

# Small Representations of Big Kidney Exchange Graphs

John P. Dickerson<sup>\*,+</sup> and Aleksandr M. Kazachkov<sup>+</sup> and Ariel Procaccia<sup>+</sup> and Tuomas Sandholm<sup>+</sup>

University of Maryland\* Carnegie Mellon University<sup>+</sup>

john@cs.umd.edu, {arielpro,sandholm}@cs.cmu.edu, akazachk@cmu.edu

## Abstract

Kidney exchanges are organized markets where patients swap willing but incompatible donors. In the last decade, kidney exchanges grew from small and regional to large and national—and soon, international. This growth results in more lives saved, but exacerbates the empirical hardness of the  $\mathcal{NP}$ -complete problem of optimally matching patients to donors. State-of-the-art matching engines use integer programming techniques to clear fielded kidney exchanges, but these methods must be tailored to specific models and objective functions, and may fail to scale to larger exchanges. In this paper, we observe that if the kidney exchange compatibility graph can be encoded by a constant number of patient and donor attributes, the clearing problem is solvable in polynomial time. We give necessary and sufficient conditions for losslessly shrinking the representation of an arbitrary compatibility graph. Then, using real compatibility graphs from the UNOS US-wide kidney exchange, we show how many attributes are needed to encode real graphs. The experiments show that, indeed, small numbers of attributes suffice.

## 1 Introduction

There are over 100,000 needy patients waiting for a kidney transplant in the United States, with similar—and increasing—demand worldwide.<sup>1</sup> Complementing potential cadaveric transplantation via the deceased donor waiting list, a recent innovation—kidney exchange (Rapaport 1986; Roth, Sönmez, and Ünver 2004)—allows patients with willing *living* donors to participate in cyclic donor swaps or altruist-initiated donation chains to receive a life-saving organ. Kidney exchange now accounts for over 10% of living donation in the US, with that percentage increasing annually.

In reality, participating patients and donors are endowed with a set of attributes: blood type, tissue type, age, insurance, home transplant center, willingness to travel, and myriad other measurements of health, personal preference, and logistical constraints. While some of these features can, at a cost, be temporarily or permanently changed, the attributes determine the feasibility of a potential donation from each donor to each patient. For example, a donor with blood type AB can only give to a patient with that blood type.

A central aspect of kidney exchange is the *clearing problem*, that is, determining the “best” set of cyclic and chain-based swaps to perform in a given *compatibility graph*, which consists of all participating patients, donors, and their potential feasible transactions. For even simple (but realistic) models of kidney exchange, the clearing problem is  $\mathcal{NP}$ -hard (Abraham, Blum, and Sandholm 2007; Biró, Manlove, and Rizzi 2009) and also extremely difficult to solve in practice (Glorie, van de Klundert, and Wagelmans 2014; Anderson et al. 2015; Dickerson et al. 2016).

In this paper, we tackle the complexity of the clearing problem via the introduction of a novel model for kidney exchange that explicitly takes into account all attributes of the participating patients and donors. Under the assumption that real kidney exchange graphs can be represented using a constant number of attributes, we show that our model permits polynomial-time solutions to central  $\mathcal{NP}$ -hard problems in general kidney exchange. Inspired by classical results from intersection graph theory, we give conditions on the representation of arbitrary graphs in our model, and generalize to the case where participants are allowed to have a thresholded number of negative interactions between attributes. Noting that real-life kidney exchange graphs are *not* arbitrary, we show on actual data from the United Network for Organ Sharing (UNOS) US-wide kidney exchange that our model permits lossless representation of true graphs with far fewer attributes than the worst-case theoretical results require.

## 2 A New Model for Kidney Exchange

In this section, we formalize our model of kidney exchange. We prove that under this model certain well-known  $\mathcal{NP}$ -hard problems in general kidney exchange are solvable in polynomial time. We also show that, given a compatibility graph, determining the best set of attributes to change (at some cost) is solvable in polynomial time.

### 2.1 Notation & Preliminaries

A kidney exchange can be represented by a directed *compatibility graph*  $G = (V, E)$ . Each patient-donor pair, or unpaired altruistic donor, forms a vertex  $v \in V$ , and a directed edge exists from one vertex to another if the donor at the former can give to the patient at the latter, i.e., are compatible (Roth, Sönmez, and Ünver 2004; 2005).

In kidney exchange, patients and donors participate in *cycles* or *chains*. In a cycle, each participating vertex receives the kidney of the previous vertex. All transplants in a cycle must be performed simultaneously to ensure participation, and thus are limited to some small length in practice. This ensures that no donor backs out after her patient has received a kidney but before she has donated her kidney. Most fielded kidney exchanges—including UNOS—allow only 2- and 3-cycles. In a chain, a donor *without a paired patient* enters the pool, donating his kidney to a patient, whose paired donor donates his kidney to another patient, and so on (Montgomery et al. 2006; Roth et al. 2006; Rees et al. 2009). Chains can be executed non-simultaneously<sup>2</sup> and thus chains can be longer (but typically not infinite) in length. Most exchanges—including UNOS—see great gains through the use of such “altruist-initiated” chains.

We consider a model that imposes additional structure on an arbitrary compatibility graph. For each vertex  $v_i \in V$ , associate attribute vectors  $\mathbf{d}_i$  and  $\mathbf{p}_i$  with its constituent donor and patient, respectively. The  $q$ th element  $d_i^q$  of  $\mathbf{d}_i$  takes on one of a fixed number of types—for example, one of four blood types (O, A, B, AB), or one of a few hundred standard insurance plans. Then, for  $v_i \neq v_j \in V$ , we define a *compatibility function*  $f(\mathbf{d}_i, \mathbf{p}_j)$ , a boolean function that returns the compatibility of the donor of  $v_i$  with the patient of  $v_j$ .

Given  $V$  and associated attribute vectors, we can uniquely determine a compatibility graph  $G = (V, E)$  such that  $E = \{(v_i, v_j) : f(\mathbf{d}_i, \mathbf{p}_j) = 1 \ \forall v_i \neq v_j \in V\}$ . We claim that this model accurately mimics reality, and we later support that claim with strong experimental results on real-world data. Furthermore, under this new model, certain complexity results central to kidney exchange change (for the better), as we discuss next.

## 2.2 The Clearing Problem is Easy (in Theory)

We now tackle the central computational challenge of kidney exchange: the clearing problem. Well-known to be  $\mathcal{NP}$ -hard (Abraham, Blum, and Sandholm 2007; Biró, Manlove, and Rizzi 2009), a variety of custom clearing algorithms address adaptations of the clearing problem in practice.<sup>3</sup> We show that, in our model, the clearing problem itself is solvable in polynomial time.

Formally, we are interested in a polynomial-time algorithm that solves the  $L$ -CYCLE-COVER problem—that is, finding the largest disjoint packing of cycles of length at most  $L$ . For ease of exposition, in this section we use “cycles” to refer to both cycles and chains; indeed, it is easy to see that altruist donors are equivalent to standard patient-donor pairs with a patient who is compatible with all non-altruist vertices in the pool. Then, a chain is equivalent to a cycle with a “dummy” edge returning to the altruist. Also,

<sup>2</sup>To see why this is, take the case where a donor backs out of a chain after his paired patient received a kidney, but before his own donation. Unlike in the case of a broken cycle, no pair in the remaining tail of the planned chain is strictly worse off; that is, no donor was “used up” before her paired patient received a kidney.

<sup>3</sup>For an overview of practical approaches to solving the clearing problem, see a recent survey due to Mak-Hau (2015).

again for ease of exposition, we assume the value of a chain of length  $L$  is equal to a cycle of length  $L$ , due to the final donor giving to a patient on the deceased donor waiting list.

Recall that we are working in a model where each vertex  $v_i$  belongs to one of a *fixed* number of types determined solely by its attribute vectors  $\mathbf{d}_i$  and  $\mathbf{p}_i$ . Let  $\Theta$  be the set of all possible types, and  $\theta \in \Theta$  represent one such individual type. With a slight abuse of notation, we can define a type compatibility function  $f(\theta, \theta') = 1$  if and only if there is a directed edge between vertices of type  $\theta$  and  $\theta'$ . (Note that this captures chains and altruist donors as described above.)

A key observation of this section is that any additional edge structure that is imposed on the graph—such as a cycle cover—would be independent of the identity of *specific* vertices; rather, it would only depend on their types, as vertices of the same type have the exact same incoming and outgoing neighborhoods. For example, in any cycle cover, if  $v_i$  and  $v_j$  are two vertices of the same type, we can swap  $v_i$  and  $v_j$  and obtain a feasible cycle cover of the same size. This observation drives our theoretical algorithmic results.

In more detail, every cycle through vertices of  $G$  can be interpreted as a closed walk through the type space. Every such cycle can be represented by  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_\ell) \in \Theta^\ell$ , where  $\ell$  is the length of the cycle. Let us define  $f_C$  as the boolean function with  $f_C(\boldsymbol{\theta}) = 1$  if and only if

$$f(\theta_1, \theta_2) = \dots = f(\theta_{\ell-1}, \theta_\ell) = f(\theta_\ell, \theta_1) = 1.$$

Furthermore, for  $L \leq n = |V|$ , let  $\mathcal{T}(L)$  denote the set of closed walks through the type space of length at most  $L$ . Formally

$$\mathcal{T}(L) = \bigcup_{\ell=2}^L \{\boldsymbol{\theta} \in \Theta^\ell : f_C(\boldsymbol{\theta}) = 1\}.$$

Let  $\mathcal{C}$  be a cycle cover in  $G$ , and denote the number of unique vertices matched in  $\mathcal{C}$  by  $\|\mathcal{C}\|_V$ . Suppose  $\mathcal{C}$  has cycle cap  $L$ ; then it is equivalent, in our setting, to a vector  $\mathbf{m} \in \mathbb{Z}_+^{|\mathcal{T}(L)|}$ , where, for  $\boldsymbol{\theta} \in \mathcal{T}(L)$ ,  $m_\theta$  equals the number of cycles in  $\mathcal{C}$  that can be represented in the type space by  $\boldsymbol{\theta}$ . Let  $|\boldsymbol{\theta}|$  be the length of the vector  $\boldsymbol{\theta}$ , and  $\|\mathbf{m}\|_V = \sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m_\theta |\boldsymbol{\theta}|$ . Then  $\|\mathbf{m}\|_V = \|\mathcal{C}\|_V$ , that is,  $\|\mathbf{m}\|_V$  is another way to express the number of matched vertices in the equivalent cycle cover.

Now consider Algorithm 1 for  $L$ -CYCLE-COVER in our model, which we claim is optimal and computationally efficient in our setting.

---

### Algorithm 1 $L$ -CYCLE-COVER

---

1.  $\mathcal{C}^* \leftarrow \emptyset$
  2. **for** every vector of non-negative integers  $\mathbf{m} \in \mathbb{Z}_+^{|\mathcal{T}(L)|}$  such that  $\|\mathbf{m}\|_V \leq n$ 
    - **if**  $\|\mathbf{m}\|_V > \|\mathcal{C}^*\|_V$  and there exists cycle cover  $\mathcal{C}$  in  $G$  such that for each  $\boldsymbol{\theta} \in \mathcal{T}(L)$ ,  $\mathcal{C}$  contains  $m_\theta$  cycles of type  $\boldsymbol{\theta}$ , **then**  $\mathcal{C}^* \leftarrow \mathcal{C}$
  3. **return**  $\mathcal{C}^*$
- 

**Theorem 1.** *Given constants  $L$  and  $|\Theta|$ , Algorithm 1 is a polynomial-time algorithm for  $L$ -CYCLE-COVER.*

*Proof.* We start by verifying that Algorithm 1 is indeed optimal. Consider the optimal cycle cover  $C^*$ . For each  $\theta \in \mathcal{T}(L)$ , let  $m_\theta^*$  be the number of cycles in  $C^*$  that are consistent with the types in  $\theta$ . Clearly  $\sum_{\theta \in \mathcal{T}(L)} m_\theta^* |\theta| \leq n$ , as there are only  $n$  vertices. Therefore, Algorithm 1 considers the collection of numbers  $m_\theta^*$  in Step 2. Because this collection of numbers does induce a valid cycle cover that is of the same size as  $C^*$ , the algorithm would update its incumbent cycle cover if it were not already optimal.

We next analyze the running time of the algorithm. First, note that it is straightforward to check whether the vector  $\mathbf{m}$  induces a valid cycle cover. Since  $\mathcal{T}(L)$  consists only of valid cycles according to the compatibility function  $f_C$ , we just need to check that there are enough vertices of each type  $\theta \in \Theta$  to construct all the cycles that require them. This can be checked individually for each  $\theta \in \Theta$ . For a particular  $\theta \in \mathcal{T}(L)$ , the number of vertices of type  $\theta$  required is  $m_\theta$  multiplied by the number of times type  $\theta$  appears in  $\theta$ . Then the sum of these products over all  $\theta \in \mathcal{T}(L)$  is at most the number of vertices of type  $\theta$  in  $G$ .

Second, there is only a polynomial number of possibilities to construct a collection of numbers  $\mathbf{m} = \{m_\theta\}_{\theta \in \mathcal{T}(L)}$  such that  $\|\mathbf{m}\|_V \leq n$ . Indeed, this number is at most  $(n+1)^{|\mathcal{T}(L)|}$ . Moreover,  $|\mathcal{T}(L)| \leq L \cdot |\Theta|^L$ . Because  $|\Theta|$  and  $L$  are constants,  $|\mathcal{T}(L)|$  is also a constant. The expression  $(n+1)^{|\mathcal{T}(L)|}$  is therefore a polynomial in  $n$ .  $\square$

Even for constant  $L$ , the running time of Algorithm 1 is exponential in  $|\Theta|$ . But this is to be expected. Indeed, any graph can trivially be represented using a set  $\Theta$  of types of size  $n$ , where each vertex has a unique type, and a compatibility function  $f_C$  that assigns 1 to an ordered pair of types if the corresponding edge exists in  $G$ . Therefore, if the running time of Algorithm 1 were polynomial in  $n$  and  $|\Theta|$ , we would solve the general  $L$ -CYCLE-COVER problem in polynomial time—and that problem is  $\mathcal{NP}$ -hard (Abraham, Blum, and Sandholm 2007).

### 2.3 Flipping Attributes is Also Easy (in Theory)

While patients and donors in a kidney exchange are endowed with an initial set of attributes, it may be possible in practice to—at a cost—change some number of those attributes to effect change in the final matching. For example, the human body naturally tries to reject, to varying degrees, a transplanted organ. Due to this, nearly all recipients of kidneys are placed on immunosuppressant drugs after transplantation occurs.<sup>4</sup> However, *preoperative* immunosuppression can also be performed to increase transplant opportunity—but at some cost to the patient’s overall health.

With this in mind, we extend the model of Section 2.2 as follows. Associate with each pair of types  $\theta, \theta' \in \Theta$  a cost function  $c : \Theta \times \Theta \rightarrow \mathbb{R}$  representing the cost of changing a vertex of type  $\theta$  to type  $\theta'$ . Then, the  $L$ -FLIP-AND-CYCLE-COVER problem is to find a disjoint packing of cycles of length at most  $L$  that maximizes the size of the packing minus the sum of costs spent changing types. Building on Theorem 1, this problem is also solvable in polynomial time.

**Theorem 2.** *Suppose that  $L$  and  $|\Theta|$  are constants. Then  $L$ -FLIP-AND-CYCLE-COVER is solvable in polynomial-time.*

*Proof sketch.* For any type  $\theta \in \Theta$ , there are  $n_\theta$  vertices. Then, for each of the  $(|\Theta| - 1)$  choices of which type  $\theta' \neq \theta$  to switch to, choose how many vertices from  $\theta$  will switch to that type; there are at most  $(n_\theta + 1)$  possibilities. Do this for all types in  $\Theta$ , resulting in

$$\prod_{\theta \in \Theta} (n_\theta + 1)^{|\Theta| - 1} \leq (n + 1)^{|\Theta|^2}$$

possibilities. Since  $|\Theta|$  is a constant, this is polynomial in  $n$ .

For each of these polynomially-many type-switch possibilities, we can compute the optimal cycle cover in polynomial time using Algorithm 1, and subtract  $c(\theta, \theta')$  for each vertex that switched from  $\theta$  to  $\theta'$ . Taking the best of these solutions gives the optimal solution in polynomial time.  $\square$

## 3 A Concrete Instantiation: Thresholding

As motivated in Sections 1 and 2, compatibility in real kidney exchange graphs is determined by patient and donor attributes, such as blood or tissue type. In particular, if an attribute for a donor and patient is in conflict, they are deemed incompatible. Motivated by that reality, in this section, we associate with each patient and donor a bit vector of length  $k$ , and count incompatibilities based on any shared activated bits between a patient and potential donor.

As a concrete example, consider human blood types. At a high level, human blood contains A antigens, B antigens, both (type AB), or neither (type O). AB-type patients can receive from any donor, A-type (B-type) can receive from O-type and A-type (B-type) donors, and O-type patients can only receive from O-type donors. In our bit model, this is represented with  $k = 2$ . The first bit represents compatibility with A antigens and the second bit represents compatibility with B antigens. Thus, the type space  $\Theta = 2^{\{\text{has-A, has-B}\}} \times 2^{\{\text{no-A, no-B}\}}$ ; in general,  $|\Theta| = 2^{2k}$ .

Formally, unless otherwise stated, throughout this section  $G$  will refer to a directed graph with vertex set  $V = [n] := \{1, \dots, n\}$  and edge set  $E$ , and with each  $i \in V$  associated with two  $k$ -bit vectors  $\mathbf{d}_i, \mathbf{p}_i \in \{0, 1\}^k$ . Let  $Q_d(i) = \{q \in [k] : \mathbf{d}_{iq} = 1\}$  be the set of conflict bits for the donor associated with vertex  $i \in V$ , and similarly let  $Q_p(i) = \{q \in [k] : \mathbf{p}_{iq} = 1\}$ . For  $i, j \in V$  such that  $i \neq j$ , the threshold feasibility function  $f_{\text{thresh}}^t$  is defined as

$$f_{\text{thresh}}^t(\mathbf{d}_i, \mathbf{p}_j) = \begin{cases} 1 & \text{if } |Q_d(i) \cap Q_p(j)| \leq t, \\ 0 & \text{otherwise.} \end{cases}$$

Note that  $|Q_d(i) \cap Q_p(j)| \leq t$  if and only if  $\langle \mathbf{d}_i, \mathbf{p}_j \rangle \leq t$ .

Kidney exchange graphs constructed using threshold compatibility functions are closely related to complements of *intersection graphs* (McKee and McMorris 1999), which are graphs that have a set associated with each vertex and an edge between two vertices if and only if the sets intersect. Given  $t \in \mathbb{N}$ , the function  $f_{\text{thresh}}^t$  is related to *p-intersection graphs* (Chung and West 1994; Eaton, Gould, and Rödl 1996), where an edge connects two vertices if their corresponding sets intersect in at least  $p \geq 1$  elements.

<sup>4</sup><https://www.kidney.org/atoz/content/immuno>

In particular, our model is similar to that of *intersection digraphs* (Sen et al. 1989), or equivalently *bipartite intersection graphs* (Harary, Kabell, and McMorris 1982), both also considered in (Orlin 1977). These have mainly been studied under the assumption that the sets used to represent the graph have the “consecutive ones” property, i.e., each set is an interval from the set of integers. Our model is more general: we do not place such an assumption on the set of conflict bits. Moreover, most treatments of intersection digraphs consider loops on the vertices, whereas in our thresholding model, whether or not donor  $i$  and patient  $i$  are compatible is not considered. In addition, the directed and bipartite intersection graph literature has focused on the case that  $t = 0$  (in our terminology). To the best of our knowledge, this paper is the first treatment *p-intersection digraphs*, and certainly their first real-world application.

### 3.1 Existence of Small Representations

It is natural to ask for what values of  $t$  and  $k$  we can select vertices with bit vectors  $\mathbf{d}_i$  and  $\mathbf{p}_i$  of length  $k$  such that  $f_{\text{thresh}}^t$  can create *any* graph of a specific size?

Formally, we say that  $G$  is  $(k, t)$ -representable (by feasibility function  $f_{\text{thresh}}^t$ ) if, for all  $i \in V$  there exist  $\mathbf{d}_i, \mathbf{p}_i \in \{0, 1\}^k$  such that for all  $j_1 \in V, j_2 \in V \setminus \{j_1\}, (j_1, j_2) \in E$  if and only if  $f_{\text{thresh}}^t(\mathbf{d}_{j_1}, \mathbf{p}_{j_2}) = 1$ .

It is known (Erdős, Goodman, and Pósa 1966) that any graph can be represented as an intersection graph with  $k \leq n^2/4$ . Yet, we show next that, in our model,  $k \leq n$  suffices to represent any graph. It is akin to a result on the *term rank* of the adjacency matrix of  $\bar{G}$  (Orlin 1977, Thm 6.6).

**Theorem 3.** *Let  $G = (V, E)$  be a digraph on  $n$  vertices. Let  $n_1$  be the number of vertices with outgoing edges, let  $n_2$  be the number of vertices with incoming edges, and  $n' = \min\{n_1 + 1, n_2 + 1, n\}$ . Then  $G$  can be  $(n', 0)$ -represented.*

*Proof.* We first show that the graph can be  $(n_1 + 1, 0)$ -represented. Assume without loss of generality that vertices  $1, \dots, n_1$  have outgoing edges. We show how to set  $\mathbf{d}_i, \mathbf{p}_i \in \{0, 1\}^{n_1+1}$  for each vertex  $i$  in  $V$ . To set the donor attributes, for each  $i \in [n_1]$ , let  $\mathbf{d}_i$  be  $e_i$ , the  $i$ th standard basis vector, i.e., the vector of length  $n_1 + 1$  with a 1 in the  $i$ th coordinate and 0 everywhere else. For  $i > n_1$ , set  $\mathbf{d}_i$  to be  $e_{n_1+1}$ . For the patient attributes of vertex  $j \in [n]$ , for each  $i \in [n_1]$  such that  $(i, j) \in E$ , set  $\mathbf{p}_{ji} = 0$ , and set  $\mathbf{p}_{ji} = 1$  otherwise. Note that if all the vertices have outgoing edges, then  $n_1 = n$  unit vectors suffice. A similar approach works to  $(\min\{n, n_2 + 1\}, 0)$ -represent  $G$ , by using the  $n_2$  unit vectors as the patient vectors of those vertices with incoming edges, and (if needed) one additional unit vector for any remaining vertices. In both of these cases,  $\langle \mathbf{d}_i, \mathbf{p}_j \rangle = 0$  if and only if  $(i, j) \in E$ , which represents  $G$  by  $f_{\text{thresh}}^0$ .  $\square$

In particular, Theorem 3 implies that any graph is  $(n, 0)$ -representable. The next theorem shows a matching lower bound. The same construction and bound also hold if loops are considered (Sen et al. 1989).

**Theorem 4.** *For any  $n \geq 3$ , there exists a graph on  $n$  vertices that is not  $(k, 0)$ -representable for all  $k < n$ .*

*Proof.* Define  $G$  to be the digraph on  $n$  vertices,  $V = [n]$ , with an edge from vertex  $i$ , for each  $i \in V$ , to every vertex except  $i - 1$  (and itself), where vertex  $n$  is also referred to as vertex 0.

Assume that  $G$  is  $(k, 0)$ -representable, and consider vertex 1. Since  $(1, n) \notin E$ , and  $(i, n) \in E$  for all  $i \notin \{1, n\}$ , there exists a conflict bit  $q_1 \in Q_d(1) \cap Q_p(n)$  such that  $q_1 \notin Q_p(V \setminus \{1, n\})$ . More generally, there exists such a conflict bit  $q_i$  for all  $i \in V$ .

We claim that these conflict bits are all unique, which directly implies that  $k \geq n$ . Indeed, otherwise we can assume that  $q_1 = q_i$  for some  $i \neq 1$  (without loss of generality, as the graph is symmetric subject to cyclic permutations). But then  $(1, i - 1)$  and  $(i, n)$  do not appear as edges in  $G$ , which is not true for any  $i \in V \setminus \{1\}$ .  $\square$

More generally, it is easy to see that any graph that is  $(k, 0)$ -representable is also  $(k + t, t)$ -representable for any  $t \geq 0$ . Indeed, simply take the  $(k, 0)$ -representation of the graph, and append  $t$  ones to every vector. Together with Theorem 3, this shows that any graph is  $(n + t, t)$ -representable. However, the lower bound given by Theorem 4 does not extend to  $t > 0$ . We conjecture that for any  $n$  and  $t$  there exists a graph that can only be  $(k, t)$ -represented with  $k = \Omega(n)$ —this remains an open question.

### 3.2 Computational Issues

Given a real compatibility graph with  $n$  vertices, we know by Theorem 3 that we can  $(k, 0)$ -represent that graph for  $k = n$ . But, in practice, how large of a  $k$  is actually needed?

Various problems related to intersection graphs are  $\mathcal{NP}$ -complete for general graphs (Kou, Stockmeyer, and Wong 1978; Orlin 1977), but we work in a setting with additional structure. And while we do not show that finding a  $(k, t)$ -representation is  $\mathcal{NP}$ -hard, we do show that a slightly harder problem, which we refer to as  $(k, t)$ -REPRESENTATION-WITH-IGNORED-EDGES, is  $\mathcal{NP}$ -hard. Given an input of a directed graph  $G = (V, E)$ , a subset  $F$  of  $\binom{V}{2}$ , and integers  $k \geq 1$  and  $t \geq 0$ , this problem asks whether there exist bit vectors  $\mathbf{d}_i$  and  $\mathbf{p}_i$  of length  $k$  for each  $i \in V$  such that for any  $(i, j) \in F$ , we have  $(i, j) \in E$  if and only if  $\langle \mathbf{d}_i, \mathbf{p}_j \rangle \leq t$ .

**Theorem 5.** *The  $(k, t)$ -REPRESENTATION-WITH-IGNORED-EDGES problem is  $\mathcal{NP}$ -complete.*

The theorem’s nontrivial proof is relegated to the appendix.<sup>5</sup> Here we give a proof sketch. One major idea is the construction of a *bit-grounding gadget*  $G_k$ —a subgraph where the bits are set uniquely (up to permutations) in any valid representation, and can be used to set the bits in other vertices. The gadget has  $\binom{k}{2}$  vertices; we prove that there is a unique (up to permutations)  $(k, 1)$ -representation of  $G_k$ , where each donor vector has a unique pair of ones, and similarly for each patient vector. Figure 1 shows  $G_4$ .

Then, we prove  $\mathcal{NP}$ -hardness by reduction from 3SAT. In the constructed instance of our problem, we set the threshold to 1. The crux of the reduction is to add a vertex for each clause in the given 3SAT formula, where in the patient vector, the bit corresponding to each literal in the clause is set to 1. This can be done by connecting the vertex to the bit-grounding gadget. Moreover, there is a special vertex that

<sup>5</sup><https://arxiv.org/abs/1605.07728>

does not have outgoing edges to any of the clause vertices. This means that it must have a 1 in a position that corresponds to one of the literals in each clause. A different part of the construction ensures that there is at most a single 1 in the two positions corresponding to a variable and its negation. Therefore, a valid assignment of the donor bits corresponds to a satisfying assignment for the 3SAT formula.

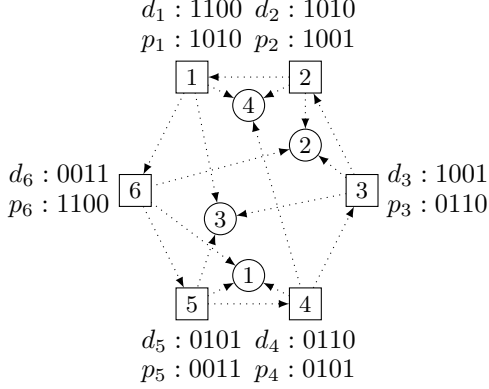


Figure 1: Gadget  $G_4$  with a subset of *non*-edges shown; all edges between circle vertices are also not in  $E$ .

## 4 Computing Small Representations of Real Kidney Exchange Compatibility Graphs

In this section, we test our hypothesis that real compatibility graphs can be represented by a substantially smaller number of attributes than required by the worst-case theoretical results of Section 3. We begin by presenting general math programming techniques to determine, given  $k, t \in \mathbb{Z}$ , whether a specific graph  $G = (V, E)$  is  $(k, t)$ -representable. We then show on real and generated compatibility graphs from the UNOS US-wide kidney exchange that small  $k$  suffices for  $(k, 0)$ -representation, and conclude by exploring the allowance of greater thresholds  $t$  on match size. Even small thresholds  $t > 0$  result in substantial societal gain.<sup>6</sup>

### 4.1 Mathematical Programming Formulations

Implementation of  $f_{\text{thresh}}^t$  can be written succinctly as a quadratically-constrained discrete feasibility program (QCP) with  $2k|V|$  binary variables, given as M1 below.

$$\begin{aligned}
 \langle \mathbf{d}_i, \mathbf{p}_j \rangle &\leq t & \forall (v_i, v_j) \in E \\
 \langle \mathbf{d}_i, \mathbf{p}_j \rangle &\geq (t+1) & \forall (v_i, v_j) \notin E \\
 \mathbf{d}_i, \mathbf{p}_i &\in \{0, 1\}^k & \forall v_i \in V
 \end{aligned} \tag{M1}$$

The constraint matrix for this program is not positive semi-definite, and thus the problem is not convex. Exploratory use of heuristic search via state-of-the-art integer nonlinear solvers (Bonami et al. 2008) resulted in poor performance (in terms of runtime and solution quality) on even small graphs. With that in mind, and motivated by the presence of substantially more mature integer *linear* program (ILP) solvers, we linearize M1, presented as M2 below.

<sup>6</sup>Code for this section will be made available once the double-blind period is over; the code itself uniquely identifies the authors.

$$\begin{aligned}
 \min \quad & \sum_{v_i \in V} \sum_{v_j \neq v_i \in V} \xi_{ij} \\
 \text{s.t.} \quad & d_i^q \geq c_{ij}^q \wedge p_j^q \geq c_{ij}^q & \forall v_i \neq v_j \in V, q \in [k] \\
 & d_i^q + p_j^q \leq 1 + c_{ij}^q & \forall v_i \neq v_j \in V, q \in [k] \\
 & \sum_q c_{ij}^q \leq t + (k-t)\xi_{ij} & \forall (v_i, v_j) \in E \\
 & \sum_q c_{ij}^q \geq (t+1)\xi_{ij} & \forall (v_i, v_j) \in E \\
 & \sum_q c_{ij}^q \geq t+1-k\xi_{ij} & \forall (v_i, v_j) \notin E \\
 & \sum_q c_{ij}^q \leq k - (k-t)\xi_{ij} & \forall (v_i, v_j) \notin E \\
 & d_i^q, p_i^q \in \{0, 1\} & \forall v_i \in V, q \in [k] \\
 & c_{ij}^q, \xi_{ij} \in \{0, 1\} & \forall v_i \neq v_j \in V, q \in [k]
 \end{aligned} \tag{M2}$$

M2 generalizes M1; while M1 searches for a feasible solution to the  $(k, t)$ -representation problem, M2 finds the “best” (possibly partially-incorrect) solution by minimizing the total number of edges that exist in the solution but not in the base graph  $G$ , or do not exist in the solution but do in  $G$ . This flexibility may be desirable in practice to strike a tradeoff between small  $k$  and accuracy of representation.

Interestingly, neither the fully general ILP nor its (smaller) instantiations for the special cases of feasibility and/or threshold  $t = 0$  were solvable by a leading commercial ILP solver (IBM ILOG Inc 2015) within 12 hours for even small graphs, primarily due to the model’s loose LP relaxation. Indeed, the model we are solving is inherently logical, which is known to cause such problems in traditional mathematical programming (Hooker 2002). With that in mind, we note that the special case of  $t = 0$  can be represented compactly as a satisfiability (SAT) problem in conjunctive normal form, given below as M3.

$$\begin{aligned}
 & \bigwedge_{q \in [k]} (\neg d_i^q \vee \neg p_j^q) & \forall (v_i, v_j) \in E \\
 & (z_{ij}^1 \vee z_{ij}^2 \vee \dots \vee z_{ij}^k) \wedge & \forall (v_i, v_j) \notin E \\
 & \bigwedge_{q \in [k]} [(\neg z_{ij}^q \vee d_i^q) \wedge (\neg z_{ij}^q \vee p_j^q)] &
 \end{aligned} \tag{M3}$$

This formulation maintains two sets of clauses: the first set enforces no bit-wise conflicts for edges in the underlying graph, while the second set enforces at least one conflict via  $k$  auxiliary variables  $z_{ij}^q$  for each possible edge  $(v_i, v_j) \notin E$ . M3 was amenable to parallel SAT solving (Biere 2014). Next, we present results on real graphs with this formulation.

### 4.2 $(k, 0)$ -representations of Real Graphs

Can real kidney exchange graphs be represented by a small number of attributes? To answer that question, we begin by testing on real match run data from the first two years of the United Network for Organ Sharing (UNOS) kidney exchange, which now contains 143 transplant centers, that is, 60% of all transplant centers in the US. We translate each compatibility graph into a CNF-SAT formulation according to M3, and feed that into a SAT solver (Biere 2014) with access to 16GB of RAM, 4 cores, and 60 minutes of wall time. (Timeouts are counted—conservatively against our paper’s qualitative message—as negative answers.)

Figure 2 shows a classical phase transition from unsatisfiability to satisfiability as  $k$  increases as a fraction of graph size, as well as an associated substantial increase in computational intractability centered around that phase transition. This phenomenon is common to many central problems in artificial intelligence (Cheeseman, Kanefsky,

and Taylor 1991; Hogg, Huberman, and Williams 1996; Walsh 2011). Indeed, we see that substantially fewer than  $|V|$  attributes are required to represent real graphs; compare with the lower bound of Theorem 4.

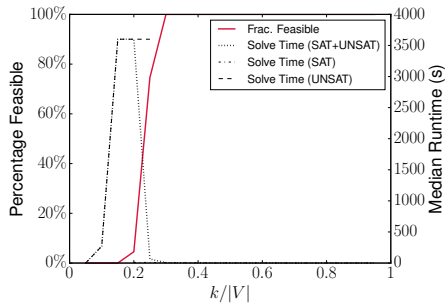


Figure 2: Classical hardness spike near the phase transition for  $(k, 0)$ -representation on real UNOS graphs.

Figure 3 explores the minimum  $k$  required to represent each graph as a function of  $|V|$ , compared against the theoretical upper bound of  $|V|$ . The shaded area represents those values of  $k$  where the SAT solver timed out; thus, the reported values of  $k$  are a conservative *upper* bound on the required minimum, which is still substantially lower than  $|V|$ .

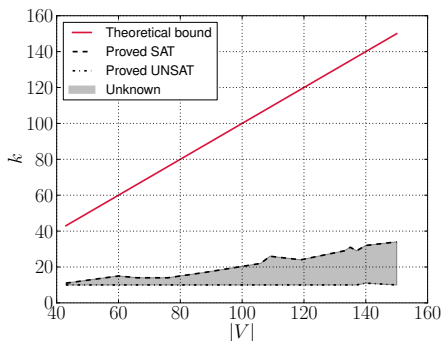


Figure 3: Comparison of number of bits (y-axis) required to  $(k, 0)$ -represent real UNOS compatibility graphs of varying sizes (x-axis). The theoretical bound of  $k = |V|$  is shown in red; it is substantially higher than the conservative upper bound of  $k$  solved by our SAT solver (upper dotted line).

### 4.3 Thresholding Effects on Matching Size

A motivation of this work is to provide a principled basis for optimally “flipping bits” of participants (via, e.g., immunosuppression) in fielded kidney exchanges, in the hope that additional edges in the compatibility graph will result in gains in the final algorithmic matchings. We now explore this line of reasoning—that is, increasing the  $t$  in  $f_{\text{thresh}}^t$  instead of the  $k$ , which is now endogenous to the underlying model—on realistic generated UNOS graphs of varying sizes.

Figure 4 shows the effect on the percentage of patient-donor pairs matched by 2- and 3-cycles as a global threshold  $t$  is raised incrementally from  $t = 0$  (the current status quo) to  $t = 5$ . Intuitively, larger compatibility graphs result in a higher fraction of pairs being matched; however, a complementary approach—making the graph denser via even small increases in  $t$ —also results in tremendous efficiency gains

of 3–4x (depending on  $|V|$ ) over the baseline for  $t = 0$ , and quickly increasing to all pairs being matched by  $t = 5$ .

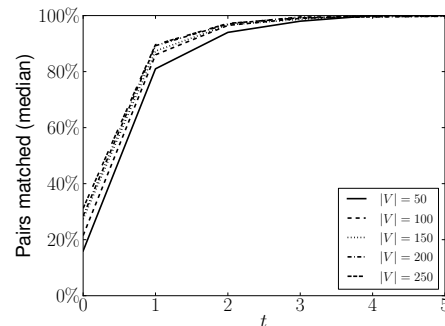


Figure 4: Pairs matched (%), y-axis) in generated UNOS graphs of varying sizes (lines), as  $t$  increases (x-axis).

We note that any optimal matching found after increasing a global threshold  $t$  could also be created by paying to change at most  $t$  bits per vertex in a graph; however, the practical *selection* of the minimum-sized set of at most  $t$  bits per vertex such that the size of the resulting optimal matching is equal to that found under the global threshold of  $t$  is a difficult two-stage problem and is left as future research. The large efficiency gains realized by moving from  $f_{\text{thresh}}^0$  to even  $f_{\text{thresh}}^1$  motivate this direction of research.

## 5 Conclusions & Future Research

Motivated by the increasing size of real-world kidney exchanges, we presented a compact approach to modeling kidney exchange compatibility graphs. Our approach is intimately connected to classical intersection graph theory, and can be viewed as the first exploration and practical application of  $p$ -intersection digraphs. We gave necessary and sufficient conditions for losslessly shrinking the representation of an arbitrary compatibility graph in this model. Real compatibility graphs, however, are not arbitrary, and are created from characteristics of the patients and donors; using real data from the UNOS US-wide kidney exchange, we showed that using only a small number of attributes suffices to represent real graphs. This observation is of potential practical importance; if real graphs can be represented by a constant number of attributes, then central  $\mathcal{NP}$ -hard problems in general kidney exchange are solvable in polynomial time.

This paper only addresses the representation of static compatibility graphs; in reality, exchanges are dynamic, with patients and donors arriving and departing over time (Ünver 2010). Extending the proposed method to cover time-evolving graphs is of independent theoretical interest, but may also be useful in speeding up the (presently intractable) dynamic clearing problem (Awasthi and Sandholm 2009; Dickerson, Procaccia, and Sandholm 2012; Anderson 2014; Dickerson and Sandholm 2015; Glorie et al. 2015). Better exact and approximate methods for computing  $(k, t)$ -representations of graphs would likely be a prerequisite for that research. Adaptation of the theoretical results to models of lung, liver, and multi-organ exchange would also be of practical use (Ergin, Sönmez, and Ünver 2014; 2015; Luo and Tang 2015; Dickerson and Sandholm 2016).

## References

- Abraham, D.; Blum, A.; and Sandholm, T. 2007. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 295–304.
- Anderson, R.; Ashlagi, I.; Gamarnik, D.; and Roth, A. E. 2015. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences* 112(3):663–668.
- Anderson, R. 2014. *Stochastic models and data driven simulations for healthcare operations*. Ph.D. Dissertation, Massachusetts Institute of Technology.
- Awasthi, P., and Sandholm, T. 2009. Online stochastic optimization in the large: Application to kidney exchange. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 405–411.
- Biere, A. 2014. Yet another local search solver and lingeling and friends entering the SAT Competition 2014. *SAT Competition 2014*:2.
- Biró, P.; Manlove, D. F.; and Rizzi, R. 2009. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications* 1(04):499–517.
- Bonami, P.; Biegler, L. T.; Conn, A. R.; Cornuéjols, G.; Grossmann, I. E.; Laird, C. D.; Lee, J.; Lodi, A.; Margot, F.; Sawaya, N.; et al. 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* 5(2):186–204.
- Cheeseman, P.; Kanefsky, B.; and Taylor, W. 1991. Where the really hard problems are. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI)*, 331–337.
- Chung, M. S., and West, D. B. 1994. The  $p$ -intersection number of a complete bipartite graph and orthogonal double coverings of a clique. *Combinatorica* 14(4):453–461.
- Dickerson, J. P., and Sandholm, T. 2015. FutureMatch: Combining human value judgments and machine learning to match in dynamic environments. In *AAAI Conference on Artificial Intelligence (AAAI)*, 622–628.
- Dickerson, J. P., and Sandholm, T. 2016. Multi-organ exchange: The whole is greater than the sum of its parts. *Journal of Artificial Intelligence Research*. To appear.
- Dickerson, J. P.; Manlove, D.; Plaut, B.; Sandholm, T.; and Trimble, J. 2016. Position-indexed formulations for kidney exchange. In *Proceedings of the ACM Conference on Economics and Computation (EC)*.
- Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2012. Dynamic matching via weighted myopia with application to kidney exchange. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1340–1346.
- Eaton, N.; Gould, R. J.; and Rödl, V. 1996. On  $p$ -intersection representations. *J. Graph Theory* 21(4):377–392.
- Erdős, P.; Goodman, A. W.; and Pósa, L. 1966. The representation of a graph by set intersections. *Canad. J. Math.* 18:106–112.
- Ergin, H.; Sönmez, T.; and Ünver, M. U. 2014. Lung exchange. Working paper.
- Ergin, H.; Sönmez, T.; and Ünver, M. U. 2015. Liver exchange. Working paper.
- Glorie, K.; Carvalho, M.; Constantino, M.; Bouman, P.; and Viana, A. 2015. Robust models for the kidney exchange problem. Working paper.
- Glorie, K. M.; van de Klundert, J. J.; and Wagelmans, A. P. M. 2014. 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.
- Harary, F.; Kabell, J. A.; and McMorris, F. R. 1982. Bipartite intersection graphs. *Comment. Math. Univ. Carolin.* 23(4):739–745.
- Hogg, T.; Huberman, B. A.; and Williams, C. P. 1996. Phase transitions and the search problem. *Artificial Intelligence* 81(1-2):1–15.
- Hooker, J. N. 2002. Logic, optimization and constraint programming. *INFORMS Journal of Computing* 14:295–321.
- IBM ILOG Inc. 2015. CPLEX 12.6 User’s Manual.
- Kou, L. T.; Stockmeyer, L. J.; and Wong, C. K. 1978. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Comm. ACM* 21(2):135–139.
- Luo, S., and Tang, P. 2015. Mechanism design and implementation for lung exchange. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 209–215.
- Mak-Hau, V. 2015. On the kidney exchange problem: cardinality constrained cycle and chain problems on directed graphs: a survey of integer programming approaches. *Journal of Combinatorial Optimization* 1–25.
- McKee, T. A., and McMorris, F. R. 1999. *Topics in intersection graph theory*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.
- Montgomery, R.; Gentry, S.; Marks, W. H.; Warren, D. S.; Hiller, J.; Houpp, J.; Zachary, A. A.; Melancon, J. K.; Maley, W. R.; Rabb, H.; Simpkins, C.; and Segev, D. L. 2006. Domino paired kidney donation: a strategy to make best use of live non-directed donation. *The Lancet* 368(9533):419–421.
- Orlin, J. 1977. Contentment in graph theory: covering graphs with cliques. *Nederl. Akad. Wetensch. Proc. Ser. A* 39(5):406–424.
- Rapaport, F. T. 1986. The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proceedings* 18:5–9.
- Rees, M.; Kopke, J.; Pelletier, R.; Segev, D.; Rutter, M.; Fabrega, A.; Rogers, J.; Pankewycz, O.; Hiller, J.; Roth, A.; Sandholm, T.; Ünver, U.; and Montgomery, R. 2009. A nonsimultaneous, extended, altruistic-donor chain. *New England Journal of Medicine* 360(11):1096–1101.
- Roth, A.; Sönmez, T.; Ünver, U.; Delmonico, F.; and Saidman, S. L. 2006. Utilizing list exchange and nondirected donation through ‘chain’ paired kidney donations. *American Journal of Transplantation* 6:2694–2705.
- Roth, A.; Sönmez, T.; and Ünver, U. 2004. Kidney exchange. *Quarterly Journal of Economics* 119(2):457–488.
- Roth, A.; Sönmez, T.; and Ünver, U. 2005. A kidney exchange clearinghouse in New England. *American Economic Review* 95(2):376–380.
- Sen, M.; Das, S.; Roy, A. B.; and West, D. B. 1989. Interval digraphs: an analogue of interval graphs. *J. Graph Theory* 13(2):189–202.
- Ünver, U. 2010. Dynamic kidney exchange. *Review of Economic Studies* 77(1):372–414.
- Walsh, T. 2011. Where are the hard manipulation problems? *Journal of Artificial Intelligence Research* 42:1–29.