\mathbf{w}$ is equivalent to a stochastic optimization approach that maximizes *expected* edge weight. We refer to this edge uncertainty model as *interval uncertainty*.

Definition 1 (Interval Edge Weight Uncertainty)**.** Let $\hat{w}_e$ be the realized weight of edge e, with nominal weight w_e, and maximum discount $0 \leq d_e \leq w_e$. Let $\hat{w}_e \equiv w_e + d_e \alpha_e$, where α_e is the *fractional deviation* of edge e. Both α_e and $\hat{w}_e$ are continuous random variables, symmetrically distributed on $[-1, 1]$ and $[w_e - d_e, w_e + d_e]$ respectively.

Each discount factor d_e should reflect the level of uncertainty in w_e. If w_e is known exactly, then $d_e = 0$; if w_e is very uncertain, then we might set $d_e = w_e$, or higher.

To vary the degree of uncertainty, we use an *uncertainty budget* Γ, which limits the total deviation from nominal edge weights. With our uncertainty model, it is natural to let Γ limit the total fractional deviation of each edge weight—i.e., sum of all α_e. This uncertainty set $\mathcal{U}^I_\Gamma$ is defined as:

$$\mathcal{U}^I_\Gamma = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = w_e + d_e \alpha_e, |\alpha_e| \leq 1, \sum_{e \in E} |\alpha_e| \leq \Gamma \right\}$$

For example if $\Gamma = 3$, there may be three edges with $|\alpha_e| = 1$, or one edge with $|\alpha_e| = 1$ and four edges with $|\alpha_e| = 1/2$, and so on.

Choosing an appropriate Γ is not straightforward. Matchings often use only a small fraction of the decision variables (e.g., transplant edges), and it is difficult to predict the size of the optimal matching. Intuitively, Γ should reflect the size of the final matching: for example if we assume that half of any matching's edges will be discounted, then we should set $\Gamma \simeq |\mathbf{x}|/2$. Generalizing this concept, we define a *variable*-budget uncertainty set $\mathcal{U}^I_\gamma$, with budget function $\gamma(|\mathbf{x}|)$.

$$\mathcal{U}^I_\gamma = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = w_e + d_e \alpha_e, |\alpha_e| \leq 1, \sum_{e \in E} |\alpha_e| \leq \gamma(|\mathbf{x}|) \right\}$$

Next, to define γ, we relate it to a much more intuitive parameter: the protection level ϵ.

3.1 Uncertainty Budget γ and Protection Level ϵ

The protection level ϵ mediates between a completely conservative approach, and the non-robust approach: as $\epsilon \to 0$ the approach becomes more conservative, and $\epsilon = 1$ corresponds to a non-robust approach. In this section we relate γ to ϵ, beginning with the following Theorem 1.

Theorem 1 (Adapted from Theorem 3 of (Bertsimas and Sim 2004))**.** *For a matching $\mathbf{x} \in \mathcal{M}$ with $|\mathbf{x}|$ edges, and uncertainty set $\mathcal{U}^I_\Gamma$, the probability that $\mathcal{U}^I_\Gamma$ contains the realized edge weights for $\mathbf{x}$ is bounded below by*

$$P(\hat{\mathbf{w}} \in \mathcal{U}^I_\Gamma) \geq 1 - B(|\mathbf{x}|, \Gamma),$$

with

$$B(|\mathbf{x}|, \Gamma) = \frac{1}{2^{|\mathbf{x}|}} \left((1 - \mu) \binom{|\mathbf{x}|}{\lfloor \eta \rfloor} + \sum_{l = \lfloor \eta \rfloor + 1}^{|\mathbf{x}|} \binom{|\mathbf{x}|}{l} \right),$$

with $\eta = (\Gamma + |\mathbf{x}|)/2$ and $\mu = \eta - \lfloor \eta \rfloor$.

That is, for some ϵ, if Γ is chosen such that $\epsilon = B(|\mathbf{x}|, \Gamma)$, then the inequality $P(\hat{\mathbf{w}} \in \mathcal{U}^I_\Gamma) \geq 1 - \epsilon$ holds by Theorem 1. Next, we use this result to define a variable uncertainty budget function γ, using the intuitive definition introduced by Poss (2014): for matching $\mathbf{x} \in \mathcal{M}$ and protection level ϵ, we find the minimum Γ such that $B(|\mathbf{x}|, \Gamma) \leq \epsilon$. If this is not possible (i.e., the matching is too small, or ϵ is too small), then $\gamma = |\mathbf{x}|$. This budget function is defined as:

$$\beta(|\mathbf{x}|) = \begin{cases} |\mathbf{x}| & \text{if } \min\limits_{\Gamma > 0} \{\Gamma \mid B(|\mathbf{x}|, \Gamma) \leq \epsilon\} \text{ is infeasible,} \\ \min\limits_{\Gamma > 0} \{\Gamma \mid B(|\mathbf{x}|, \Gamma) \leq \epsilon\} & \text{otherwise.} \end{cases}$$

It may not be clear how to solve the edge weight robust problem with this variable uncertainty budget. We use the approach of Poss (2014), which solves the variable-budget robust problem by solving several instances of the *constant*-budget robust problem; details of this approach can be found in Appendix A.4[2]. Thus, to solve the variable-budget robust problem we first solve the constant-budget robust problem.

3.2 Constant-Budget Edge Weight Robust Approach

We now describe our approach to the constant-budget edge weight robust problem; a full discussion and derivation can be found in Appendix A. We need to solve Problem (1) with edge weight uncertainty set $\mathcal{U}^I_\Gamma$. This requires a minimization of the objective, over $\hat{\mathbf{w}} \in \mathcal{U}^I_\Gamma$, followed by a maximization over matchings in $\mathcal{M}$.

First we *directly minimize* the objective of Problem (1) over $\mathcal{U}^I_\Gamma$. That is, for any matching $\mathbf{x} \in \mathcal{M}$, we find the minimum objective value for any realized edge weights in $\mathcal{U}^I_\Gamma$, denoted by $Z(\mathbf{x})$:

$$Z(\mathbf{x}) = \min_{\hat{\mathbf{w}} \in \mathcal{U}^I_\Gamma} \mathbf{x} \cdot \hat{\mathbf{w}} \tag{2}$$

Thus, solving the robust problem corresponds to maximizing $Z(\mathbf{x})$ over all feasible matchings. Our approach to doing so is as follows. First, we linearize $Z(\mathbf{x})$ using several new variables and constraints; we then add these to an existing kidney exchange formulation (Dickerson *et al.* 2016). The complete linear formulations of $Z(\mathbf{x})$ and Problem (1) are given in Appendix A.2, but are omitted here for space. Our robust formulation is scalable—it has a polynomial count of variables and constraints, regardless of finite chain cap; on realistic exchanges it takes only a few seconds to solve. We demonstrate our method's impact on match composition in Section 5, and show how it effectively controls for the impact of robustness using protection level ϵ.

[2] All appendices are included in the full version of this paper, available on arXiv: `https://arxiv.org/abs/1811.03532`

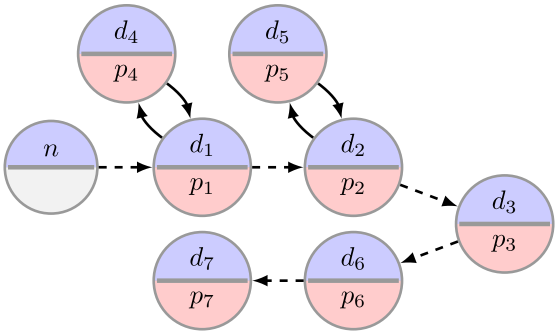

Figure 1: Sample exchange graph with a 5-chain and two 2-cycles. The NDD is denoted by n, and each patient (and her associated donor) is denoted by p_i (d_i). A maximum-cardinality matching algorithm would select the 5-chain, denoted with dashed edges; however, the smaller matching consisting of two disjoint 2-cycles, shown with solid edges, may be more robust to edge failure.

4 Optimization in the Presence of Edge Existence Uncertainty

In this section we consider *edge existence uncertainty*, where an algorithmic match must be chosen before the full realization of edges is revealed. Algorithmically-matched transplants in a kidney exchange can fail before transplantation for a variety of reasons: a patient may become too ill to undergo transplantation, or pre-transplantation testing may reveal that a patient is incompatible with her planned donor kidney. Furthermore, some edges are more likely to fail than others (e.g., edges into particularly sick patients). Edge failure significantly impacts fielded exchanges–with failure rates above 50% in many cases (Dickerson *et al.* 2018; Anderson *et al.* 2015; Ashlagi *et al.* 2013).

For illustration, consider the simple exchange in Figure 1 with two potential matchings: single 5-chain initiated by the NDD, or two 2-cycles (with pairs $\{1, 4\}$ and $\{2, 5\}$). The 5-chain matches the most patient, but is less robust to edge failures. Consider the *worst-case* outcome for each matching, when 1 edge is *guaranteed* to fail: with the 5-chain, in the worst-case the *first* edge fails, causing the entire chain to fail; with the 2-cycles, a single edge failure only causes a *single* cycle to fail, leaving the other cycle complete. With this notion of edge existence uncertainty (which we define later), the 2-cycles are more robust than the 5-chain.

Managing edge failure in kidney exchange has been addressed in the AI and optimization literature in application-specific (Manlove and O'Malley 2015; Chen *et al.* 2012) or stochastic-optimization-based (Dickerson *et al.* 2018; 2016; Anderson *et al.* 2015; Klimentova *et al.* 2016) ways. These *failure-aware* approaches associate with each edge a predetermined failure probability p_e; these probabilities are used to then maximize *expected* matching score, possibly subject to some recourse actions. This stochastic approach is tractable when p_e is identical for each edge. Our work addresses two major drawbacks of the failure-aware approach. First, when each edge has a unique p_e, those models require enumerating every cycle and chain, which is intractable for large graphs or long chains. Second, the failure-aware approach is very sensitive to p_e (as discussed in, e.g., §4.4 of Dickerson *et al.* (2018)). In practice, precise values of p_e are not known, thus the failure-aware approach can easily produce unreliable results. We use a simpler notion of edge existence uncertainty, which assumes that in any matching, the number of edges is *bounded* by a constant (Γ). This parameter is intuitive and simple to estimate from past exchanges.

To our knowledge, ours is the first *scalable* robust optimization approach to edge existence uncertainty in kidney exchange. Glorie (2014) develops several elegant robust methods for edge existence uncertainty, but requires that all cycles and chains are found during pre-processing and stored in memory. The number of chains grows exponentially in both the number of edges and the maximum chain length; thus, these approaches are intractable for exchanges involving more than a few dozen patient-donor pairs and NDDs.

Edge existence uncertainty. Here we briefly describe our robust approach to edge existence uncertainty; a full discussion and derivation can be found in Appendix B. For ease of exposition, in this section, decision variables $x_c \in \mathbf{x}$ correspond to cycles and chains rather than edges. We use the following model of edge existence uncertainty.

Definition 2 (Γ-Failures Edge Existence Uncertainty)**.** Up to Γ edges may fail in any matching. After failures occur, the realized exchange graph is $\hat{G} = (V, \hat{E})$, such that edges $\hat{E} \subseteq E$ succeed and remain in existence, while all other edges fail and do not exist.

With this notion of uncertainty, without regard to computational or memory constraints, a stochastic-optimization-based approach could identify the best matching over all possible realizations $\hat{G}$ (Anderson *et al.* 2015). This is clearly intractable, as the number of realized graphs is exponential in $|E|$. Instead, we take a robust optimization approach by maximizing the worst-case (minimum) matching score over a set of realizable graphs $\hat{G}$ in an uncertainty set $\mathcal{U}$. Like the stochastic approach, the robust approach considers a huge number of realizations $\hat{G}$; however the robust approach is far more tractable, as it need only find the worst-case realization and need not represent all realizable graphs explicitly.

Uncertainty set. Let $F \subseteq E$ be the subset of failed edges for a realized graph $\hat{G}$; thus, $\hat{E} = E \setminus F$ is the set of realized edges. Equation (3) defines uncertainty set $\mathcal{U}_\Gamma^{ex}$ in this way: up to Γ edges may fail (i.e., $|F| \leq \Gamma$).

$$\mathcal{U}_\Gamma^{ex} = \left\{ \hat{G} = (V, \hat{E}) \mid \hat{E} = E \setminus F, |F| \leq \Gamma \right\} \tag{3}$$

In kidney exchange, one edge failure can cause other edge failures: if one cycle edge fails, all edges in the cycle also fail; edge failure in a chain causes all *subsequent* chain edges to also fail. This leads to a notion of weight uncertainty for cycles and chains, where the realized weight of a cycle or chain $\hat{w}_c$ may be smaller than nominal weight w_c. Let α_c be a discount parameter for cycle or chain c, such that $\hat{w}_c = w_c(1 - \alpha_c)$. For example, if any edge fails in cycle

c, then the entire cycle fails and $\alpha_c = 0$. We define the cycle/chain weight uncertainty set $\mathcal{U}^w_\Gamma$ in this way:

$$\mathcal{U}^w_\Gamma = \left\{ \hat{\mathbf{w}}_\mathbf{c} \mid \hat{w}_c = w_c(1-\alpha_c), \alpha_c \in [0,1], \sum_{c \in X} \alpha_i \leq \Gamma \right\}$$

This uncertainty set is less intuitive than $\mathcal{U}^{ex}_\Gamma$, but more suited to the robust approach. In Appendix B we show that $\mathcal{U}^w_\Gamma$ and $\mathcal{U}^{ex}_\Gamma$ are equivalent for integer Γ, and thus can be used for our robust approach.

4.1 Robust Optimization Approach

In this section we briefly describe our robust approach; for a full discussion and derivation, please see Appendix B. Our robust formulation for uncertainty set $\mathcal{U}^w_\Gamma$ follows a similar approach to Section 3. First, we directly minimize the kidney exchange objective over $\mathcal{U}^w_\Gamma$, for some feasible solution $\mathbf{x} \in \mathcal{M}$. We express this minimization as a function $Z(\mathbf{x})$: in effect, $Z(\mathbf{x})$ discounts the Γ largest-weight cycles and chains. We then linearize $Z(\mathbf{x})$ using several variables and constraints—this requires a formulation with variables tracking individual total chain weights—which is not possible in any existing compact kidney exchange formulations. For this purpose, we introduce a new kidney exchange formulation.

The PI-TSP formulation. We propose the position-indexed TSP formulation (PI-TSP); for details, please see Appendix B. Our formulation combines innovations from the two leading kidney exchange clearing approaches: PICEF (Dickerson *et al.* 2016) and PC-TSP (Anderson *et al.* 2015). PICEF introduced an indexing schema that enables a more compact formulation in the context of long chains; our formulation builds on this to allow tracking of individual chain weights, a necessity that PICEF could not do. PC-TSP builds on techniques from the prize-collecting travelling salesperson problem (Balas 1989) to provide a tight linear programming relaxation; in general, the PC-TSP formulation has exponentially many constraints and thus requires constraint generation to solve. Our formulation uses an efficient version of position indexing that also requires only $O(|E|)+O(|V|\cdot|N|)$ constraints. Unlike PICEF, our formulation does not grow with the chain cap L: PICEF uses $O(|V|^3)$ variables (when $L \to |V|$); for large graphs, the PICEF model becomes too large to fit into memory (Dickerson *et al.* 2016). Our formulation uses a fixed number of variables—$O(|V|^2)$—for any chain cap, alleviating this memory problem. This is especially relevant to existing exchanges, as long chains can significantly increase efficiency in kidney exchange (Ashlagi *et al.* 2012). PI-TSP uses the following parameters:

- a kidney exchange graph G with cycles C
- L: chain cap (maximum number of edges used in a chain),
- w_e: edge weights for each edge $e \in E$,
- w^C_c: cycle weights for each cycle $c \in C$,

and the following decision variables:

- $p_{\{e,v\}} \geq 1$: the position of edge e or vertex v in any chain,
- $\hat{p}_e \geq 0$: equal to p_e if e is used in a chain, and 0 otherwise,
- $z_c \in \{0,1\}$: 1 if cycle c is used in the matching,
- $y_e \in \{0,1\}$: 1 if edge e is used in a chain, and 0 otherwise,
- $y^n_e \in \{0,1\}$: 1 if edge e is used in a chain starting with NDD n, and 0 otherwise,
- w^N_n: total weight of the chain starting with NDD n,
- f^i_v and f^o_v: chain flow into and out of vertex $v \in P$,
- $f^{i,n}_v$ and $f^{i,n}_v$: chain flow into and out of vertex $v \in P$, from the chain starting with NDD $n \in N$.

The PI-TSP formulation with chain cap L is given in Problem 4. We use the notation $\delta^-(v)$ for the set of edges into vertex v and $\delta^+(v)$ for the set of edges out of v.

$$\max \quad \sum_{n \in N} w^N_n + \sum_{c \in C} w^C_c z_c \tag{4a}$$

$$\text{s.t.} \quad \sum_{e \in E} w_e y^n_e = w^N_n \qquad n \in N \tag{4b}$$

$$\sum_{n \in N} y^n_e = y_e \qquad e \in E \tag{4c}$$

$$\sum_{e \in \delta^-(v)} y_e = f^i_v \qquad v \in V \tag{4d}$$

$$\sum_{e \in \delta^+(v)} y_e = f^o_v \qquad v \in V \tag{4e}$$

$$\sum_{e \in \delta^-(v)} y^n_e = f^{i,n}_v \qquad v \in V, n \in N \tag{4f}$$

$$\sum_{e \in \delta^+(v)} y^n_e = f^{o,n}_v \qquad v \in V, n \in N \tag{4g}$$

$$f^o_v + \sum_{c \in C : v \in c} z_c \leq \quad f^i_v + \sum_{c \in C : v \in c} z_c \leq 1 \qquad v \in P \tag{4h}$$

$$f^o_v \leq 1 \qquad v \in N \tag{4i}$$

$$p_e = 1 \qquad e \in \delta^+(N) \tag{4j}$$

$$\hat{p}_e = p_e y_e \qquad e \in E \tag{4k}$$

$$p_v = \sum_{e \in \delta^-(v)} \hat{p}_e \qquad v \in P \tag{4l}$$

$$p_e = p_v + 1 \qquad v \in P, e \in \delta^+(v) \tag{4m}$$

$$\sum_{e \in E} y^n_e \leq L \qquad n \in N \tag{4n}$$

$$f^{o,n}_v \leq f^{i,v} \leq 1 \qquad v \in V, n \in N \tag{4o}$$

$$y_e \in \{0,1\} \qquad e \in E \tag{4p}$$

$$z_c \in \{0,1\} \qquad c \in C \tag{4q}$$

$$y^n_e \in \{0,1\} \qquad e \in E, n \in N \tag{4r}$$

The ability to express individual chain weights as decision variables has applications beyond robustness. For particularly valuable NDDs (such as those with so-called "universal donor" blood-type O), exchanges may enforce a *minimum* chain length or chain weight, to ensure that these rare NDDs are not "used up" on short chains; such a policy was formerly used by the United Network for Organ Sharing (Dickerson *et al.* 2012), using a much less scalable form of optimization—that also does not consider uncertainty—than our approach (Abraham *et al.* 2007). Such a policy can be implemented efficiently with PI-TSP, inefficiently with PC-TSP, and not with PICEF, where decision variables do not indicate from which NDD a chain originated. In Appendix B we show–using real kidney exchange data–that PI-TSP can enforce a minimum chain length, and that this restriction has *almost no* impact on overall matching score.

5 Experimental Results

In this section, we compare each robust formulation against the leading non-robust formulation, PICEF (Dickerson *et* *al.* 2016), with varying levels of uncertainty. These experiments use real exchange graphs collected from the United Network for Organ Sharing (UNOS)—a large US-wide kidney exchange with over 160 participating transplant centers—between 2010 and 2016, as well simulated exchanges generated from known patient statistics using the standard method (Dickerson *et al.* 2018).[3]

For each exchange, we calculate the optimal non-robust matching M_{OPT} (with total score $|M_{\text{OPT}}|$), and the robust matching M_{R}, for varying uncertainty budgets. We then draw many *realizations* of the exchange graph, based on the uncertainty model, and calculate the realized scores of the robust matching $|M_{\text{R}}|$ and non-robust matching $|M_{\text{NR}}|$. We are primarily interested in the fractional difference from $|M_{\text{OPT}}|$, calculated as $\Delta OPT\left(M_{\{\text{R,NR}\}}\right) = \left(|M_{\text{OPT}}| - |M_{\{\text{R,NR}\}}|\right)/|M_{\text{OPT}}|$.

We calculate $\Delta OPT\left(M_{\text{R}}\right)$ and $\Delta OPT\left(M_{\text{NR}}\right)$ for $N = 400$ realizations, and compare the robust and non-robust approaches. In rare cases the optimal matching is empty (i.e., there is no solution, or the uncertainty budget exceeds the matching size), we exclude these exchanges from the results.

Edge Weight Uncertainty We begin by exploring the impact on match utility of robust approaches to managing edge weight uncertainty. Here, each edge is randomly labeled as *probabilistic* (P) or *deterministic* (D). P edges receive weight 0 or 1 with probability 0.5, while D edges always receive weight 0.5; thus, expected edge weight is always 0.5. The non-robust approach maximizes *expected* edge weight, making this a kind of stochastic approach. The robust approach considers the discount value (0 or 0.5) of each edge, and avoids edges with a positive discount value. To vary the level of uncertainty, we vary the fraction of P edges (α). Each realization is drawn by assigning the P edges to have weight either 0 or 1.

We compute M_{R} for protection levels $\epsilon \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5\}$, and then calculate both $\Delta OPT\left(M_{\text{R}}\right)$ and $\Delta OPT\left(M_{\text{NR}}\right)$. Figure 2 shows ΔOPT on realistic 64-vertex simulated graphs (left) and larger (typically 150–300-vertex) real UNOS graphs (right); these figures show results for each protection level ϵ and for various α. Note that M_{NR} does not depend on ϵ, and thus the non-robust results are shown as (constant) dashed lines.

The robust approach guarantees a better worst-case (minimum) ΔOPT, but results in a lower median ΔOPT. The protection level ϵ controls the robustness of our approach; smaller ϵ protects against more uncertain outcomes, but at greater cost to nominal behavior. As $\epsilon \to 1$, the robust approach protects against fewer bad outcomes, and approaches the behavior of non-robust.

Edge Existence Uncertainty We now address edge existence uncertainty, and compare the robust and non-robust approaches with Γ edge failures, for $\Gamma \in \{1, 2, 3, 4, 5\}$. Each Γ corresponds to a different notion of uncertainty, such that exactly Γ edges fail.[4] For each Γ, we calculate M_{R}, and draw $N = 400$ realizations by failing Γ edges in the matching.

We calculate ΔOPT for each realization, and compare these results for the robust and non-robust matchings. With edge existence uncertainty, the worst-case outcome is almost always an empty matching ($\Delta OPT\left(\right) = -1$). Thus, rather than compare the worst-case ΔOPT, we compare the *distribution* of ΔOPT for each approach: we treat ΔOPT as a random variable, and use three simple statistical tests to demonstrate that—as expected—the robust approach produces more conservative and predictable results.

First, we use the Wilcoxon signed-rank test to determine that the robust and non-robust approaches produce a different distribution of ΔOPT. For each Γ, this test produces p-values well below 10^{-3}, indicating that the distributions of ΔOPT *are different* for the robust and non-robust approach. Second, for all exchanges and all Γ, the mean ΔOPT is typically 1% *higher*, and the standard deviation 1–2% *lower* for the robust approach. That is, the robust approach more consistently produces higher-weight solutions.

Third, we visualize the difference between these distributions using their histograms. Figure 3 shows the bin-wise difference between the histograms of ΔOPT (robust minus non-robust), with mean ΔOPT for non-robust shown as a dotted line. In these plots, the height of the bars indicate the change in probability density due to robustness. On all plots, the bars are *negative* for high and low values of ΔOPT, meaning that the robust matching is *less likely* to have an abnormally high or low ΔOPT. The bars are *positive* when ΔOPT is near its mean non-robust value—meaning that the robust matching is *more likely* to have a ΔOPT near the mean non-robust value. This is exactly the desired behavior: robustness produces more predictable and less varied results. In this application robustness exceeds expectations: the robust approach achieves a lower variance, *and* slightly improves nominal performance.

6 Robustness as Fairness

Balancing efficiency and fairness is a classic economic problem; recently, a body of literature covering fairness in kidney exchange has developed in the AI/Economics (Dickerson *et al.* 2014; McElfresh and Dickerson 2018; Ashlagi *et al.* 2013; Ding *et al.* 2018) and medical ethics (Gentry *et al.* 2005) communities; Appendix C presents a more thorough discussion. Though seemingly unrelated, fairness and robustness share a key characteristic: the balance between two competing properties. Fairness rules in kidney exchange often mediate between a fair and efficient outcome, using a parameter to set the balance. Similarly, robustness mediates between a good nominal outcome with the worst-case outcome, using an uncertainty budget or protection level to set that balance.

In kidney exchange, fairness often refers to the prioritization of pediatric or disadvantaged (*highly-sensitized*) patients, who are unlikely to find a compatible donor. In the

[3] All experiments were implemented in Python and used Gurobi (Gurobi Optimization, Inc. 2018), a state-of-the-art industrial combinatorial optimization toolkit, as a sub-solver. Our code is available on GitHub: `https://github.com/duncanmcelfresh/RobustKidneyExchange`.

[4] This is slightly more conservative than the notion of uncertainty introduced previously; in Section 4, *up to* Γ edges may fail, while in the experiments *exactly* Γ edges fail.

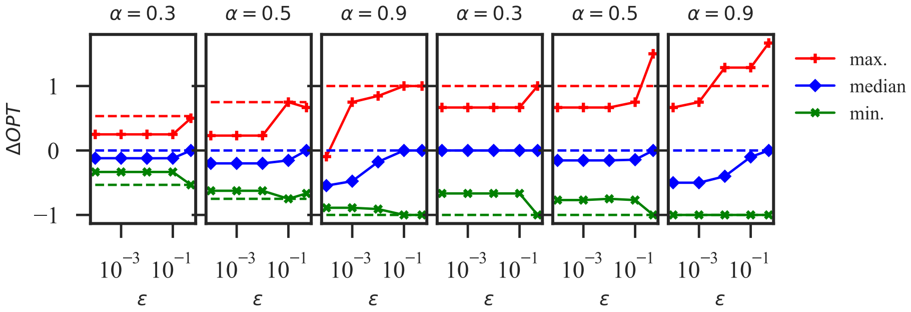

Figure 2: ΔOPT for non-robust (dashed lines) and edge weight robust (solid lines) matchings, for 64-vertex simulated exchange graphs (3 left plots) and real UNOS exchanges (3 right plots).

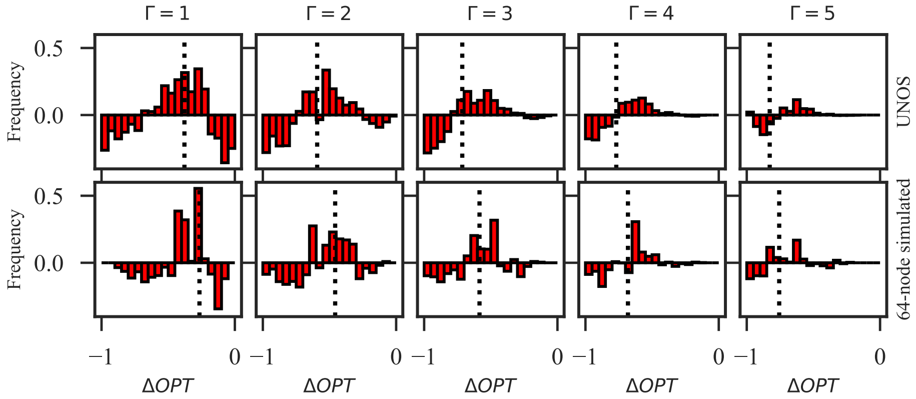

Figure 3: Difference between the robust and non-robust histograms of ΔOPT (robust minus non-robust) for real UNOS (top) and simulated exchanges (bottom), for various Γ. Dotted line: mean ΔOPT for non-robust.

weighted fairness approach, edges that represent transplants to prioritized patients receive additional edge weight, making them more likely to be matched by standard algorithms; versions of this prioritization scheme are used by most exchanges, including UNOS. To generalize weighted fairness, let each edge have a *priority weight* $\hat{w}_e \in [0, \infty)$, equal to the nominal weight w_e multiplied by a factor $(1 + \alpha_e)$, with $\alpha_e \in [-1, \infty)$. For example, we might set $\alpha_e > 0$ for all edges *into* prioritized patients; this will help prioritized patients, but will likely lower overall efficiency (a tradeoff often described as the *price of fairness* (Caragiannis *et al.* 2009; Bertsimas *et al.* 2011b; Dickerson *et al.* 2014; McElfresh and Dickerson 2018)).

To balance fairness with efficiency, policymakers limit the degree of prioritization. Let $\mathcal{P}_\Gamma$ be a *budgeted prioritization set*, which bounds the sum of absolute differences between each w_e and $\hat{w}_e$; this prioritization set is given as:

$$\mathcal{P}_\Gamma = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = w_e(1 + \alpha_e), \alpha_e \geq -1, \sum_{e \in E} \alpha_e w_e \leq \Gamma \right\}$$

As with edge weight uncertainty, the budget Γ balances between fairness and efficiency. If Γ is large, the algorithm might sacrifice matching size in order to match prioritized patients—but the maximum amount of efficiency sacrificed will be predictable, given Γ, which is attractive to policymakers. In Appendix C we further develop this concept, propose fairness rules that use $\mathcal{P}_\Gamma$, and present some theoretical results regarding the balance between fairness and efficiency.

7 Conclusions & Future Research

In this paper, we presented the first *scalable* robust formulations of kidney exchange. Our methods address both uncertainty over the *quality* and the *existence* of a potential transplant. On real and simulated data from a large, fielded kidney exchange, we showed that our methods (i) clear the market within seconds and (ii) result in more predictable and better quality matchings than the status quo.

Adapting automated ethical decision-making frameworks that aggregate noisy human value judgments (Noothigattu *et al.* 2018; Freedman *et al.* 2018; Bonnefon *et al.* 2016) into our robust formulation is a natural way to handle uncertainty in the weights determined by a committee of stakeholders. Approaching *dynamic* kidney exchange, where participants arrive and depart over time, via robust reinforcement learning methods would be fruitful (Lim *et al.* 2013; Xu and Mannor 2010).

References

[Abraham *et al.* 2007] David Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 295–304, 2007.

[Anderson *et al.* 2015] Ross Anderson, Itai Ashlagi, David Gamarnik, and Alvin E Roth. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences*, 112(3):663–668, 2015.

[Ashlagi *et al.* 2012] Itai Ashlagi, David Gamarnik, Michael Rees, and Alvin E. Roth. The need for (long) chains in kidney exchange. NBER Working Paper No. 18202, July 2012.

[Ashlagi *et al.* 2013] Itai Ashlagi, Patrick Jaillet, and Vahideh H. Manshadi. Kidney exchange in dynamic sparse heterogenous pools. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 25–26, 2013.

[Balas 1989] Egon Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.

[Ben-Tal *et al.* 2009] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[Bertsimas and Sim 2004] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[Bertsimas *et al.* 2011a] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[Bertsimas *et al.* 2011b] Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. The price of fairness. *Operations Research*, 59(1):17–31, 2011.

[Biró *et al.* 2009] Péter Biró, David F Manlove, and Romeo Rizzi. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications*, 1(04):499–517, 2009.

[Biró *et al.* 2017] Péter Biró, Lisa Burnapp, Bernadette Haase, Aline Hemke, Rachel Johnson, Joris van de Klundert, and David Manlove. Kidney exchange practices in Europe, 2017. First Handbook of the COST Action CA15210: European Network for Collaboration on Kidney Exchange Programmes.

[Bonnefon *et al.* 2016] Jean-François Bonnefon, Azim Shariff, and Iyad Rahwan. The social dilemma of autonomous vehicles. *Science*, 352(6293):1573–1576, 2016.

[Caragiannis *et al.* 2009] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. The efficiency of fair division. International Workshop on Internet and Network Economics (WINE), 2009.

[Chen *et al.* 2012] Yanhua Chen, Yijiang Li, John D. Kalbfleisch, Yan Zhou, Alan Leichtman, and Peter X.-K. Song. Graph-based optimization algorithm and software on kidney exchanges. *IEEE Transactions on Biomedical Engineering*, 59:1985–1991, 2012.

[Chen *et al.* 2017] Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 4708–4717, 2017.

[Dickerson *et al.* 2012] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Optimizing kidney exchange with transplant chains: Theory and reality. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 711–718, 2012.

[Dickerson *et al.* 2014] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Price of fairness in kidney exchange. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1013–1020, 2014.

[Dickerson *et al.* 2016] John P. Dickerson, David Manlove, Benjamin Plaut, Tuomas Sandholm, and James Trimble. Position-indexed formulations for kidney exchange. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016.

[Dickerson *et al.* 2018] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Failure-aware kidney exchange. *Management Science*, 2018. To appear; earlier version appeared at EC-13.

[Ding *et al.* 2018] Yichuan Ding, Dongdong Ge, Simai He, and Christopher T. Ryan. A non-asymptotic approach to analyzing kidney exchange graphs. *Operations Research*, 2018. To appear; earlier version appeared at EC-15.

[Freedman *et al.* 2018] Rachel Freedman, J Schaich Borg, Walter Sinnott-Armstrong, J Dickerson, and Vincent Conitzer. Adapting a kidney exchange algorithm to align with human values. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[Gentry *et al.* 2005] Sommer Gentry, Dorry Segev, and R. A. Montgomery. A comparison of populations served by kidney paired donation and list paired donation. *American Journal of Transplantation*, 5(8):1914–1921, 2005.

[Glorie *et al.* 2014] Kristiaan M. Glorie, J. Joris van de Klundert, and Albert P. M. Wagelmans. Kidney exchange with long chains: An efficient pricing algorithm for clearing barter exchanges with branch-and-price. *Manufacturing & Service Operations Management (MSOM)*, 16(4):498–512, 2014.

[Glorie 2012] Kristiaan M. Glorie. Estimating the probability of positive crossmatch after negative virtual crossmatch. Technical report, Erasmus School of Economics, 2012.

[Glorie 2014] Kristiaan Glorie. *Clearing barter exchange markets: Kidney exchange and beyond*. PhD dissertation, Erasmus University Rotterdam, 2014.

[Gurobi Optimization, Inc. 2018] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2018.

[Klimentova *et al.* 2016] Xenia Klimentova, João Pedro Pedroso, and Ana Viana. Maximising expectation of the number of transplants in kidney exchange programmes. *Computers & Operations Research*, 73:1–11, 2016.

[Lim *et al.* 2013] Shiau Hong Lim, Huan Xu, and Shie Mannor. Reinforcement learning in robust markov decision processes. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2013.

[Manlove and O'Malley 2015] David Manlove and Gregg O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM Journal of Experimental Algorithmics*, 19(1), 2015.

[McElfresh and Dickerson 2018] Duncan C. McElfresh and John P. Dickerson. Balancing lexicographic fairness and a utilitarian objective with application to kidney exchange. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[Mevissen *et al.* 2013] Martin Mevissen, Emanuele Ragnoli, and Jia Yuan Yu. Data-driven distributionally robust polynomial optimization. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 37–45, 2013.

[Noothigattu *et al.* 2018] Ritesh Noothigattu, Snehalkumar Neil S. Gaikwad, Edmond Awad, Sohan D'Souza, Iyad Rahwan, Pradeep Ravikumar, and Ariel D. Procaccia. A voting-based system for ethical decision making. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[Petrik and Subramanian 2014] Marek Petrik and Dharmashankar Subramanian. RAAM: The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1979–1987, 2014.

[Plaut *et al.* 2016] Benjamin Plaut, John P. Dickerson, and Tuomas Sandholm. Fast optimal clearing of capped-chain barter exchanges. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 601–607, 2016.

[Poss 2014] Michael Poss. Robust combinatorial optimization with variable cost uncertainty. *European Journal of Operational Research*, 237(3):836–845, 2014.

[Rapaport 1986] F. T. Rapaport. The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proceedings*, 18:5–9, 1986.

[Roth *et al.* 2004] Alvin Roth, Tayfun Sönmez, and Utku Ünver. Kidney exchange. *Quarterly Journal of Economics*, 119(2):457–488, 2004.

[Roth *et al.* 2005] Alvin Roth, Tayfun Sönmez, and Utku Ünver. A kidney exchange clearinghouse in New England. *American Economic Review*, 95(2):376–380, 2005.

[UNOS 2015] UNOS. Revising kidney paired donation pilot program priority points, 2015. OPTN/UNOS Public Comment Proposal.

[Xu and Mannor 2010] Huan Xu and Shie Mannor. Distributionally robust Markov Decision Processes. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2010.

[Xu *et al.* 2009] Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and LASSO. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1801–1808, 2009.

Appendix to: Scalable Robust Kidney Exchange

A Edge Weight Robust Formulation

We develop an edge weight robust formulation with uncertainty set $\mathcal{U}^I_\Gamma$, based on the position-indexed chain-edge formulation formulation (PICEF) introduced by Dickerson *et al.* (2016). In Section A.1 we review the PICEF formulation, and in Section A.2 we introduce our linear formulation for edge-weight robust kidney exchange.

Section A.3 and Section A.4 describe the solution methods for constant uncertainty budget Γ and variable uncertainty budget $\gamma(|\mathbf{x}|)$ for decision variables $\mathbf{x}$, respectively.

For simplicity, we use the abbreviation $KEX(\mathcal{U})$ to refer to the robust kidney exchange problem, with uncertainty set $\mathcal{U}$.

A.1 PICEF Formulation

The position-indexed chain-edge formulation (PICEF) is a compact formulation proposed by Dickerson *et al.* (2016), with a polynomial (with regard to the compatibility graph size and exogenous cycle cap) count of both variables and constraints. This formulation uses the following parameters:

- G: kidney exchange graph, consisting of edges $e \in E$ and vertices $v \in V = P \cup N$, including patient-donor pairs P and NDDs N
- C: a set of cycles on exchange graph G
- L: chain cap (maximum number of edges used in a chain)
- w_e: edge weights for each edge $e \in E$
- w^C_c: cycle weights for each cycle $c \in C$, defined as $w^C_c = \sum_{e\in c} w_e$

This formulation uses one decision variable for each cycle, and several decision variables for each edge to represent chains:

- $z_c \in \{0,1\}$: 1 if cycle c is used in the matching, and 0 otherwise
- $y_{ek} \in \{0,1\}$: 1 if edge e is used at position k in a chain, and 0 otherwise

Note that edges between an NDD $n \in N$ and a patient-donor vertex $v \in P$ may only take position 1 in a chain, while edges between two patient-donor pairs may take any position $1, 2, \ldots, L$ in a chain. For convenience, we define the function $\mathcal{K}$ for each edge e, such that $\mathcal{K}(e)$ is the set of all possible positions that edge e may take in a chain.

$$\mathcal{K}(e) = \begin{cases} \{1\} & e \text{ begins in } n \in N \\ \{1, 2, \ldots, L\} & e \text{ begins in } v \in P \end{cases}$$

We also use the following notation for flow into and out of vertices:

- $\delta^-(s)$ and $\delta^-(S)$: the set of edges into vertex s or set of vertices S
- $\delta^+(s)$ and $\delta^+(S)$: the set of edges out of vertex s or set of vertices S

The PICEF formulation is given in Problem (5).

$$\max \quad \sum_{e\in E} \sum_{k\in\mathcal{K}(e)} w_e y_{ek} + \sum_{c\in C} w_c z_c \tag{5a}$$

$$\text{s.t.} \quad \sum_{e\in\delta^-(i)} \sum_{k\in\mathcal{K}(e)} y_{ek} + \sum_{c\in C: i\in c} z_c \leq 1 \qquad i \in P \tag{5b}$$

$$\sum_{e\in\delta^+(i)} y_{e1} \leq 1 \qquad i \in N \tag{5c}$$

$$\sum_{\substack{e\in\delta^-(i)\wedge \\ k\in\mathcal{K}(e)}} y_{ek} \geq \sum_{e\in\delta^+(i)} y_{e,k+1} \qquad \begin{array}{l} i \in P \\ k \in \{1,\ldots,L-1\} \end{array} \tag{5d}$$

$$y_{ek} \in \{0,1\} \qquad e \in E, k \in \mathcal{K}(e) \tag{5e}$$

$$z_c \in \{0,1\} \qquad c \in C \tag{5f}$$

The Objective (5a) maximizes the total weight of a matching, defined by the cycle decision variables z_c and edge variables y_{ek}. Feasible matchings may only use each edge once, and must contain valid chains. Capacity constraints ensure that each edge is used at most once:
The capacity constraints for each vertex are as follows:

- Constraint 5b: each patient-donor vertex $i \in P$ may only participate in one cycle or one chain
- Constraint 5c: each NDD $i \in N$ may only participate in one chain

Valid chains must begin in an NDD, and conserve flow through patient-donor pairs:

- Constraint 5d: a patient-donor vertex $i \in P$ can only have an outgoing edge at position $k+1$ in a chain if it has an incoming edge at position k

In the next section we present the mixed integer linear program formulation for $KEX(\mathcal{U}^I_\Gamma)$, based on PICEF.

A.2 Our Robust Formulation

To simplify notation, let $\mathcal{M}^P$ be the set of all feasible matchings for the PICEF formulation. The edge weight robust kidney exchange problem $KEX(\mathcal{U}^I_\Gamma)$ is given in Equation (6).

$$\max \min_{\mathbf{w} \in \mathcal{U}^I_\Gamma} \quad \sum_{e \in E} \sum_{k \in \mathcal{K}(e)} w_e y_{ek} + \sum_{c \in C} w_c z_c \tag{6a}$$

$$\text{s.t.} \quad (\mathbf{z}, \mathbf{y}) \in \mathcal{M}^P \tag{6b}$$

Proposition 1 states that this problem is identical to the robust formulation with one-sided uncertainty set $\mathcal{U}^{I1}_\Gamma$—that is, $KEX(\mathcal{U}^I_\Gamma) = KEX(\mathcal{U}^{I1}_\Gamma)$.

Proposition 1. *The problems $KEX(\mathcal{U}^I_\Gamma)$ and $KEX(\mathcal{U}^{I1}_\Gamma)$ are equivalent.*

Proof. In $KEX(\mathcal{U}^I_\Gamma)$ (Problem 6), edge weights are minimized with respect to uncertainty set $\mathcal{U}^I_\Gamma$. The objective is minimized when up to Γ edge weights are reduced by the maximum amount within $\mathcal{U}^I_\Gamma$ (d_e), and one edge weight is reduced by $(\Gamma - \lfloor \Gamma \rfloor) d_e$. That is, $KEX(\mathcal{U}^I_\Gamma)$ only considers realized edge weights on the interval $\hat{w}_e \in [w_e - d_e, w_e]$. This is equivalent to restricting α_e to the interval $[-1, 0]$ in $\mathcal{U}^I_\Gamma$, which is equivalent to $\mathcal{U}^{I1}_\Gamma$. □

Thus we must solve Problem (7), with uncertainty set $\mathcal{U}^{I1}_\Gamma$.

$$\max \min_{\mathbf{w} \in \mathcal{U}^{I1}_\Gamma} \quad \sum_{e \in E} \sum_{k \in \mathcal{K}(e)} w_e y_{ek} + \sum_{c \in C} w_c z_c \tag{7a}$$

$$\text{s.t.} \quad (\mathbf{z}, \mathbf{y}) \in \mathcal{M}^P \tag{7b}$$

Next we develop a MILP formulation for Problem 7 by directly minimizing its Objective (7a). This minimum occurs when $\lfloor \Gamma \rfloor$ edge weights are reduced by d_e, and one edge weight is reduced by $(\Gamma - \lfloor \Gamma \rfloor) d_e$. For this reason we refer to d_e as the *discount value* of edge e, and all edges that receive reduced weight in the robust matching are *discounted*.

For simplicity, we define a variable $\hat{y}_e$ for each edge $e \in E$ such that $\hat{y}_e$ is 1 if edge e is used in the matching, and 0 otherwise. Note that edge e is used in the matching if it is used in a chain (i.e. any $y_{ek} = 1$), or if it is used in a cycle (i.e. $z_c = 1$ for any cycle c containing e). Thus we define variables $\hat{y}_e$ using the following constraint.

$$\begin{aligned} \sum_{k \in \mathcal{K}(e)} y_{ek} + \sum_{c \in C : e \in c} z_c = \hat{y}_e \quad &, e \in E \\ \hat{y}_e \in \{0,1\}, e \in E& \end{aligned}$$

Next we minimize the Objective (7a)w.r.t. $\mathcal{U}^{I1}_\Gamma$, by discounting up to Γ edges. Note that if only $G < \Gamma$ edges are used in a matching, only G edge weights may be discounted. Thus let $\Gamma' \equiv \min\{G, \Gamma\}$ be the number of discounted edges, with

$$G = \sum_{e \in E} \hat{y}_e,$$

the total number of edges used in the matching. To linearize the definition of Γ' we introduce variable h, which is 1 if $G < \Gamma$ and 0 otherwise. The statement $\Gamma' = \min(G, \Gamma)$ can be linearized using the following constraints,

$$\begin{aligned} \Gamma - G &\leq Wh \\ G - \Gamma &\leq W(1-h) \\ G - Wh &\leq \Gamma' \\ \Gamma - W(1-h) &\leq \Gamma' \\ h &\in \{0,1\} \end{aligned}$$

where W is a large constant.

The objective of Problem (7) is minimized the the Γ' discounted edges are those with the *largest* discount value d_e. To select these edges we introduce variables $g_e \in \{0, 1\}$ for each edge $e \in E$. Let m be the smallest d_e of any discounted edge—that is, m is the $\lceil \Gamma' \rceil^{th}$ highest d_e of any edge used in the matching. We define g_e as follows

$$g_e = \begin{cases} 1 & \text{if } d_e \geq m \\ 0 & \text{otherwise} \end{cases}$$

That is, g_e is 0 if d_e is smaller than the $\lceil \Gamma' \rceil^{th}$ highest discount value of edges used in the matching, and 1 otherwise. We can define these variables using linear constraints in two steps. First, note that variables g_e and d_e must obey the same ordering relation. That is, $g_i \geq g_j \Leftrightarrow d_i \geq d_j$ must hold for all $i, j \in E, i \neq j$. Note that variables d_e are constant, and can be sorted during pre-processing. Let $\geq_d$ indicate this ordering relation.

Next we ensure that only Γ' edges are discounted. Note that $\hat{y}_e = 1$ implies that edge e is used in the matching. Edge e should be discounted if it is used in the matching, and if d_e is above the minimum discount value (that is, $g_e = 1$). Thus, edge e should be discounted if the following identity holds

$$g_e \hat{y}_e = 1$$

Using this observation, we can ensure that exactly Γ' edges are discounted with the following constraint,

$$\sum_{e \in E} g_e \hat{y}_e = \Gamma'.$$

For any feasible matching $M = (\mathbf{y}, \mathbf{z})$, we can directly solve the minimization in Problem (7) by discounting the Γ' edges used in M with the largest discount values. This is accomplished using variables g_e; Equation (8) gives the solution of this minimization when Γ is integer, which is expressed as a function $Z(\mathbf{y}, \mathbf{z})$; the next section extends this formulation to accommodate non-integer Γ.

$$\begin{aligned}
Z(\mathbf{y}, \mathbf{z}) = \quad & \sum_{e \in E} \sum_{k \in \mathcal{K}(e)} w_e y_{ek} + \sum_{c \in C} w_c z_c - \sum_{e \in E} g_e d_e \hat{y}_e && \text{(8a)} \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}(e)} y_{ek} + \sum_{c \in C : e \in c} z_c = \hat{y}_e, e \in E && \text{(8b)} \\
& \sum_{e \in E} \hat{y}_e = G && \text{(8c)} \\
& \Gamma - G \leq Wh && \text{(8d)} \\
& G - \Gamma \leq W(1 - h) && \text{(8e)} \\
& G - Wh \leq \Gamma' && \text{(8f)} \\
& \Gamma - W(1 - h) \leq \Gamma' && \text{(8g)} \\
& \sum_{e \in E} g_e \hat{y}_e = \Gamma' && \text{(8h)} \\
& g_e, \hat{y}_e \in \{0, 1\}, e \in E && \text{(8i)} \\
& g_a \geq_d g_b, a, b \in E, a \neq b && \text{(8j)} \\
& h \in \{0, 1\} && \text{(8k)}
\end{aligned}$$

Note that this formulation contains two sets of quadratic terms: $g_e y_{ek}$ for $k \in \mathcal{K}(e)$ for $e \in E$, and $g_e z_c$ for $c \in C$ and $e \in E$. We linearize these terms in the following section, after considering non-integer Γ.

Non-Integer Γ The number of discounted edges Γ' may be integer or non-integer valued. When Γ' is not integer valued, up to $\lfloor \Gamma' \rfloor$ edges are fully discounted by value d_e, and the edge with the smallest discount value is discounted by $(\Gamma - \lfloor \Gamma \rfloor) d_e$. We include this fractional discount by using two sets of indicator variables g^f_e and g^p_e for all $e \in E$, and then discount each edge e as follows:

- e is fully discounted if $g^p_e = g^f_e = 1$.
- e is discounted by fractional amount $(\Gamma - \lfloor \Gamma \rfloor)$ if $g^f_e = 0$ and $g^p_e = 1$
- e is not discounted if $g^f_e = g^p_e = 0$.

Thus if Γ' is integer, $g_e^f = g_e^p$ for all $e \in E$; if Γ' is not integer, then $\lceil \Gamma' \rceil$ matching edges should be at least partially discounted ($g_e^p = 1$), and $\lfloor \Gamma' \rfloor$ matching edges should be fully discounted ($g_e^p = g_e^f = 1$). These indicator variables are defined in the same way as g_e in Equation (8): $g_e^f, g_e^p \in \{0, 1\}$, and they obey the same ordering relation as d_e. However, the number of matching edges with $g_e^f = 1$ can be different than the number of edges with $g_e^p = 1$.

First note that $\lceil \Gamma' \rceil$ matching edges must have $g_e^p = 1$. Recall that G is the number of matching edges, and $\Gamma' = \min(\Gamma, G)$; if $\Gamma < G$, then $\lceil \Gamma' \rceil = \lceil \Gamma \rceil$, and otherwise $\lceil \Gamma' \rceil = G$. The variable h is defined to be 1 if $G < \Gamma$ and 0 otherwise. Thus, we use the following constraint to require that $\lceil \Gamma' \rceil$ matching edges have $g_e^p = 1$:

$$\sum_{e \in E} g_e^p \hat{y}_e = hG + (1-h)\lceil \Gamma \rceil.$$

Similarly, we can require that $\lfloor \Gamma' \rfloor$ edges have $g_e^f = 1$ with the following constraint

$$\sum_{e \in E} g_e^f \hat{y}_e = hG + (1-h)\lfloor \Gamma \rfloor.$$

Thus if $G < \Gamma$, then all G matching edges have $g_e^f = g_e^p = 1$; otherwise, there are $\lceil \Gamma \rceil$ matching edges with $g_e^p = 1$, and $\lfloor \Gamma \rfloor$ matching edges with $g_e^f = 1$, where the matching edge with the smallest discount has $g_e^f = 0$ and $g_e^p = 1$.

Using these indicator variables, the new objective of the robust formulation is

$$\begin{aligned} \max \sum_{e \in E} \sum_{k \in \mathcal{K}(e)} w_e y_{ek} + \sum_{c \in C} w_c z_c - (1 - \Gamma + \lfloor \Gamma \rfloor) \sum_{e \in E} g_e^f d_e \hat{y}_e \\ - (\Gamma - \lfloor \Gamma \rfloor) \sum_{e \in E} g_e^p d_e \hat{y}_e \end{aligned}$$

which discounts an edge e by weight d_e if $g_e^f = g_e^p = 1$, and by weight $d_e\,(\Gamma - \lfloor \Gamma \rfloor)$ if $g_e^f = 0$ and $g_e^p = 1$. Note that there are two sets of quadratic terms in this problem: $g_e^f \hat{y}_e$ and $g_e^p \hat{y}_e$ for all $e \in E$. To linearize these terms we introduce the variables $\hat{g}_e^f \equiv g_e^f \hat{y}_e$ and $\hat{g}_e^p \equiv g_e^p \hat{y}_e$, which we define using the following constraints.

$$\left.\begin{array}{ll} \hat{g}_e^f & \leq g_e^f \\ \hat{g}_e^f & \leq \hat{y}_e \\ \hat{g}_e^f & \geq g_e^f + \hat{y}_e - 1 \end{array}\right. \quad , e \in E$$

$$\hat{g}_e^f \in \{0, 1\}, e \in E$$

$$\left.\begin{array}{ll} \hat{g}_e^p & \leq g_e^p \\ \hat{g}_e^p & \leq \hat{y}_e \\ \hat{g}_e^p & \geq g_e^p + \hat{y}_e - 1 \end{array}\right. \quad , e \in E$$

$$\hat{g}_e^p \in \{0, 1\}, e \in E$$

To linearize the term hG, we introduce variable $\hat{g} \equiv hG$, which is defined using the following constraints. As before, W is a large constant.

$$\begin{aligned} &\hat{h} \leq hW \\ &\hat{h} \leq G \\ &\hat{h} \geq G - (1-h)W \\ &\hat{h} \geq 0 \end{aligned}$$

Finally, for any feasible matching $M = (\mathbf{y}, \mathbf{z})$, we can directly solve the minimization in problem 7 by discounting the Γ' edges used in M with the largest discount values. This is accomplished using variables g_e^f and g_e^p; Equation (9) gives the solution of this minimization for general $\Gamma > 0$.

$$Z(\mathbf{y}, \mathbf{z}) = \sum_{e \in E} \sum_{k \in \mathcal{K}(e)} w_e y_{ek} + \sum_{c \in C} w_c z_c - (1 - \Gamma + \lfloor \Gamma \rfloor) \sum_{e \in E} \hat{g}_e^f d_e \tag{9a}$$

$$- (\Gamma - \lfloor \Gamma \rfloor) \sum_{e \in E} \hat{g}_e^p d_e \tag{9b}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}(e)} y_{ek} + \sum_{c \in C : e \in c} z_c = \hat{y}_e, e \in E \tag{9c}$$

$$\sum_{e \in E} \hat{y}_e = G \tag{9d}$$

$$\Gamma - G \leq Wh \tag{9e}$$

$$G - \Gamma \leq W(1-h) \tag{9f}$$

$$G - Wh \leq \Gamma' \tag{9g}$$

$$\Gamma - W(1-h) \leq \Gamma' \tag{9h}$$

$$\sum_{e \in E} \hat{g}^p_e = \hat{h} + (1-h)\lceil \Gamma \rceil \tag{9i}$$

$$\sum_{e \in E} \hat{g}^f_e = \hat{h} + (1-h)\lfloor \Gamma \rfloor \tag{9j}$$

$$\begin{array}{ll} \hat{g}^f_e & \leq g^f_e \\ \hat{g}^f_e & \leq \hat{y}_e \\ \hat{g}^f_e & \geq g^f_e + \hat{y}_e - 1 \end{array} \quad , e \in E \tag{9k}$$

$$\begin{array}{ll} \hat{g}^p_e & \leq g^p_e \\ \hat{g}^p_e & \leq \hat{y}_e \\ \hat{g}^p_e & \geq g^p_e + \hat{y}_e - 1 \end{array} \quad , e \in E \tag{9l}$$

$$\hat{h} \leq hW \tag{9m}$$

$$\hat{h} \leq G \tag{9n}$$

$$\hat{h} \geq G - (1-h)W \tag{9o}$$

$$g^p_e, g^f_e, \hat{y}_e \in \{0,1\}, e \in E \tag{9p}$$

$$g^f_a \geq_d g^f_b, a, b \in E, a \neq b \tag{9q}$$

$$g^p_a \geq_d g^p_b, a, b \in E, a \neq b \tag{9r}$$

$$\hat{g}^p_e, \hat{g}^f_e \in \{0,1\}, e \in E \tag{9s}$$

$$h \in \{0,1\} \tag{9t}$$

$$\hat{h} \geq 0 \tag{9u}$$

Equation (9) is the direct minimization of the Objective of $KEX(\mathcal{U}^{I1}_{\Gamma})$ (7a). Thus we directly apply this minimization solution to the original Problem (7), to obtain the final linear formulation in Equation (10).

$$\max \sum_{e \in E} \sum_{k \in \mathcal{K}(e)} w_e y_{ek} + \sum_{c \in C} w_c z_c - (1 - \Gamma + \lfloor \Gamma \rfloor) \sum_{e \in E} \hat{g}^f_e d_e \tag{10a}$$

$$- (\Gamma - \lfloor \Gamma \rfloor) \sum_{e \in E} \hat{g}^p_e d_e \tag{10b}$$

$$\text{s.t.} \quad \sum_{e \in \delta^-(i)} \sum_{k \in \mathcal{K}(e)} y_{ek} + \sum_{c \in C : i \in c} z_c \leq 1 \qquad i \in P \tag{10c}$$

$$\sum_{e \in \delta^+(i)} y_{e1} \leq 1 \qquad i \in N \tag{10d}$$

$$\sum_{\substack{e \in \delta^-(i) \wedge \\ k \in \mathcal{K}(e)}} y_{ek} \geq \sum_{e \in \delta^+(i)} y_{e,k+1} \qquad \begin{array}{l} i \in P \\ k \in \{1, \ldots, L-1\} \end{array} \tag{10e}$$

$$\sum_{k \in \mathcal{K}(e)} y_{ek} + \sum_{c \in C : e \in c} z_c = \hat{y}_e \qquad e \in E \tag{10f}$$

$$\sum_{e \in E} \hat{y}_e = G \tag{10g}$$

$$\Gamma - G \leq Wh \tag{10h}$$

$$G - \Gamma \leq W(1-h) \tag{10i}$$

$$G - Wh \leq \Gamma' \tag{10j}$$

$$\Gamma - W(1-h) \leq \Gamma' \tag{10k}$$

$$\sum_{e \in E} \hat{g}^p_e = \hat{h} + (1-h)\lceil \Gamma \rceil \tag{10l}$$

$$\sum_{e \in E} \hat{g}^f_e = \hat{h} + (1-h)\lfloor \Gamma \rfloor \tag{10m}$$

$$\begin{aligned} \hat{g}^f_e &\leq g^f_e \\ \hat{g}^f_e &\leq \hat{y}_e \\ \hat{g}^f_e &\geq g^f_e + \hat{y}_e - 1 \end{aligned} \qquad e \in E \tag{10n}$$

$$\begin{aligned} \hat{g}^p_e &\leq g^p_e \\ \hat{g}^p_e &\leq \hat{y}_e \\ \hat{g}^p_e &\geq g^p_e + \hat{y}_e - 1 \end{aligned} \qquad e \in E \tag{10o}$$

$$\hat{h} \leq hW \tag{10p}$$

$$\hat{h} \leq G \tag{10q}$$

$$\hat{h} \geq G - (1-h)W \tag{10r}$$

$$g^f_a \geq_d g^f_b \qquad a, b \in E, a \neq b \tag{10s}$$

$$g^p_a \geq_d g^p_b \qquad a, b \in E, a \neq b \tag{10t}$$

$$y_{ek} \in \{0, 1\} \qquad e \in E, k \in \mathcal{K}(e) \tag{10u}$$

$$z_c \in \{0, 1\} \qquad c \in C \tag{10v}$$

$$g^p_e, g^f_e, \hat{y}_e \in \{0, 1\} \qquad e \in E \tag{10w}$$

$$\hat{g}^p_e, \hat{g}^f_e \in \{0, 1\} \qquad e \in E \tag{10x}$$

$$h \in \{0, 1\} \tag{10y}$$

$$\hat{h} \geq 0 \tag{10z}$$

A.3 Solution Method for Constant Budget Γ

This section describes the algorithm for solving the edge-weight robust formulation in Section A.2, when it is unreasonable to find all cycles in the exchange graph during preprocessing. We build on the cycle pricing method in Dickerson *et al.* (2016), which in turn built on corrected versions of methods presented by Glorie *et al.* (2014) and Plaut *et al.* (2016).

This method begins by solving the LP relaxation of Problem (10) on a *reduced model* (using a small number of cycles), and then identifying *positive-price cycles*—which may improve the solution—and adding these to the model. If no positive-price cycles exist, then the solution is optimal on the reduced LP relaxation. This process is known as the *pricing problem*.

After optimizing the reduced LP relaxation, we proceed in one of two ways

1. If the solution is fractional, then we fix one of the fractional variables and branch, as in a standard branch-and-bound tree,
2. If the solution is integral, then it is the optimal solution to Problem (10).

This combination of cycle pricing and branch-and-bound is known as *branch-and-price*.

Algorithm 1 is the branch-and-price method for solving Problem (10). There are only two inputs to this algorithm: the kidney exchange graph G, and the set of fixed decision variables $\mathbf{X}_F$. At each branch in the search tree, a new decision variable is fixed to either 0 or 1 and added to $\mathbf{X}_F$. When both 1) no positive price cycles exist for reduced model $\mathbf{M}$ and solution $\mathbf{X}$, and 2) the solution $\mathbf{X}$ is integral, then $\mathbf{X}$ is returned.

The branch-and-price method in Algorithm 1 requires a cycle-pricing algorithm $\mathrm{GetCycles}$. This algorithm either returns positive-price cycles—using the reduced model $\mathbf{M}$ and the current solution to the LP relaxation, $\mathbf{X}$—or determines that none exist. We adapt the cycle-pricing algorithm usesd by Dickerson *et al.* (2016) to solve the PICEF formulation, which is based on (Glorie *et al.* 2014) and (Plaut *et al.* 2016). These algorithms calculate the price p_c of cycle c as

$$p_c = \sum_{e \in c} (w_e - \delta_v)$$

where w_e is the weight of edge e in cycle c, and δ_e is the dual value of the vertex where e ends. In the edge-weight robust problem, each edge e may receive its nominal weight w_e or its discounted weight $(w_e - d_e)$. It is not obvious whether the nominal or discounted weights should be used during cycle pricing.

To illustrate this problem, assume we know the optimal solution $\mathbf{X}$ to Problem (10), and the set of cycles C used in $\mathbf{X}$. We consider two methods for cycle pricing.

ALGORITHM 1: BranchAndPrice

```
input   : $G, \mathbf{X}_F$
output  : Optimal Matching $X$
Generate subset of cycles $C'$, in $G$;
Create reduced model $\mathbf{M}$, with cycles $C'$;
$\mathbf{X} \leftarrow$ Solve LP relaxation of $\mathbf{M}$ ;
$C^+ \leftarrow$ CyclePrice$(G, \mathbf{X})$;
while $C^+ \neq \varnothing$ do
    Add cycles $C^+$ to $\mathbf{M}$;
    $\mathbf{X} \leftarrow$ solve LP relaxation of $\mathbf{M}$;
    $C^+ \leftarrow$ CyclePrice$(G, \mathbf{X})$;
end
if $\mathbf{X}$ is fractional then
    Find fractional binary variable $X_i \in \mathbf{X}$ closest to $0.5$;
    $\mathrm{BranchAndPrice}(G, \mathbf{X}_F \cup (X_i = 0))$;
    $\mathrm{BranchAndPrice}(G, \mathbf{X}_F \cup (X_i = 1))$;
else
    return X
```

1. Calculate cycle prices using discounted edge weights $(w_e - d_e)$.

   Assume that, for some cycle $c \in C$, none of the edges in c are discounted in $\mathbf{X}$. During branch-and-price, it may occur that—before adding c to the reduced model—the following inequalities hold

$$\sum_{e \in c} (w_e - d_e - \delta_v) \leq 0$$
$$\sum_{e \in c} (w_e - \delta_v) > 0$$

   If discounted edge weights are used during pricing, c appears to have negative price—and will not be added to the reduced model. In this case, the calculated price is incorrectly negative, branch-and-price may return a sub-optimal solution.

2. Calculate cycle prices using nominal edge weights w_e.

   Assume that, for some cycle $c' \notin C$, *all* of the edges in c' are discounted when it is added to the reduced model. It may occur that the following inequalities hold:

$$\sum_{e \in c'} (w_e - d_e - \delta_v) \leq 0$$
$$\sum_{e \in c'} (w_e - \delta_v) > 0$$

   In this case, using nominal edge weights for cycle pricing will incorrectly determine that c' has a positive price, and will add c' to the reduced model.

Neither of these methods is ideal—using discounted weights can result in a sub-optimal solution, while using nominal weights adds cycles to the reduced model. Instead, we calculate cycle prices using discounted edge weights *only* for edges that will be discounted in *any* matching, and nominal edge weights for all other edges. As discussed in Section A.2, up to Γ edges are discounted in every solution to Problem (10); these are the edges with the largest discount values d_e. For any exchange graph with $|E|$ edges, the $\min(\Gamma, |M|)$ edges with the largest discount values are *always* discounted if they are used in a solution to Problem (10). Algorithm 2 describes this method, which uses the cycle pricer of (Glorie *et al.* 2014) as a subroutine. Proposition 2 states that this method never incorrectly determines that a cycle has negative price—and therefore never results in a sub-optimal solution.

Proposition 2. *Algorithm 2 never determines that a positive-price cycle has a negative price.*

A.4 Solution Method for Variable Budget γ

In this section we describe a method for solving the edge-weight robust kidney exchange problem with variable budget, $KEX(\mathcal{U}^{I1}_{\gamma})$. Theorem 2 is a direct adaptation of Theorem 4 of (Poss 2014) to the edge-weight uncertain kidney exchange problem, which states that the solution of $KEX(\mathcal{U}^{I1}_{\gamma})$ can be found by solving several cardinality-restricted instances of $KEX(\mathcal{U}^{I1}_{\Gamma})$.

```
ALGORITHM 2: CyclePrice
input  : $G = (V, E), \mathbf{X}$
output : Cycle Prices
$d^* \leftarrow \Gamma^{th}$ highest discount value $d_e$ in $E$;
$w_e^* \leftarrow \begin{cases} w_e - d_e & \text{if } d_e \geq d^* \\ w_e & \text{otherwise} \end{cases}$;
return PositivePriceCycles$(G, L, \mathbf{X}, w_e^*)$, the cycle pricer from (Glorie et al. 2014)
```

Theorem 2. *Let $\mathcal{M}$ be the set of feasible matchings, with edge decision variables $\mathbf{x} \in \mathcal{M} \subset \{0,1\}^{|E|}$. The solution to $KEX(\mathcal{U}_\gamma^{I1})$ can be found by solving $|E|$ cardinality-restricted instances of $KEX(\mathcal{U}_\Gamma^{I1})$,*

$$\begin{aligned} \max \min_{\hat{\mathbf{w}} \in \mathcal{U}_\Gamma^I} \quad & \mathbf{x} \cdot \hat{\mathbf{w}} \\ \textit{s.t.} \quad & \mathbf{x} \in \mathcal{M} \\ & \|\mathbf{x}\| \leq k \\ & \Gamma = \gamma(k) \end{aligned}$$

with $k = 1, \ldots, |E|$, and taking the maximum-weight solution.

The proof of this theorem is identical to the proof of Theorem 4 in Poss (2014), and is omitted here. In practice, feasible matchings use far fewer than $|E|$ edges, and thus many fewer than $|E|$ instances of $KEX(\mathcal{U}_\Gamma^{I1})$ must be solved. Algorithm 3 describes our method for solving $KEX(\mathcal{U}_\gamma^{I1})$, which first finds the maximum cardinality matching, and then solves each cardinality-restricted problem $KEX(\mathcal{U}_\Gamma^{I1})$.

```
ALGORITHM 3: EdgeWeightRobust-$\gamma$
input  : Function $\gamma$, exchange graph $G$
output : Optimal matching $\mathbf{x}$
Find the maximum cardinality matching $\mathbf{x}_C$;
for $k \leftarrow 1$ to $\|\mathbf{x}_C\|$ do
    $\Gamma \leftarrow \gamma(k)$;
    $\mathbf{x}_k^* \leftarrow$ solution to $KEX(\mathcal{U}_\Gamma^{I1})$, restricting cardinality to $k$;
end
return The maximum-weight matching in $\{\mathbf{x}_k^*\}$
```

B Edge Existence Robust Formulation

In this section we develop an edge existence robust formulation for kidney exchange, using uncertainty set $\mathcal{U}_\Gamma^w$. Our approach is based on a formulation introduced by Anderson *et al.* (2015), which adapts a formulation of the prize-collecting traveling salesman problem (PC-TSP). For simplicity, we use the abbreviation $KEX(\mathcal{U})$ to refer to the robust kidney exchange problem, with uncertainty set $\mathcal{U}$.

B.1 PC-TSP Formulation

We begin by overviewing the PC-TSP method proposed by Anderson *et al.* (2015); it is based on a method for solving the prize-collecting traveling salesman problem (PC-TSP) introduced by Balas (1989). We use a version of the PC-TSP formulation with a finite chain cap; the uncapped formulation is much more compact. (Due to high failure rates, most fielded exchanges incorporate a finite maximum length of chains. That cap can be quite high, e.g., 20 or more, but is typically not allowed to float freely with parts of the input size, e.g., $|V|$.) This formulation is especially useful because it allows us to define decision variables equal to each chain weight used in the matching, without explicitly enumerating all possible chains.

This formulation uses all of the same parameters as PICEF:

- G: kidney exchange graph, consisting of edges $e \in E$ and vertices $v \in V = P \cup N$, including patient-donor pairs P and NDDs N.
- C: a set of cycles on exchange graph G.
- L: chain cap (maximum number of edges used in a chain).
- w_e: edge weights for each edge $e \in E$.

- w_c^C: cycle weights for each cycle $c \in C$, defined as $w_c^{\mathcal{C}} = \sum_{e \in c} w_e$.

PC-TSP uses one decision variable for each cycle (z_c) and each edge (y_e), and several auxiliary decision variables that help define the constraints:

- $z_c \in \{0, 1\}$: 1 if cycle c is used in the matching, and 0 otherwise.
- $y_e \in \{0, 1\}$: 1 if edge e is used in a chain, and 0 otherwise.
- $y_e^n \in \{0, 1\}$: 1 if edge e is used in a chain starting with NDD n, and 0 otherwise.
- w_n^N (auxiliary): total weight of the chain starting with NDD n.
- f_v^i and f_v^o (auxiliary): chain flow into and out of vertex $v \in P$, respectively.
- $f_v^{i,n}$ and $f_v^{i,n}$ (auxiliary): chain flow into and out of vertex $v \in P$, respectively, from a chain beginning with NDD $n \in N$.

The PC-TSP formulation with chain cap L is given in Problem 12. As before, we use the notation $\delta^-(v)$ for the set of edges into vertex v and $\delta^+(v)$ for the set of edges out of v.

$$\max \quad \sum_{n \in N} w_n^N + \sum_{c \in C} w_c^C z_c \tag{12a}$$

$$\text{s.t.} \quad \sum_{e \in E} w_e y_e^n = w_n^N \qquad n \in N \tag{12b}$$

$$\sum_{n \in N} y_e^n = y_e \qquad e \in E \tag{12c}$$

$$\sum_{e \in \delta^-(v)} y_e = f_v^i \qquad v \in V \tag{12d}$$

$$\sum_{e \in \delta^+(v)} y_e = f_v^o \qquad v \in V \tag{12e}$$

$$\sum_{e \in \delta^-(v)} y_e^n = f_v^{i,n} \qquad v \in V, n \in N \tag{12f}$$

$$\sum_{e \in \delta^+(v)} y_e^n = f_v^{o,n} \qquad v \in V, n \in N \tag{12g}$$

$$f_v^o + \sum_{c \in C : v \in c} z_c \leq \quad f_v^i + \sum_{c \in C : v \in c} z_c \leq 1 \qquad v \in P \tag{12h}$$

$$f_v^o \leq 1 \qquad v \in N \tag{12i}$$

$$\sum_{e \in \delta^-(S)} y_e \geq f_v^i \qquad S \subseteq P, v \in S \tag{12j}$$

$$\sum_{e \in E} y_e^n \leq L \qquad n \in N \tag{12k}$$

$$f_v^{o,n} \leq f^{i,v} \leq 1 \qquad v \in V, n \in N \tag{12l}$$

$$y_e \in \{0, 1\} \qquad e \in E \tag{12m}$$

$$z_c \in \{0, 1\} \qquad c \in C \tag{12n}$$

$$y_e^n \in \{0, 1\} \qquad e \in E, n \in N \tag{12o}$$

The objective 12a maximizes the total weight of a matching, defined by the cycle decision variables z_c and edge decision variables y^e. The auxiliary variables are defined using the following constraints:

- Constraint 12b: defines w_n^N.
- Constraint 12c: defines y_e, using y_e^n.
- Constraints 12d and 12e: define auxiliary variables f_v^i and f_v^o.
- Constraints 12f and 12g: define auxiliary variables $f_v^{i,n}$ and $f_v^{o,n}$.

There is only one capacity constraint for each patient-donor vertex and each NDD:

- Constraint 12h: each patient-donor vertex v may only be used in one cycle c; or, if v is used in a chain, chain flow out of v can only be nonzero if there is chain flow out of v.

- Constraint 12i: each NDD n may only start one chain.

The follow constraints ensure that chain flow is conserved, and enforce the chain cap L:

- Constraint 12k: chains can use no more than L edges.
- Constraint 12l: chain flow out of v can only be nonzero if there is chain flow out of v. This constraint is equivalent to 12h, but for variables $f_v^{o,n}$.

The final constraints ensure that each chain includes an NDD. These are very similar to the generalized subtour elimination constraints in the TSP literature.

- Constraint 12j: for every subset S of the donor-patient vertices, each vertex in S can only participate in a chain if there is chain flow into S.

The number of constraints in 12j grows exponentially with the number of patient-donor vertices, so it is necessary to use constraint generation with the PC-TSP formulation. We avoid constraint generation by developing a new formulation, which draws on concepts of both PC-TSP and PICEF; this formulation is introduced in the following section.

B.2 Our PI-TSP Formulation

In this section we present the new position-indexed PC-TSP formulation (PI-TSP), which combines concepts from both the PC-TSP formulation and the PICEF formulation. The main advantage of our approach is in the formulation of chains. PC-TSP uses a fixed number of decision variables to allow long (or uncapped) chains, but requires constraint generation. PICEF does not require constraint generation, but the number of decision variables grows polynomially with the chain cap.

Our approach achieves the best of both worlds: PI-TSP uses a *fixed* number of decision variables for any chain cap, and does not require constraint generation. To our knowledge, ours is the first formulation to exhibit this behavior.

PI-TSP uses the same parameters as PICEF and PC-TSP:

- G: kidney exchange graph, consisting of edges $e \in E$ and vertices $v \in V = P \cup N$, including patient-donor pairs P and NDDs N.
- C: a set of cycles on exchange graph G.
- L: chain cap (maximum number of edges used in a chain).
- w_e: edge weights for each edge $e \in E$.
- w_c^C: cycle weights for each cycle $c \in C$, defined as $w_c^C = \sum_{e \in c} w_e$.

PI-TSP also uses the same decision variables (and auxiliary variables) as PC-TSP. Two additional variables are added to the formulation: $p_e, p_v \geq 1$ for each edge $e \in E$ and patient-donor vertex $v \in P$, to represent e and v's position in a chain.

- $p_e \geq 1$: the position of edge e in any chain.
- $p_v \geq 1$: the position of patient-donor vertex v in any chain (equal to the position of any incoming edge).
- $\hat{p}_e \geq 0$: equal to p_e if e is used in a chain, and 0 otherwise. (i.e. $\hat{p}_e = p_e \cdot y_e$)
- $z_c \in \{0, 1\}$: 1 if cycle c is used in the matching, and 0 otherwise.
- $y_e \in \{0, 1\}$: 1 if edge e is used in a chain, and 0 otherwise.
- $y_e^n \in \{0, 1\}$: 1 if edge e is used in a chain starting with NDD n, and 0 otherwise.
- w_n^N (auxiliary): total weight of the chain starting with NDD n.
- f_v^i and f_v^o (auxiliary): chain flow into and out of vertex $v \in P$, respectively.
- $f_v^{i,n}$ and $f_v^{i,n}$ (auxiliary): chain flow into and out of vertex $v \in P$, respectively, from a chain beginning with NDD $n \in N$.

The PI-TSP formulation with chain cap L is given in Problem 13. As before, we use the notation $\delta^-(v)$ for the set of edges into vertex v and $\delta^+(v)$ for the set of edges out of v.

$$\max \quad \sum_{n \in N} w_n^N + \sum_{c \in C} w_c^C z_c \tag{13a}$$

$$\text{s.t.} \quad \sum_{e \in E} w_e y_e^n = w_n^N \qquad n \in N \tag{13b}$$

$$\sum_{n \in N} y_e^n = y_e \qquad e \in E \tag{13c}$$

$$\sum_{e \in \delta^-(v)} y_e = f_v^i \qquad v \in V \tag{13d}$$

$$\sum_{e\in\delta^+(v)} y_e = f_v^o \qquad v \in V \tag{13e}$$

$$\sum_{e\in\delta^-(v)} y_e^n = f_v^{i,n} \qquad v \in V, n \in N \tag{13f}$$

$$\sum_{e\in\delta^+(v)} y_e^n = f_v^{o,n} \qquad v \in V, n \in N \tag{13g}$$

$$f_v^o + \sum_{c\in C: v\in c} z_c \leq \quad f_v^i + \sum_{c\in C: v\in c} z_c \leq 1 \qquad v \in P \tag{13h}$$

$$f_v^o \leq 1 \qquad v \in N \tag{13i}$$

$$p_e = 1 \qquad e \in \delta^+(N) \tag{13j}$$

$$\hat{p}_e = p_e y_e \qquad e \in E \tag{13k}$$

$$p_v = \sum_{e\in\delta^-(v)} \hat{p}_e \qquad v \in P \tag{13l}$$

$$p_e = p_v + 1 \qquad v \in P, e \in \delta^+(v) \tag{13m}$$

$$\sum_{e\in E} y_e^n \leq L \qquad n \in N \tag{13n}$$

$$f_v^{o,n} \leq f^{i,v} \leq 1 \qquad v \in V, n \in N \tag{13o}$$

$$y_e \in \{0,1\} \qquad e \in E \tag{13p}$$

$$z_c \in \{0,1\} \qquad c \in C \tag{13q}$$

$$y_e^n \in \{0,1\} \qquad e \in E, n \in N \tag{13r}$$

All constraints are identical to those of PC-TSP, but without the subtour elimination constraints 12j, and with the addition of the following constraints:

- Constraints 13j: sets $p_e = 1$ for all edges out of NDD vertices.
- Constraints 13k: defines $\hat{p}_e$.
- Constraints 13l: for all vertices v, sets p_v equal to the variable p_e of *any* incoming edge.
- Constraints 13m: for all outgoing edges of all vertices v, sets $p_e = p_v + 1$.

Two adjustments may be made to this formulation: first, the variables p_v are not necessary, but are useful for illustration. We can remove these variables by combining Constraints 13l and 13m as follows:

$$p_{\overline{e}} = 1 + \sum_{e\in\delta^-(v)} \hat{p}_e \qquad v \in P, \overline{e} \in \delta^+(v)$$

Second, Constraints 13k are nonlinear; we linearize these by replacing 13k with the following constraints for each $e \in E$:

$$\hat{p}_e \leq y_e M$$
$$\hat{p}_e \leq p_e$$
$$p_e - (1 - y_e)M \leq \hat{p}_e$$

Experiments: Minimum Chain Length We demonstrate the utility of the PI-TSP formulation by finding optimal matchings with a *minimum* chain length (L_{min}). We set a *maximum* chain length of $L_{max} = 3$, and vary the L_{min} from 0 to 3. For some exchange graph, let $|M_{OPT}|$ be the score of the *optimal* matching (i.e. with no minimum chain length, and maximum chain length 3); we calculate the fractional optimality gap for the matching M_l (with score $|M_l|$), which has minimum chain length $L_{min} = l$. We define $\Delta OPT(M_l)$ as

$$\Delta OPT(M_l) = \frac{|M_l| - |M_{OPT}|}{|M_{OPT}|}$$

We calculate optimal matchings for $L_{min} = 0, 1, 2, 3$, for each of the UNOS exchange graphs used in Section 5. Only 154 of the roughly 300 UNOS graphs contain chains; the remaining graphs may have no NDDs, or the NDDs may have no feasible donors. Focusing on these 154 graphs, we calculate ΔOPT and the chain lengths of each optimal matching, for each L_{min}. Figure 4 shows histograms of ΔOPT and the chain lengths for all optimal matchings, for each $L_{min} = 0, 1, 2, 3$. Note that ΔOPT is zero for $L_{min} = 0$, by definition.

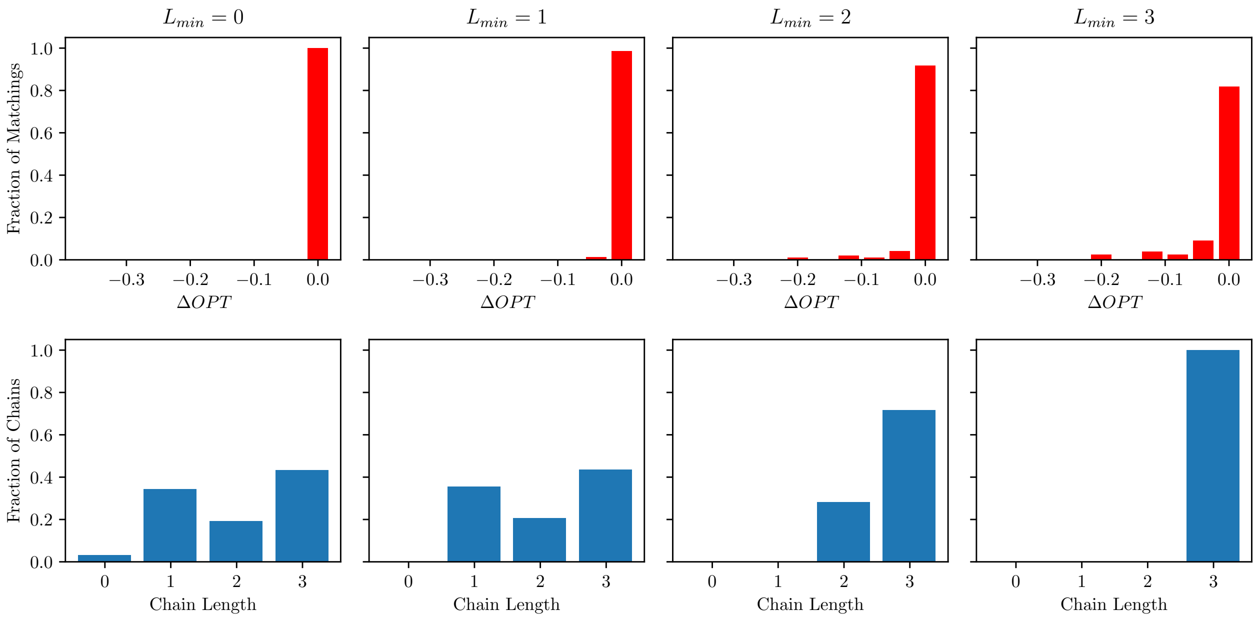

Figure 4: ΔOPT (top row) and chain lengths (bottom row) for the optimal matchings with minimum chain length L_{min}, and maximum chain length of 3.

For some of these exchanges, a minimum chain length of 2 or 3 was infeasible (58 for $L_{min} = 2$, and 77 for $L_{min} = 3$, out of 154 total exchanges); we do not consider these cases.

As expected, enforcing $L_{min} > 0$ results in longer chains – such that when $L_{min} = L_{max} = 3$, *all* chains have length 3. Surprisingly, enforcing a minimum chain length does not impact the overall matching score. Indeed, even when $L_{min} = 3$, 70% of all matchings have a *zero optimality gap*. However these experiments did not consider edge failures. As discussed in Section 4, edge failures impact long cycles and chains more than short cycles and chains; in practice, when edges have a nonzero failure probability, setting a high L_{min} makes the matching more susceptible to failure (i.e. less robust).

B.3 Edge Existence Robust Formulation

In this section we develop a mixed integer linear program formulation for the edge existence robust kidney exchange ($KEX(\mathcal{U}^w_\Gamma)$). This problem maximizes the matching score while minimizing the objective with respect to realized cycle and chain weights $\hat{\mathbf{w}}^{\mathbf{M}}_{\mathbf{c}}$ for the current matching M. We develop an edge-existence robust formulation by directly minimizing the PI-TSP Objective (13a) over all cycle and chain weight realizations in $\mathcal{U}^w_\Gamma$. For brevity, let $\mathcal{M}^P$ be the set of all possible feasible solutions to the PI-TSP formulation; we represent these feasible solutions as $(\mathbf{y}, \mathbf{z}) \in \mathcal{M}$, where $\mathbf{y}$ are the edge decision variables for chains, and $\mathbf{z}$ are the cycle decision variables.

For any feasible solution $(\mathbf{y}, \mathbf{z}) \in \mathcal{M}^P$, we find the minimum objective value for any realized cycle and chain weights in $\mathcal{U}^w_\Gamma$ With some abuse of notation, this minimum is represented by the function $Z(\mathbf{y}, \mathbf{z})$. In this section we separate the realized weights $\hat{\mathbf{w}}_{\mathbf{c}}$ into the realized cycle weights $\hat{\mathbf{w}}^{\mathbf{C}}_{\mathbf{c}}$ and the realized chain weights $\hat{\mathbf{w}}^{\mathbf{N}}_{\mathbf{c}}$.

$$Z(\mathbf{y}, \mathbf{z}) = \min_{(\hat{\mathbf{w}}^{\mathbf{C}}_{\mathbf{c}}, \hat{\mathbf{w}}^{\mathbf{N}}_{\mathbf{c}}) \in \mathcal{U}^w_\Gamma} \sum_{n \in N} \hat{w}^N_n + \sum_{c \in C} \hat{w}^C_c z_c$$

Note that maximizing $Z(\mathbf{y}, \mathbf{z})$ is equivalent to solving $KEX(\mathcal{U}^w_\Gamma)$ – the robust kidney exchange problem with uncertainty set $KEX(\mathcal{U}^w_\Gamma)$. The following lemma states that this is equivalent to solving the constant-budget edge existence robust kidney exchange problem $KEX(\mathcal{U}^E_\Gamma)$.

Lemma 1. *$KEX(\mathcal{U}^E_\Gamma)$ is equivalent to $KEX(\mathcal{U}^w_\Gamma)$*

Proof. Consider a feasible matching $M = (z_c, y_e)$. The only difference between $KEX(\mathcal{U}^E_\Gamma)$ and $KEX(\mathcal{U}^w_\Gamma)$ is the minimization of the objective over uncertainty sets $\mathcal{U}^E_\Gamma$ and $\mathcal{U}^w_\Gamma$ respectively.

Problem $KEX(\mathcal{U}^E_\Gamma)$ minimizes the matching weight over edge subsets $\hat{E} \subseteq E$, where $R = E \setminus \hat{E}$ contains up to Γ edges:

- If $\Gamma = 1$, the largest decrease in matching weight occurs if the *highest weight* cycle or chain is discounted – that is, if R contains the first edge in the highest weight chain, or any edge in the highest weight cycle.

- Similarly if $\Gamma = 2$, the largest decrease in matching weight occurs when the two highest-weight cycles and chains are discounted.

Thus, for any positive Γ and any feasible matching M, the minimum objective in $KEX(\mathcal{U}^E_\Gamma)$ occurs when the Γ highest-weight cycles and chains in M are discounted.

In $KEX(\mathcal{U}^w_\Gamma)$, for any Γ and any feasible matching M, the minimum occurs (trivially) when the Γ highest-weight cycles or chains are discounted in $\mathcal{U}^w_\Gamma$.

For any matching M, minimizing the KEX objective over $\mathcal{U}^E$ and $\mathcal{U}^w_\Gamma$ produce the same outcome – the Γ highest-weight cycles and chains are discounted. Thus, the minimization in $KEX(\mathcal{U}^E)$ and $KEX(\mathcal{U}^w_\Gamma)$ is equivalent. □

Thus, to solve the constant-budget edge existence robust kidney exchange problem, we can solve Problem (14) – which maximizes $Z(\mathbf{y}, \mathbf{z})$ over all feasible matchings $(\mathbf{y}, \mathbf{z}) \in \mathcal{M}^P$.

$$\max \quad Z(\mathbf{y}, \mathbf{z}) \tag{14a}$$

$$(\mathbf{y}, \mathbf{z}) \in \mathcal{M}^P \tag{14b}$$

We proceed by solving Problem (14), which is equivalent to $KEX(\mathcal{U}^w_\Gamma)$. To solve this problem we first develop a linear formulation for Z using the PC-TSP decision variables, and then we maximize this linear expression.

B.4 Linear Formulation for Z

In this section we minimize the function Z for any matching $(\mathbf{y}, \mathbf{z}) \in \mathcal{M}^P$, within uncertainty set $\mathcal{U}^w_\Gamma$. Within this uncertainty set, up to $\lfloor \Gamma \rfloor$ cycles and chains can have zero realized weight (i.e. $\hat{w}_c = 0$), and if Γ is not integer, then one cycle or chain will have realized weight equal to the fraction $(\Gamma - \lfloor \Gamma \rfloor)$ of its total nominal weight (i.e. $\hat{w}_c = (\Gamma - \lfloor \Gamma \rfloor) w_c$. We say that any cycle or chain c with $\hat{w}_c < w_c$ is *discounted*.

First note that if a matching uses G cycles and chains, and $G < \Gamma$, only G objects are discounted. Thus let $\Gamma' = \min\{G, \Gamma\}$ be the number of discounted cycles and chains, i.e.,

$$G = \sum_{c \in C} z_c + \sum_{n \in N} \sum_{e \in \delta^+(n)} y_e.$$

To linearize the definition of Γ', we introduce variable h, which is 1 if $G < \Gamma$ and 0 otherwise. The statement $\Gamma' = \min\{G, \Gamma\}$ is linearized using the following constraints:

$$\begin{aligned} \Gamma - G &\leq Wh \\ G - \Gamma &\leq W(1-h) \\ G - Wh &\leq \Gamma' \\ \Gamma - W(1-h) &\leq \Gamma' \\ h &\in \{0, 1\} \end{aligned}$$

where W is a large constant.

The function Z is minimized w.r.t. the realized weights, when the Γ' discounted cycles and chains are those with the *largest* weight. To select these objects we introduce variables $g^C_c, g^N_n \in \{0, 1\}$ for each cycle $c \in C$ and each chain's NDD $n \in N$. For any matching, let m be the smallest weight of any discounted cycle or chain – that is, m is the $\lceil \Gamma' \rceil^{th}$ highest weight of any cycle or chain used in the matching. We define g^C_c and g^N_n as follows

$$g^C_c = \begin{cases} 1 & \text{if } w^C_c \geq m \\ 0 & \text{otherwise} \end{cases} \qquad g^N_n = \begin{cases} 1 & \text{if } w^N_n \geq m \\ 0 & \text{otherwise} \end{cases}$$

Thus $g^C_c = 1$ or $g^N_n = 1$ implies that cycle c or chain n should be discounted if used in the matching. We define these variables using linear constraints, in two steps. First, we require that $g^{\{C,N\}}_j = 1$ only if $g^{\{C,N\}}_k = 1$ for all cycles and chains k with weight larger than $w^{\{C,N\}}_j$. That is, we require that variables $g^{\{C,N\}}$ obey the same ordering as $w^{\{C,N\}}$. This ordering requirement can be defined using the following correspondences

$$g^C_i \geq g^C_j \Leftrightarrow w^C_i \geq w^C_j, \qquad i, j \in C \tag{15}$$

$$g^C_c > g^N_n \Leftrightarrow w^C_c > w^N_n, \qquad c \in C, n \in N \tag{16}$$

$$g^C_c \leq g^N_n \Leftrightarrow w^C_c \leq w^N_n, \qquad c \in C, n \in N \tag{17}$$

$$g^N_i \geq g^N_j \Leftrightarrow w^N_i \geq w^N_j, \qquad i, j \in N \tag{18}$$

Note that cycle weights are fixed but chain weights depend on the decision variables. Thus we determine ordering relation 15 by sorting all cycle weights during preprocessing, and enforcing this ordering over g_i^C using the relation $\geq_C$, defined as

$$\geq_C = \left\{ (g_a^C, g_b^C) \in \mathbf{g}^C \times \mathbf{g}^C \mid w_a^C \geq w_b^C \right\}.$$

Using this notation, the ordering relation $\geq_C$ contains all pairs of cycles (a, b) such that $w_a^C \geq w_b^C$. For simplicity, I will denote this ordering relation as

$$a \geq_C b.$$

This ordering relation is enforced on variables g_i^C using $(|C| - 1)$ constraints. The ordering required by correspondence 16, 17, and 18 depend on the chain weights, which in turn depend on decision variables. We can linearize these correspondences using the following inequalities

$$\begin{aligned} g_c^C + q_{cn} &\geq g_n^N \\ W(1 - q_{cn}) &\geq w_c^C - w_n^N \\ \\ g_n^N + (1 - q_{cn}) &\geq g_c^C \\ W q_{cn} &\geq w_n^N - w_c^C \\ \\ & q_{cn} \in \{0, 1\}, c \in C, n \in N, \end{aligned}$$

Where W is a large constant. When $w_c^C > w_n^N$, this forces q_{cn} to be 0; as a result, the inequality $g_c^C \geq g_n^N$ must hold. Otherwise, if $w_c^C < w_n^N$, this forces q_{cn} to be 1, which forces the inequality $g_n^N \geq g_c^C$ to hold.

Similarly, the following constraints enforce the ordering in correspondence 18 over variables g_n^N

$$\begin{aligned} g_i^N + q_{ij}^N &\geq g_j^N \\ W(1 - q_{ij}^N) &\geq w_i^N - w_j^N \\ \\ g_j^N + (1 - q_{ij}^N) &\geq g_i^N \\ W q_{ij}^N &\geq w_j^N - w_i^N \\ \\ & q_{ij}^N \in \{0, 1\}, i, j \in N, i \neq j \end{aligned}$$

Next we require that only Γ' objects are discounted. Note that if cycle c is discounted if $g_c^C w_c^C = 1$, and chain n is discounted if $g_n^N \sum\limits_{e \in E} y_e^n = 1$. Thus, the following identity requires that exactly Γ objects are discounted:

$$\sum_{c \in C} z_c g_c^C + \sum_{n \in N} g_n^N \sum_{e \in \delta^+(n)} y_e = \Gamma'$$

We use these variables to directly minimize $Z(\mathbf{y}, \mathbf{z})$ w.r.t. $\mathcal{U}_\Gamma^w$, and the result is given in Equation (19).

$$Z(\mathbf{y}, \mathbf{z}) = \sum_{n \in N} w_n^N + \sum_{c \in C} w_c z_c - \sum_{n \in N} g_n^N w_n^N - \sum_{c \in C} g_c^C w_c z_c \tag{19a}$$

$$\text{s.t.} \quad \Gamma - G \leq Wh \tag{19b}$$

$$G - \Gamma \leq W(1 - h) \tag{19c}$$

$$G - Wh \leq \Gamma' \tag{19d}$$

$$\Gamma - W(1 - h) \leq \Gamma' \tag{19e}$$

$$\left.\begin{aligned} g_c^C + q_{cn} &\geq g_n^N \\ W(1 - q_{cn}) &\geq w_c^C - w_n^N \\ g_n^N + (1 - q_{cn}) &\geq g_c^C \\ W q_{cn} &\geq w_n^N - w_c^C \end{aligned}\right. \quad , c \in C, n \in N \tag{19f}$$

$$\begin{array}{rl} g_i^N + q_{ij}^N & \geq g_j^N \\ W(1-q_{ij}^N) & \geq w_i^N - w_j^N \end{array} \qquad , q_{ij}^N \in \{0,1\}, i,j \in N, i \neq j \tag{19g}$$

$$\sum_{c \in C} z_c g_c^C + \sum_{n \in N} g_n^N \sum_{e \in \delta^+(n)} y_e = \Gamma' \tag{19h}$$

$$g_i^C \geq_C g_j^C \qquad , i,j \in C, i \neq j \tag{19i}$$

$$q_{cn} \in \{0,1\}, \qquad c \in C, n \in N, \tag{19j}$$

$$h \in \{0,1\} \tag{19k}$$

Note that there are two sets of quadratic expressions in this formulation: $g_c^C z_c$, and $w_n^N g_n^N$. These are linearized in the next section, which addresses non-integer Γ.

B.5 Non-Integer Γ

When Γ is not integer, the actual number of discounted cycles and chains (Γ') may be integer or non-integer valued. When Γ' is not integer valued, up to $\lfloor \Gamma' \rfloor$ cycles and chains are fully discounted (i.e. $\hat{w}_c = 0$), and the smallest-weight cycle or chain is discounted by fraction $(\Gamma - \lfloor \Gamma \rfloor)$. We include this fractional discount by using two sets of indicator variables $f_i^{\{C,N\}}$ and $p_i^{\{C,N\}}$ for all cycles and chains $i \in C \cup N$, and then discount each i as follows:

- i is fully discounted if $p_i^{\{C,N\}} = f_i^{\{C,N\}} = 1$.
- i is partially discounted fraction $(\Gamma - \lfloor \Gamma \rfloor)$ if $f_i^{\{C,N\}} = 0$ and $p_i^{\{C,N\}} = 1$
- i is not discounted if $f_i^{\{C,N\}} = p_i^{\{C,N\}} = 0$.

Thus if Γ' is integer, $f_i^{\{C,N\}} = p_i^{\{C,N\}}$ for all cycles and chains i; if Γ' is not integer, then $\lceil \Gamma' \rceil$ cycles and chains are least partially discounted ($p_e^{\{C,N\}} = 1$), and $\lfloor \Gamma' \rfloor$ cycles and chains are fully discounted ($p_i^{\{C,N\}} = g_i^f = 1$). These indicator variables are defined in the same way as $g_i^{\{C,N\}}$ in Equation (19): $p_i^{\{C,N\}}$, $f_i^{\{C,N\}} \in \{0,1\}$, and they obey the same ordering relation as the cycle and chain weights. However, the number of cycles and chains with $f_i^{\{C,N\}} = 1$ can be different than the number of cycles and chains with with $p_i^{\{C,N\}} = 1$. Thus we add new constraints for each of these variables.

Setting the number of discounted cycles and chains. First we require $\lceil \Gamma' \rceil$ cycles and chains have $p_i^{\{C,N\}} = 1$. Recall that G is the number of matching edges, and $\Gamma' = \min(\Gamma, G)$; if $\Gamma < G$, then $\lceil \Gamma' \rceil = \lceil \Gamma \rceil$, and $\lceil \Gamma' \rceil = G$ otherwise. The variable h is defined to be 1 if $G < \Gamma$ and 0 otherwise. Thus, the following constraint requires that $\lceil \Gamma' \rceil$ cycles and chains have $p_i^{\{C,N\}} = 1$:

$$\sum_{n \in N} p_n^N \sum_{e \in \delta^+(n)} y_e + \sum_{c \in C} p_c^C z_c = hG + (1-h)\lceil \Gamma \rceil.$$

Similarly, the following constraint requires that $\lfloor \Gamma' \rfloor$ cycles and chains have $f_i^{\{C,N\}} = 1$:

$$\sum_{n \in N} f_n^N \sum_{e \in \delta^+(n)} y_e + \sum_{c \in C} f_c^C z_c = hG + (1-h)\lfloor \Gamma \rfloor.$$

Thus if $G < \Gamma$, then all G cycles and chains will have $f_i^{\{C,N\}} = p_i^{\{C,N\}} = 1$; otherwise, there are $\lceil \Gamma \rceil$ cycles and chains with $p_i^{\{C,N\}} = 1$, and $\lfloor \Gamma \rfloor$ cycles and chains with $f_i^{\{C,N\}} = 1$, where the partially-discounted cycle or chain has $f_i^{\{C,N\}} = 0$ and $p_i^{\{C,N\}} = 1$.

Ordering relation over indicator variables. To enforce the ordering relation over indicator variables f_n^N, p_n^N, f_c^C, and p_c^C, we use constraints similar to those used in the edge weight robust formulation. The auxiliary variables q_{cn} and q_{ij}^N are defined the same way here: q_{cn} is 0 when $w_n^C > w_n^N$ and 1 otherwise; q_{ij}^N is 0 if

$$\begin{array}{rl} f_c^C + q_{cn} & \geq f_n^N \\ p_c^C + q_{cn} & \geq p_n^N \\ f_n^N + (1-q_{cn}) & \geq f_c^C \\ p_n^N + (1-q_{cn}) & \geq p_c^C \\ W(1-q_{cn}) & \geq w_c^C - w_n^N \\ Wq_{cn} & \geq w_n^N - w_c^C \end{array} \qquad , c \in C, n \in N,$$

$$q_{cn} \in \{0,1\}, c \in C, n \in N,$$

Where W is a large constant. When $w_c^C > w_n^N$, this forces q_{cn} to be 0; as a result, the inequality $f_c^C \geq f_n^N$ and $p_c^C \geq p_n^N$ must hold. Otherwise, if $w_c^C < w_n^N$, this forces q_{cn} to be 1, which forces the inequality $f_n^N \geq f_c^C$ and $p_n^N \geq p_c^C$ to hold.

Similarly, the following constraints enforce the ordering in correspondence 18 over variables f_n^N and p_n^N

$$\left.\begin{array}{rl} f_i^N + q_{ij}^N & \geq f_j^N \\ p_i^N + q_{ij}^N & \geq p_j^N \\ f_j^N + (1 - q_{ij}^N) & \geq f_i^N \\ p_j^N + (1 - q_{ij}^N) & \geq p_i^N \\ W q_{ij}^N & \geq w_j^N - w_i^N \\ W(1 - q_{ij}^N) & \geq w_i^N - w_j^N \end{array}\right. , i, j \in N, i \neq j$$

$$q_{ij}^N \in \{0,1\}, i, j \in N, i \neq j$$

As before, correspondence 15, the ordering between cycle indicator variables, is enforced using the pre-determined ordering $\geq_C$.

$$\left.\begin{array}{rl} f_a^C & \geq_C f_b^C \\ p_a^C & \geq_C p_b^C \end{array}\right. , a, b \in C, a \neq b$$

Objective for non-integer Γ. Using these indicator variables, the new objective of the robust formulation is

$$\max \sum_{n \in N} w_n^N + \sum_{c \in C} w_c z_c - (1 - \Gamma + \lfloor \Gamma \rfloor) \left(\sum_{n \in N} w_n^N f_n^N + \sum_{c \in C} f_c^C w_c z_c \right)$$
$$- (\Gamma - \lfloor \Gamma \rfloor) \left(\sum_{n \in N} w_n^N p_n^N + \sum_{c \in C} p_c^C w_c z_c \right)$$

which discounts cycle or chain i by its full weight if $f_i^{\{C,N\}} = p_i^{\{C,N\}} = 1$, and by fraction $(\Gamma - \lfloor \Gamma \rfloor)$ of its weight if $f_i^{\{C,N\}} = 0$ and $p_i^{\{C,N\}} = 1$.

Non-linear terms. There are now 7 types of nonlinear terms in this formulation:

- hG,
- $w_n^N f_n^N$,
- $w_n^N p_n^N$,
- $z_c f_c^C$,
- $z_c p_c^C$,
- $f_n^N y_e$, and
- $p_n^N y_e$.

First we linearize the chain-related quadratic terms by introducing the variables $\hat{f}_n^N \equiv w_n^N f_n^N$ and $\hat{p}_n^N \equiv w_n^N p_n^N$. The following constraints define these new variables, using a large constant W.

$$\left.\begin{array}{rl} \hat{f}_n^N & \leq f_n^N W \\ \hat{f}_n^N & \leq w_n^N \\ \hat{f}_n^N & \geq w_n^N - (1 - f_n^N) W \end{array}\right. , n \in N$$

$$\hat{f}_n^N \geq 0, n \in N$$

$$\left.\begin{array}{ll} \hat{p}^N_n & \leq p^N_n W \\ \hat{p}^N_n & \leq w^N_n \\ \hat{p}^N_n & \geq w^N_n - (1-p^N_n)W \end{array}\right. , n \in N$$

$$\hat{p}^N_n \geq 0, n \in N$$

Next we define variables $\hat{f}^C_c \equiv z_c f^C_c$ and $\hat{p}^C_c \equiv z_c p^C_c$ using the following constraints.

$$\left.\begin{array}{ll} \hat{f}^C_c & \leq f^C_c \\ \hat{f}^C_c & \leq z_c \\ \hat{f}^C_c & \geq f^C_c + z_c - 1 \end{array}\right. , c \in C$$

$$\hat{f}^C_c \in \{0,1\}, c \in C$$

$$\left.\begin{array}{ll} \hat{p}^C_c & \leq p^C_c \\ \hat{p}^C_c & \leq z_c \\ \hat{p}^C_c & \geq p^C_c + z_c - 1 \end{array}\right. , c \in C$$

$$\hat{p}^C_c \in \{0,1\}, c \in C$$

To linearize the term hG, we introduce variable $\hat{g} \equiv hG$, which is defined using the following constraints. As before, W is a large constant.

$$\hat{h} \leq hW$$

$$\hat{h} \leq G$$

$$\hat{h} \geq G - (1-h)W$$

$$\hat{h} \geq 0$$

Finally, we introduce the variables $F_n \equiv f^N_n \sum\limits_{e \in \delta^+(n)} y_e$ and $P_n \equiv p^N_n \sum\limits_{e \in \delta^+(n)} y_e$, defined with the following constraints. Note that for each NDD $n \in N$ the sum of all y_e variables is either zero (if n does not initiate a chain) or 1 (if n initiates a chain). Thus F_n and P_n are products of binary variables, which we define using the following constraints.

$$\left.\begin{array}{ll} F_n & \leq f^N_n \\ F_n & \leq \sum\limits_{e \in \delta^+(n)} y_e \\ F_n & \geq \sum\limits_{e \in \delta^+(n)} y_e + f^N_n - 1 \end{array}\right. , c \in C$$

$$F_n \in \{0,1\}, n \in N$$

$$\left.\begin{array}{ll} P_n & \leq p^N_n \\ P_n & \leq \sum\limits_{e \in \delta^+(n)} y_e \\ P_n & \geq \sum\limits_{e \in \delta^+(n)} y_e + p^N_n - 1 \end{array}\right. , n \in N$$

$$P_n \in \{0,1\}, n \in N$$

Linear formulation. Finally, for any feasible matching we directly minimize Z by discounting the Γ' largest-weight cycles and chains. This is accomplished using the variables $\hat{f}^N_n$, $\hat{p}^N_n$, $\hat{f}^C_c$, $\hat{p}^C_c$. Equation (20) gives the minimization of Z for any matching $(\mathbf{y}, \mathbf{z})$, using only linear constraints.

$$\begin{aligned} Z(\mathbf{y},\mathbf{z}) = \quad & \sum_{n \in N} w^N_n + \sum_{c \in C} w_c z_c - (1 - \Gamma + \lfloor \Gamma \rfloor) \left(\sum_{n \in N} \hat{f}^N_n + \sum_{c \in C} \hat{f}^C_c w_c \right) \\ & -(\Gamma - \lfloor \Gamma \rfloor) \left(\sum_{n \in N} \hat{p}^N_n + \sum_{c \in C} \hat{p}^C_c w_c \right) \end{aligned} \tag{20a}$$

s.t.

$$\begin{aligned} \Gamma - G &\leq Wh \\ G - \Gamma &\leq W(1-h) \\ G - Wh &\leq \Gamma' \\ \Gamma - W(1-h) &\leq \Gamma' \end{aligned} \tag{20b}$$

$$\sum_{n \in N} P_n + \sum_{c \in C} \hat{p}_c^C = \hat{h} + (1-h)\lceil \Gamma \rceil \tag{20c}$$

$$\sum_{n \in N} F_n + \sum_{c \in C} \hat{f}_c^C = \hat{h} + (1-h)\lfloor \Gamma \rfloor \tag{20d}$$

$$\left.\begin{aligned} f_c^C + q_{cn} &\geq f_n^N \\ p_c^C + q_{cn} &\geq p_n^N \\ f_n^N + (1-q_{cn}) &\geq f_c^C \\ p_n^N + (1-q_{cn}) &\geq p_c^C \\ W(1-q_{cn}) &\geq w_c^C - w_n^N \\ Wq_{cn} &\geq w_n^N - w_c^C \end{aligned}\right. \quad , \quad c \in C, n \in N \tag{20e}$$

$$\left.\begin{aligned} f_i^N + q_{ij}^N &\geq f_j^N \\ p_i^N + q_{ij}^N &\geq p_j^N \\ f_j^N + (1-q_{ij}^N) &\geq f_i^N \\ p_j^N + (1-q_{ij}^N) &\geq p_i^N \\ Wq_{ij}^N &\geq w_j^N - w_i^N \\ W(1-q_{ij}^N) &\geq w_i^N - w_j^N \end{aligned}\right. \quad , \quad i, j \in N, i \neq j \tag{20f}$$

$$\left.\begin{aligned} f_a^C &\geq_C f_b^C \\ p_a^C &\geq_C p_b^C \end{aligned}\right. \quad , \quad a, b \in C, a \neq b \tag{20g}$$

$$\left.\begin{aligned} \hat{f}_n^N &\leq f_n^N W \\ \hat{f}_n^N &\leq w_n^N \\ \hat{f}_n^N &\geq w_n^N - (1-f_n^N)W \end{aligned}\right. \quad , \quad n \in N \tag{20h}$$

$$\left.\begin{aligned} \hat{p}_n^N &\leq p_n^N W \\ \hat{p}_n^N &\leq w_n^N \\ \hat{p}_n^N &\geq w_n^N - (1-p_n^N)W \end{aligned}\right. \quad , \quad n \in N \tag{20i}$$

$$\left.\begin{aligned} \hat{f}_c^C &\leq f_c^C \\ \hat{f}_c^C &\leq z_c \\ \hat{f}_c^C &\geq f_c^C + z_c - 1 \end{aligned}\right. \quad , \quad c \in C \tag{20j}$$

$$\left.\begin{aligned} \hat{p}_c^C &\leq p_c^C \\ \hat{p}_c^C &\leq z_c \\ \hat{p}_c^C &\geq p_c^C + z_c - 1 \end{aligned}\right. \quad , \quad c \in C \tag{20k}$$

$$\begin{aligned} \hat{h} &\leq hW \\ \hat{h} &\leq G \\ \hat{h} &\geq G - (1-h)W \\ \hat{h} &\geq 0 \end{aligned} \tag{20l}$$

$$\left.\begin{aligned} F_n &\leq f_n^N \\ F_n &\leq \sum_{e \in \delta^+(n)} y_e \\ F_n &\geq \sum_{e \in \delta^+(n)} y_e + f_n^N - 1 \end{aligned}\right. , c \in C \tag{20m}$$

$$\left.\begin{aligned} P_n &\leq p_n^N \\ P_n &\leq \sum_{e \in \delta^+(n)} y_e \\ P_n &\geq \sum_{e \in \delta^+(n)} y_e + p_n^N - 1 \end{aligned}\right. , n \in N \tag{20n}$$

$$\hat{f}_c^C, \hat{p}_c^C \in \{0,1\}, \quad c \in C \tag{20o}$$

$$\hat{f}_n^N, \hat{p}_n^N \geq 0, \quad n \in N \tag{20p}$$

$$F_n \in \{0,1\}, n \in N \tag{20q}$$

$$P_n \in \{0,1\}, n \in N \tag{20r}$$

$$q_{ij}^N \in \{0,1\}, \quad i,j \in N, i \neq j \tag{20s}$$

$$q_{cn} \in \{0,1\}, \quad c \in C, n \in N \tag{20t}$$

$$h \in \{0,1\} \tag{20u}$$

The linear formulation for $KEX(\mathcal{U}_\Gamma^w)$ is obtained by adding the PI-TSP constraints to Problem (20), and mazimizing the objective 20a.

This linear formulation can be solved by any standard solver; our experiments use Gurobi (Gurobi Optimization, Inc. 2018).

C Robustness as Fairness

In this section we use the framework of edge weight uncertainty to address the problem of fairness in kidney exchange. Though seemingly unrelated, fairness and uncertainty share some key characteristics. The concept of *budgeted uncertainty* balances the nominal objective value with the worst case. A similar trade-off exists between fairness and efficiency in kidney exchange: allocating kidneys to hard-to-match patients is *fair*, but often reduces the number of possible transplants.

C.1 The Price of Fairness

In kidney exchange, fairness often pertains to *highly-sensitized* patients, who are very unlikely to find a compatible donor. Highly-sensitized patients face longer waiting times than lowly-sensitized patients[5]. In part this is because highly sensitized patients are hard to match; for this reason most kidney exchange optimization algorithms – which maximize matching size or weight – marginalize highly-sensitized patients.

A patient's sensitization level is measured by her Calculated Panel Reactive Antibody (CPRA) score, which ranges from 0 to 100. Patient-donor pair vertices in the exchange graph are highly-sensitized if the pair's patient has a CPRA score above some threshold τ, which is set by policymakers ($\tau = 80$ is common). Let V_H (V_L) be the set of highly-sensitized (lowly-sensitized) vertices in P, and let E_H (E_L) be the set of all edges that end in V_H (V_L).

Fairness for a matching M is often quantified using the *utility* assigned to V_H and V_L – i.e. the sum of edge weights into each vertex set,

$$U_H(M) = \sum_{e \in E_H} x_e w_e, \qquad U_L(M) = \sum_{e \in E_L} x_e w_e.$$

The *utilitarian* utility function is defined as $u(M) = U_H(M) + U_L(M)$ (i.e. the total edge weight of matching M). We might define a *fair* utility function $u_f : \mathcal{M} \to \mathbb{R}$, such that the matching M_f^* that maximizes u_f is considered fair:

$$M_f^* = \arg\max_{M \in \mathcal{M}} u_f(M)$$

Fairness is quantified using the *fraction of the fair score* $\%F : M, \mathcal{M} \to [0,1]$ – i.e. the fraction of the maximum possible utility awarded to highly sensitized patients

$$\%F(M, \mathcal{M}) = U_H(M) / \max_{M' \in \mathcal{M}} U_H(M').$$

Bertsimas *et al.* (2011b) defines the *price of fairness* as the "relative system efficiency loss under a fair allocation assuming that a fully efficient allocation is one that maximizes the sum of [participant] utilities." Thus the price of fairness is defined using the set of matchings $\mathcal{M}$, the fair utility function u_f, and the utilitarian utility function u:

$$\text{POF}(\mathcal{M}, u_f) = \frac{u\left(M^*\right) - u\left(M_f^*\right)}{u\left(M^*\right)} \tag{21}$$

POF$(\mathcal{M}, u_f)$ is the relative loss in (utilitarian) efficiency caused by choosing the fair outcome M_f^* rather than the most efficient outcome.

Balancing $\%F$ and POF is a key problem in kidney exchange. Achieving a high degree of fairness (high $\%F$) often incurs a high POF; on the other hand, requiring a low POF ofen results in low $\%F$. Dickerson *et al.* (2014) propose two rules for enforcing fairness in kidney exchange, and demonstrate that without chains, the price of fairness is low in theory. McElfresh and Dickerson (2018) extended this result, finding that adding chains lowers the theoretical price of fairness – eventually to zero; they also propose a fairness rule that limits the price of fairness.

In the next section we generalize one of the fairness rule proposed by Dickerson *et al.* (2014) using the framework of budgeted robust optimization, and demonstrate its versatility in balancing fairness and efficiency.

[5] `https://optn.transplant.hrsa.gov/data/`

C.2 Fairness Through Robustness

In this section we adapt the concept of budgeted uncertainty to apply budgeted *prioritization* to highly sensitized patients in kidney exchange. To prioritize certain patients over others, we assign each edge $e \in E$ a ***priority weight*** $\hat{w}_e \in [0, \infty)$, equal to the nominal weight multiplied by a factor $(1 + \alpha_e)$, with $\alpha_e \in [-1, \infty)$. There are many ways to prioritize highly sensitized vertices using priority weights: we may set $\alpha > 0$ for all edges in E_H, or we may set $\alpha = -1$ for edges in E_L, and so on.

To balance fairness with efficiency it reasonable to *limit* the degree of prioritization. To limit prioritization, we define a *budgeted prioritization set* $\mathcal{P}$, which bounds the sum of absolute differences between each w_e and $\hat{w}_e$; this prioritization set is given in Equation (22).

$$\mathcal{P}_\Gamma = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = w_e(1 + \alpha_e), \alpha_e \in [-1, \infty], \sum_{e \in E} \alpha_e w_e \le \Gamma \right\} \tag{22}$$

To prioritize V_H, we define α_e differently for each edge e. In one type of approach, we prioritize V_H by setting α_e to a constant (α) for all $e \in E_H$. This approach is given by $\mathcal{P}_\Gamma^+$, in Equation (23)

$$\mathcal{P}_\Gamma^+ = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = \begin{cases} w_e(1 + \alpha) & \text{if } e \in E_H \\ w_e & \text{otherwise} \end{cases}, \alpha \ge 0, \alpha \sum_{e \in E} w_e \le \Gamma \right\} \tag{23}$$

A different type of approach prioritizes V_H by reducing all edges into E_L; this approach is given by $\mathcal{P}_\Gamma^-$, in Equation (24).

$$\mathcal{P}_\Gamma^- = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = \begin{cases} w_e(1 - \alpha) & \text{if } e \in E_L \\ w_e & \text{otherwise} \end{cases}, \alpha \in [0, 1], \alpha \sum_{e \in E} w_e \le \Gamma \right\} \tag{24}$$

To apply prioritization to kidney exchange, we either minimize or maximize the kidney exchange objective with respect to $\mathcal{P}$. By choosing α_e and prioritization budget Γ, this general framework can implement a wide variety of prioritization requirements. Next we show how budgeted prioritization generalizes a previous fairness rule.

C.3 Weighted Fairness

Weighted fairness was proposed by Dickerson *et al.* (2014) to prioritize highly sensitized patients in kidney exchange. This fairness rule maximizes the total matching weight, after multiplying all edge weights into highly sensitized patients by a factor $(1 + \gamma)$, where parameter γ is set by policymakers. Weighted fairness is equivalent to *maximizing* the kidney exchange objective over the budgeted prioritization set $\mathcal{P}^w$, given below. This prioritization set is equivalent to $\mathcal{P}_\Gamma^+$, with prioritization budget Γ equal to γ times the total weight received by highly sensitized patients.

$$\mathcal{P}_\gamma^w = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = \begin{cases} w_e(1 + \alpha) & \text{if } e \in E_H \\ w_e & \text{otherwise} \end{cases}, \alpha \ge 0, \alpha \sum_{e \in E_H} w_e \le \gamma \sum_{e \in E_H} w_e \right\} \tag{25}$$

Note that the uncertainty budget does not depend on edge weights, and can be written succinctly as Equation (26).

$$\mathcal{P}_\gamma^w = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = \begin{cases} w_e(1 + \alpha) & \text{if } e \in E_H \\ w_e & \text{otherwise} \end{cases}, 0 \le \alpha \le \gamma \right\} \tag{26}$$

Weighted fairness is implemented by *maximizing* over priority set $\mathcal{P}_\gamma^w$, as in Problem (27)

$$\max \max_{\hat{w} \in \mathcal{P}_\gamma^w} \hat{w} \cdot x_e \tag{27a}$$

$$\mathbf{x} \in \mathcal{M} \tag{27b}$$

Proposition 3. *γ-weighted fairness is equivalent to maximizing the kidney exchange objective over $\mathcal{P}_\gamma^w$.*

As demonstrated in Equation (25), weighted fairness uses the prioritization budget $\Gamma = \gamma \sum\limits_{e \in E_H} w_e$, which is proportional to the weight received by highly sensitized patients. Thus, we may derive an upper bound on the POF for γ-weighted fairness.

Proposition 4. *For γ-weighted fairness, and some matching M the price of fairness for choosing matching M is bounded above by*

$$\mathit{POF}(u_\gamma^w, M) \le \frac{\gamma}{1 + \gamma + U_L(M)/U_H(M)}.$$

Proof. Suppose that γ-weighted fairness chooses matching M over a higher-weight matching E. In the worst case, both F and E receive nearly the same *priority weight* under γ-weighted fairness (within a small perturbation ϵ). Let the utility awarded by each outcome to highly- and lowly-sensitized patients be given by

$$\begin{aligned} U_H(M) &= A & U_L(M) &= B \\ U_H(E) &= 0 & U_L(E) &= A(1+\gamma) + B - \epsilon \end{aligned}$$

with $0 < \epsilon \ll 1$. Both M and E receive nearly the same priority weight from γ-weighted fairness, but E receives γA more weight than M:

$$u^w_\gamma(M) = A(1+\gamma) + B$$
$$u^w_\gamma(E) = A(1+\gamma) + B - \epsilon$$

And thus γ-weighted fairness selects M over E. Taking the limit as $\epsilon \to 0$, the price of fairness for choosing M is

$$\text{POF}(u^w_\gamma, M) = \frac{A(1+\gamma) + B - \epsilon - A - B}{A(1+\gamma) + B - \epsilon} = \frac{\gamma}{1+\gamma+B/A},$$

note that $A = U_H(M)$ and $B = U_L(M)$, and thus

$$\text{POF}(u^w_\gamma, M) = \frac{\gamma}{1+\gamma+U_L(M)/U_H(M)}.$$

Note that this is the worst-case POF for choosing M, and thus

$$\text{POF}(u^w_\gamma, M) \leq \frac{\gamma}{1+\gamma+U_L(M)/U_H(M)}.$$

□

It follows that this POF is maximized when $U_L(M) = 0$, which is the worst case POF for γ-weighted fairness.

Corollary 1. *For γ-weighted fairness, the price of fairness is bounded above by*

$$POF(u^w_\gamma) \leq \frac{\gamma}{1+\gamma}$$

Proposition 5. *Let U^*_H be the maximum possible utility for highly-sensitized patients. For γ-weighted fairness, and some matching M the fraction of the fair score $\%F$ for matching M is bounded below by*

$$\%F(M, \mathcal{M}) \geq 1 - \frac{U_L(M)}{U^*_H}\frac{1}{1+\gamma}.$$

Proof. Let $M \in mM$ be a feasible matching, and let U^*_H be the maximum possible utility for highly-sensitized patients. Consider the worst case scenario for γ-weighted fairness: two outcomes receive nearly equal utility from γ-weighted fairness, but the outcome chosen is far less fair. Let the *fair* outcome F assign the maximum possible utility to highly sensitized patients, and zero utility to lowly sensitized patients:

$$u_H(F) = U^*_H, \qquad u_L(F) = 0.$$

Let M be the outcome selected by γ-weighted fairness, which assigns utility $\beta U_H(M)$ to highly sensitized patients, with $0 < \beta < 1$, and some utility $A + \epsilon$ to lowly sensitized patients, with $0 < \epsilon \ll 1$:

$$u_H(M) = \beta U^*_H, \qquad u_L(M) = A + \epsilon,$$

and note that β is $\%F$, the fraction of the fair score, for outcome M.

Letting $\epsilon \to 0$, both F and M receive the same utility under γ-weighted fairness; that is,

$$U^*_H(1+\gamma) = \beta U^*_H(1+\gamma) + U_L(M).$$

Rearranging, we have

$$\%F(M, \mathcal{M}) = \beta = 1 - \frac{U_L(M)}{U^*_H}\frac{1}{1+\gamma}.$$

This is the worst-case outcome for $\%F$, and thus

$$\%F(M, \mathcal{M}) \geq \beta = 1 - \frac{U_L(M)}{U^*_H}\frac{1}{1+\gamma}.$$

□

It follows that the worst-case $\%F$ occurs when U_L is maximal, and $M = M^*$.

Corollary 2. *Let U_H^* and U_L^* be the maximum possible utility for highly- and lowly-sensitized patients, respectively. Under γ-weighted fairness, the fraction of the fair score $\%F$ is bounded below by*

$$\%F(*, \mathcal{M}) \geq 1 - \frac{U_L^*}{U_H^*}\frac{1}{1+\gamma}.$$

These results may be used to balance $\%F$ and POF, subject to policymaker requirements. For example, suppose policymakers require that $\%F \geq f$, and POF $\leq p$, for some constants f and p. If we know the maximum utility for highly- and lowly-sensitized patients, we can bound γ using the worst-case bounds from Corollary 1 and 2. Inverting the bounds from these Corollaries with $p =$ POF and $f = \%f$, we have

$$\gamma \leq \frac{p}{1-p}, \qquad \gamma \geq \frac{U_L^*}{U_H^*}\frac{1}{1-f} - 1.$$

Combining these restrictions, we arrive at the bounded prioritization set $\mathcal{P}_p^f$, given in Equation (28).

$$\mathcal{P}_p^f = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = \begin{cases} w_e(1+\gamma) & \text{if } e \in E_H \\ w_e & \text{otherwise} \end{cases}, \frac{U_L^*}{U_H^*}\frac{1}{1-f} - 1 \leq \gamma \leq \frac{p}{1-p} \right\} \tag{28}$$

There are two important observations about this prioritization set. First, not all choices of f and p are valid, and this depends on U_L^*/U_H^*; that is, choosing either f or p necessarily bounds the other. Second, there are many ways to use $\mathcal{P}_p^f$ in practice: we might *minimize* or *maximize* $\hat{\mathbf{w}}$ before maximizing the kidney exchange objective (i.e., setting γ to its maximum or minimum value; this is equivalent to the weighted fairness proposed by Dickerson *et al.* (2014).

Alternatively, we might allow γ to vary within the range of set by $\mathcal{P}_p^f$. This approach allows the optimization algorithm to *choose* the value of γ, such that priority weight is maximized. Note that this is not equivalent to weighted fairness (Problem (27)), which maximizes priority weight *before* maximizing the objective. This variable-γ approach is given in Problem (29).

$$\max_{\hat{w} \in \mathcal{P}_p^f} \sum_{e \in E} \hat{w}_e \cdot x_e$$
$$\mathbf{x} \in \mathcal{M}$$

By directly applying the definition of $\hat{w}$ to this problem, we arrive at Problem (30).

$$\max \quad (1+\gamma)\sum_{e \in E_H} w_e \cdot x_e + \sum_{e \in E_L} w_e \cdot x_e \tag{30a}$$
$$\frac{U_L^*}{U_H^*}\frac{1}{1-f} - 1 \leq \gamma \leq \frac{p}{1-p} \tag{30b}$$
$$\mathbf{x} \in \mathcal{M} \tag{30c}$$

In the next section we tighten this the bound on $\%F$ for γ-weighted fairness, by relaxing the bounds on γ.

Variable Weighted Fairness The bounds in Corollary 1 and 2 are for the *worst-case* bounds on γ; however, the worst-case scenarios that produce these bounds may never occur. Instead, we use Proposition 4 and 5 to bound γ for some feasible matching M.

As before, suppose that policymakers require that $\%F \geq f$, and POF $\leq p$, for some constants f and p. If we know the maximum utility for highly-sensitized patients, we can bound γ (for some matching M) using the worst-case bounds from Proposition 4 and 5. Inverting these bounds with $p =$ POF and $f = \%f$, we have

$$\gamma \leq \frac{p}{1-p}\left(1 + \frac{U_L(M)}{U_H(M)}\right), \qquad \gamma \geq \frac{U_L(M)}{U_H^*}\frac{1}{1-f} - 1.$$

Applying these bounds on γ results in the following prioritization set $\mathcal{P}_p^f$, given in Equation (31).

$$\mathcal{P}_p^f = \left\{ \hat{\mathbf{w}} \mid \hat{w}_e = \begin{cases} w_e(1+\gamma) & \text{if } e \in E_H \\ w_e & \text{otherwise} \end{cases}, \frac{U_L(M)}{U_H^*}\frac{1}{1-f} - 1 \leq \gamma \leq \frac{p}{1-p}\left(1 + \frac{U_L(M)}{U_H(M)}\right) \right\} \tag{31}$$

As before, we might maximize or minimize the prioritization weight over $\mathcal{P}_f^p$ (i.e., weighted fairness), or allow γ to vary within the range of $\mathcal{P}_f^p$. Note that allowing γ to vary adds variable inequalities, which depends on the decision variables of M.