Small Representations of Big Kidney Exchange Graphs

John P. Dickerson and **Aleksandr M. Kazachkov** and **Ariel Procaccia** and **Tuomas Sandholm**

Carnegie Mellon University

{dickerson,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 nationwide kidney exchange, we show how many attributes are needed to encode real compatibility 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.[1] Complementing potential cadaveric transplantation via the deceased donor waiting list, a recent innovation—kidney exchange [Rapaport, 1986; Roth *et al.*, 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 roughly 10% of living donation in the US, with that percentage increasing every year.

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 constraint. 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. As a concrete 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 *et al.*, 2007; Biró *et al.*, 2009] and also extremely difficult to solve in practice [Glorie *et al.*, 2014; Anderson *et al.*, 2015; Plaut *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 just 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 *et al.*, 2004; Roth *et al.*, 2005a; Roth *et al.*, 2005b].

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[2] 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 with its constituent donor and patient attribute vectors $\mathbf{d}_i$ and $\mathbf{p}_i$, respectively. Here, the qth 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 *et al.*, 2007; Biró *et al.*, 2009], a variety of custom clearing algorithms address adaptations of the clearing problem in practice.[3] 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, 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 donor at the end of the chain 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 Θ be the set of all possible types, and $\theta \in \Theta$ represent one such individual type. Then, 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 θ and θ'. (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 insert v_j in place of v_i, and v_i in place of v_j, and obtain a feasible cycle cover of the same size. This observation drives our theoretical algorithmic results.

In more detail, for a vector of types $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k) \in \Theta^k$, let us denote $f_C(\boldsymbol{\theta}) = 1$ if and only if $f(\theta_t, \theta_{t+1}) = 1$ for all $t < k$, and $f(\theta_k, \theta_1) = 1$. In other words, $f_C(\boldsymbol{\theta}) = 1$ if every k vertices $i_1, \ldots, i_k$ of types $\theta_1, \ldots, \theta_k$, respectively, are involved in a cycle in the graph. Furthermore, for $L \leq n = |V|$, denote

$$\mathcal{T}(L) = \{\boldsymbol{\theta} \in \Theta^k : k \leq L \text{ and } f_C(\boldsymbol{\theta}) = 1\}.$$

That is, $\mathcal{T}(L)$ contains all vectors of types that induce feasible cycles of length at most L.

Now consider the following algorithm for L-CYCLE-COVER in our model:

Algorithm 1 L-CYCLE-COVER

---

1. $\mathcal{C}^* \leftarrow \emptyset$

2. **for** every collection of numbers $\{m_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \mathcal{T}(L)}$ such that $\sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m_{\boldsymbol{\theta}} \leq n$

   - **if** there exists cycle cover $\mathcal{C}$ such that $\|\mathcal{C}\|_V > \|\mathcal{C}^*\|_V$ and for all $\boldsymbol{\theta} \in \mathcal{T}(L)$, $\mathcal{C}$ contains $m_{\boldsymbol{\theta}}$ cycles consisting of vertices of the types in $\boldsymbol{\theta}$ **then** $\mathcal{C}^* \leftarrow \mathcal{C}$

3. **return** $\mathcal{C}^*$

Here, $\|\mathcal{C}\|_V$ denotes the number of unique *vertices* matched in a cycle cover $\mathcal{C}$. We claim that, in our setting, Algorithm 1 is optimal and computationally efficient.

Theorem 1. *Suppose that L and $|\Theta|$ are constants. Then 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 $\mathcal{C}^*$. For each $\boldsymbol{\theta} \in \mathcal{T}(L)$, let $m_{\boldsymbol{\theta}}^*$ be the number of cycles in $\mathcal{C}^*$ that are consistent with the types in $\boldsymbol{\theta}$. Clearly $\sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m_{\boldsymbol{\theta}}^* \leq n$, as there are only n vertices so there cannot be more than n cycles (in

---

[2]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.

[3]For an overview of practical approaches to solving the clearing problem, see a recent survey due to Mak-Hau [2015].

fact, $n/2$ is also a valid upper bound). Therefore, Algorithm 1 considers the collection of numbers $m_{\boldsymbol{\theta}}^*$ in Step 2. Because this collection of numbers does induce a valid cycle cover that is of the same size as $\mathcal{C}^*$, the algorithm would update its incumbent cycle cover if it was not already optimal.

We next analyze the running time of the algorithm. First, note that it is straightforward to check whether the numbers $\{m_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \mathcal{T}(L)}$ induce 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 type θ to construct all the cycles that require them. This simply amounts to multiplying each $m_{\boldsymbol{\theta}}$ by the number of times type θ appears in $\boldsymbol{\theta}$, and verifying that the sum of these products over all $\boldsymbol{\theta}$ in $\mathcal{T}(L)$ is at most the number of vertices of type θ.

Second, we argue that there is only a polynomial number of possibilities to construct a collection of numbers $\{m_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \mathcal{T}(L)}$ such that $\sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m_{\boldsymbol{\theta}} \leq n$. 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. $\quad\square$

Even for constant L, the running time of Algorithm 1 is exponential in k. But this is to be expected. Indeed, any graph can trivially be represented using a set Θ 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* k, we could solve the general L-CYCLE-COVER problem in polynomial time—and that problem is $\mathcal{NP}$-hard [Abraham *et al.*, 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.[4] 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 $c : \Theta \times \Theta \to \mathbb{R}$ representing the cost of changing a vertex of type θ to type θ'. 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 $\theta_i \in \Theta$, there are n_i vertices. Then, for each of the $(|\Theta| - 1)$ choices of which type $\theta \neq \theta_i$ to switch to, choose how many vertices from θ_i will switch to

a different type; there are $(n_i + 1)$ choices. Do this for all $|\Theta|$ types, resulting in $\prod_{\theta_i \in \Theta}[(n_i + 1)(|\Theta| - 1)]$ choices. Note that $\sum_{\theta_i \in \Theta}(n_i + 1) = n + |\Theta|$, meaning $\prod_{\theta_i \in \Theta}[(n_i + 1)(|\Theta| - 1)] \leq ((n + |\Theta|) \cdot (|\Theta| - 1)/|\Theta|)^{|\Theta|} \leq (n + |\Theta|)^{|\Theta|}$. Since $|\Theta|$ is a constant, this is polynomial in n; invoking an adaptation of the polynomial time Algorithm 1 that subtracts out $c(\theta, \theta')$ for every vertex that switches from θ to θ', for each of the polynomially-many choices, concludes the proof. $\quad\square$

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 $k = 2$, such that a donor's first (resp. second) bit is set if his blood holds A (resp. B) antigens. and a patient's first (resp. second) bit is set if she cannot receive from blood containing A (resp. 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 $V = [n] := \{1, \ldots, n\}$ 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_{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 a nonnegative integer t, the function f_{thresh}^t is related to *p-intersection graphs* [Chung and West, 1994; Eaton *et al.*, 1996], in which an edge exists between two vertices if their corresponding sets intersect in at least $p \geq 1$ elements.

In particular, our model is similar to that of *intersection digraphs* [Sen *et al.*, 1989], or equivalently *bipartite intersection graphs* [Harary *et al.*, 1982], both also considered in [Orlin, 1977]. Both of 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 the thresholding model we have defined, 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 can we select vertices with bit vectors $\mathbf{d}_i$ and $\mathbf{p}_i$ of length k such that f_{thresh}^t can create *any* graph of a specific size?

Formally, we say that G is (k, t)-*representable* (by feasibility function f_{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 *et al.*, 1966] that any graph can be represented as an intersection graph with $k \le n^2/4$. In contrast, the next theorem shows that, in our model, $k \le n$ suffices to represent any graph. It is akin to a result on the *term rank* of the adjacency matrix of $\overline{G}$ [Orlin, 1977, Theorem 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 ith standard basis vector, i.e., the vector of length $n_1 + 1$ with a 1 in the ith 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 $i \in [n]$ 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_{thresh}^0. □

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 $n \ge 3$, 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 $k \ge n$. Indeed, otherwise we can assume that $q_1 = q_i$ for some $i \ne 1$ (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 $i \in V \setminus \{1\}$. □

More generally, it is easy to see that any graph that is $(k, 0)$-representable is also $(k+t, t)$-representable for any $t \ge 0$. 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 any 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 do we actually need?

Various problems related to intersection graphs are $\mathcal{NP}$-complete for general graphs [Kou *et al.*, 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 $k \ge 1$ and $t \ge 0$, 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 $\langle \mathbf{d}_i, \mathbf{p}_j \rangle \le t$.

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

The theorem's nontrivial proof is omitted due to lack of space.[5] 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 5 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 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

---

[5]The complete proof is given in Appendix A.

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.

$$d_1 : 1100 \quad d_2 : 1010$$
$$p_1 : 1010 \quad p_2 : 1001$$

$$d_6 : 0011 \qquad d_3 : 1001$$
$$p_6 : 1100 \qquad p_3 : 0110$$

$$d_5 : 0101 \quad d_4 : 0110$$
$$p_5 : 0011 \quad p_4 : 0101$$
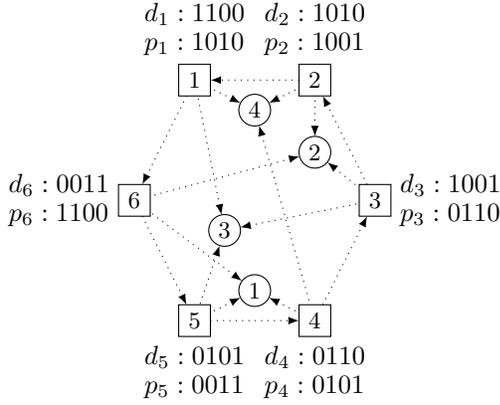
Figure 1: Gadget G_4 with a subset of *non*-edges shown; all edges between circle vertices (those in G_4^2) are also not in E.

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 mathematical 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. We find even small thresholds $t > 0$ result in substantial societal gain.[6]

4.1 Mathematical Programming Formulations

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

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

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.

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

M2 generalizes M1; while M1 searches for a feasible solution to the (k, t)-representation problem, M2 searches for 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.

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

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} 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 using this formulation.

4.2 $(k, 0)$-representations of Real Kidney Exchange 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 *et al.*, 1991;

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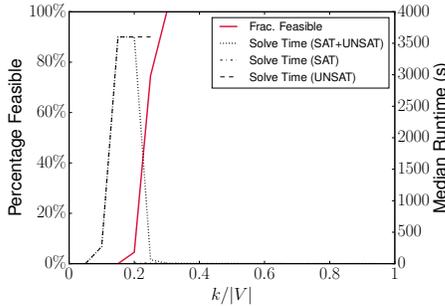
Figure 2: Classical hardness spike near the phase transition for $(k, 0)$-representation on real UNOS compatibility graphs.

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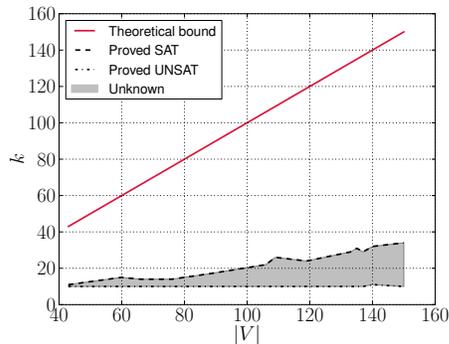
Figure 3: Comparison of number of bits (y-axis) required to $(k, 0)$-represent real UNOS compatibility graphs of varying sizes (x-axis). The theoretical bound of $k = |V|$ is shown in red; it is substantially higher than the conservative upper bound of k solved by our SAT solver (upper dotted line).

4.3 Thresholding Effects on Matching Size

One motivation of this work is to provide a principled basis for optimally "flipping bits" of participants (via, e.g., immunosuppresion) 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_{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 $t = 1$, and quickly increasing to all pairs being matched by $t = 5$.
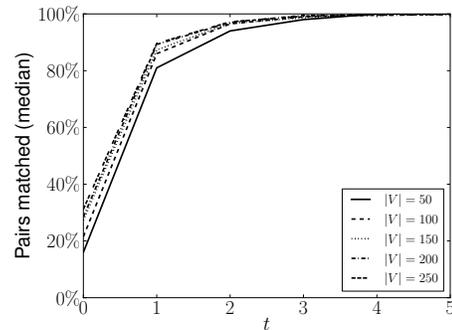
Figure 4: Pairs matched (%, y-axis) in generated UNOS graphs of varying sizes (lines), as t increases (x-axis).

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_{thresh}^0 to even f_{thresh}^1 motivate this direction of research.

5 Conclusions & Future Research

Motivated by the increasing size of real-world kidney exchanges, in this paper, 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 *et al.*, 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 line of research. Furthermore, adaptation of the theoretical results to alternate organ models like lung [Ergin *et al.*, 2014; Luo and Tang, 2015], liver [Ergin *et al.*, 2015], and multi-organ [Dickerson and Sandholm, 2016] exchange would be of practical use.

References

[Abraham *et al.*, 2007] David Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *EC*, 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. *P Natl Acad Sci*, 112(3):663–668, 2015.

[Anderson, 2014] Ross Anderson. *Stochastic models and data driven simulations for healthcare operations*. PhD thesis, Massachusetts Institute of Technology, 2014.

[Awasthi and Sandholm, 2009] Pranjal Awasthi and Tuomas Sandholm. Online stochastic optimization in the large: Application to kidney exchange. In *IJCAI*, 2009.

[Biere, 2014] Armin Biere. Yet another local search solver and lingeling and friends entering the SAT Competition. *SAT*, 2014.

[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. *Discret Math, Alg, Appl*, 1(04):499–517, 2009.

[Bonami *et al.*, 2008] Pierre Bonami, Lorenz T Biegler, Andrew R Conn, Gérard Cornuéjols, Ignacio E Grossmann, Carl D Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, et al. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optim*, 5(2):186–204, 2008.

[Cheeseman *et al.*, 1991] P Cheeseman, B Kanefsky, and W Taylor. Where the really hard problems are. In *IJCAI*, 1991.

[Chung and West, 1994] Myung S. Chung and Douglas B. West. The p-intersection number of a complete bipartite graph and orthogonal double coverings of a clique. *Combinatorica*, 14(4):453–461, 1994.

[Dickerson and Sandholm, 2015] John P. Dickerson and Tuomas Sandholm. FutureMatch: Combining human value judgments and machine learning to match in dynamic environments. In *AAAI*, 2015.

[Dickerson and Sandholm, 2016] John P. Dickerson and Tuomas Sandholm. Multi-organ exchange: The whole is greater than the sum of its parts. *J Artif Intell Res*, 2016. To appear.

[Dickerson *et al.*, 2012] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Dynamic matching via weighted myopia with application to kidney exchange. In *AAAI*, 2012.

[Eaton *et al.*, 1996] Nancy Eaton, Ronald J. Gould, and Vojtech Rödl. On p-intersection representations. *J Graph Theor*, 21(4):377–392, 1996.

[Erdős *et al.*, 1966] Paul Erdős, A. W. Goodman, and Lajos Pósa. The representation of a graph by set intersections. *Canadian J Math*, 18:106–112, 1966.

[Ergin *et al.*, 2014] Haluk Ergin, Tayfun Sönmez, and M Utku Ünver. Lung exchange, 2014. Working paper.

[Ergin *et al.*, 2015] Haluk Ergin, Tayfun Sönmez, and M Utku Ünver. Liver exchange, 2015. Working paper.

[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. *M&SOM-Manuf Serv Op*, 16(4):498–512, 2014.

[Glorie *et al.*, 2015] Kristiaan Glorie, Margarida Carvalho, Miguel Constantino, Paul Bouman, and Ana Viana. Robust models for the kidney exchange problem, 2015. Working paper.

[Harary *et al.*, 1982] Frank Harary, Jerald A. Kabell, and F. R. McMorris. Bipartite intersection graphs. *Math Univ Carolin*, 23(4):739–745, 1982.

[Hogg *et al.*, 1996] Tad Hogg, Bernardo A. Huberman, and Colin P. Williams. Phase transitions and the search problem. *Artif Intell*, 81(1-2):1–15, 1996.

[Hooker, 2002] John N. Hooker. Logic, optimization and constraint programming. *INFORMS J Comput*, 14:295–321, 2002.

[IBM ILOG Inc, 2015] IBM ILOG Inc. CPLEX 12.6 User's Manual, 2015.

[Kou *et al.*, 1978] L. T. Kou, L. J. Stockmeyer, and C. K. Wong. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Commun ACM*, 21(2):135–139, 1978.

[Luo and Tang, 2015] Suiqian Luo and Pingzhong Tang. Mechanism design and implementation for lung exchange. In *IJCAI*, 2015.

[Mak-Hau, 2015] Vicky Mak-Hau. On the kidney exchange problem: cardinality constrained cycle and chain problems on directed graphs: a survey of integer programming approaches. *J Comb Optim*, pages 1–25, 2015.

[McKee and McMorris, 1999] Terry A. McKee and F. R. McMorris. *Topics in intersection graph theory*. SIAM, Philadelphia, PA, 1999.

[Montgomery *et al.*, 2006] Robert Montgomery, Sommer Gentry, William H Marks, Daniel S Warren, Janet Hiller, Julie Houp, Andrea A Zachary, J Keith Melancon, Warren R Maley, Hamid Rabb, Christopher Simpkins, and Dorry L Segev. Domino paired kidney donation: a strategy to make best use of live non-directed donation. *Lancet*, 368(9533):419–421, 2006.

[Orlin, 1977] James Orlin. Contentment in graph theory: covering graphs with cliques. *Indag Math*, 39(5):406–424, 1977.

[Plaut *et al.*, 2016] Benjamin Plaut, John P. Dickerson, and Tuomas Sandholm. Fast optimal clearing of capped-chain barter exchanges. In *AAAI*, 2016.

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

[Rees *et al.*, 2009] Michael Rees, Jonathan Kopke, Ronald Pelletier, Dorry Segev, Matthew Rutter, Alfredo Fabrega, Jeffrey Rogers, Oleh Pankewycz, Janet Hiller, Alvin Roth, Tuomas Sandholm, Utku Ünver, and Robert Montgomery. A nonsimultaneous, extended, altruistic-donor chain. *New Engl J Med*, 360(11):1096–1101, 2009.

[Roth *et al.*, 2004] Alvin Roth, Tayfun Sönmez, and Utku Ünver. Kidney exchange. *Q J Econ*, 119(2):457–488, 2004.

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

[Roth *et al.*, 2005b] Alvin Roth, Tayfun Sönmez, and Utku Ünver. Pairwise kidney exchange. *J Econ Theory*, 125(2):151–188, 2005.

[Roth *et al.*, 2006] Alvin Roth, Tayfun Sönmez, Utku Ünver, Frank Delmonico, and Susan L. Saidman. Utilizing list exchange and nondirected donation through 'chain' paired kidney donations. *Am J Transplant*, 6:2694–2705, 2006.

[Sen *et al.*, 1989] M. Sen, S. Das, A. B. Roy, and D. B. West. Interval digraphs: an analogue of interval graphs. *J Graph Theor*, 13(2):189–202, 1989.

[Ünver, 2010] Utku Ünver. Dynamic kidney exchange. *Rev Econ Stud*, 77(1):372–414, 2010.

[Walsh, 2011] Toby Walsh. Where are the hard manipulation problems? *J Artif Intell Res*, 42:1–29, 2011.

Appendix A: Omitted Proofs

In this section, we provide the full proof of Theorem 5. Recall the (k, t)-REPRESENTATION WITH IGNORED EDGES: given an input of a directed graph $G = (V, E)$, a subset F of E, 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 the $\{i, j\} \in F$ if and only if $\langle \mathbf{d}_i, \mathbf{p}_j \rangle \leq t$.

Consider the gadget G_k defined as follows on a graph on $\binom{k}{2} + k$ vertices. Let G_k^1 be the graph defined in Theorem 4 on $\binom{k}{2}$ vertices, i.e., the complement of a directed cycle on this many vertices. Associate with each vertex $u \in G_k^1$ a unique element from $\binom{[k]}{2}$ (all subsets of $[k]$ of size 2). Let G_k^2 be an independent set of k vertices. For each vertex $i \in G_k^2$, $i \in [k]$, add an incoming edge into i from $u \in G_k^2$ if and only if $i \in S_u$. Figure 5 shows G_4.

$$d_1 : 1100 \quad d_2 : 1010$$
$$p_1 : 1010 \quad p_2 : 1001$$

$$d_6 : 0011 \qquad\qquad\qquad d_3 : 1001$$
$$p_6 : 1100 \qquad\qquad\qquad p_3 : 0110$$

$$d_5 : 0101 \quad d_4 : 0110$$
$$p_5 : 0011 \quad p_4 : 0101$$
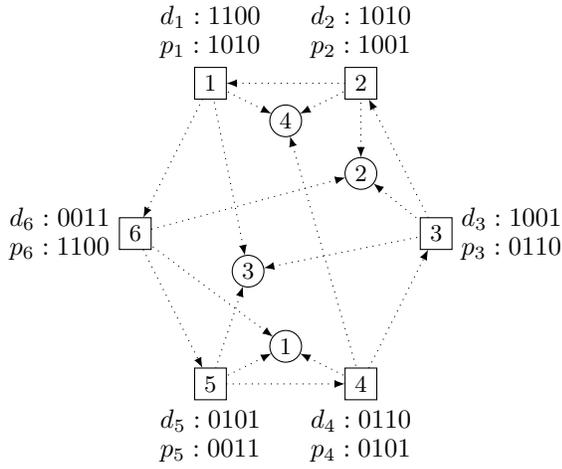
Figure 5: Gadget G_4 with a subset of *non*-edges shown; all edges between circle vertices (those in G_4^2) are also not in E.

Denote the *donor neighborhood* of $i \in V$ by $N_d(i) = \{j \in V : (i, j) \in E, i \neq j\}$, i.e., the set of patients compatible with donor i. Similarly, the *patient neighborhood* of $j \in V$ is $N_p(j) = \{i \in V : \{i, j\} \in E, i \neq j\}$.

Lemma 1. *There is a unique (up to permutations) $(k, 1)$-representation of G_k.*

Proof. First consider G_k^1. For all $u \in V(G_k^1)$, since $\{u, u - 1\} \notin E(G_k^1)$, and the compatibility function is f_{thresh}^1, there exist two distinct conflict bits q_1^u and q_2^u in $Q_d(u) \cap Q_p(u-1)$. Moreover, for any u, v distinct, $\{q_1^u, q_2^u\} \neq \{q_1^v, q_2^v\}$. Otherwise, $\{q_1^u, q_2^u\} \subseteq Q_p(v-1)$ and $\{q_1^v, q_2^v\} \subseteq Q_p(u-1)$, but at least one of the edges $\{u, v-1\}$ or $\{v, u-1\}$ exists in G_k^1.

In addition, $|Q_d(u)| = 2$ for all $u \in V(G_k^1)$. Suppose not, and there exists a third distinct (from q_1^u and q_2^u) conflict bit q_3^u in $Q_d(u)$. As the number of vertices is $\binom{k}{2}$, there exists a vertex v_1 with $\{q_1^{v_1}, q_2^{v_1}\} = \{q_1^u, q_3^u\}$, and a (different) vertex v_2 with $\{q_1^{v_2}, q_2^{v_2}\} = \{q_2^u, q_3^u\}$. Then $\{u, v_1 - 1\}$ and $\{u, v_2 - 1\}$ are both not in $E(G_k^1)$. However, u has edges to all vertices except itself and $u - 1$, which is a contradiction,

as u, v_1, and v_2 are all distinct. From this, it also follows that $|Q_p(u)| = 2$.

We have thus shown that every vertex $u \in G_k^1$ has exactly two bits set to one in its donor attribute vector, with a unique pair of bits per vertex, and $Q_d(u) = Q_p(u - 1)$. However, without more structure, it is not possible to tell in which donor vectors a particular conflict bit appears. The additional graph G_k^2 allows us to identify this, up to permutations.

Since there are no outgoing edges from any of the vertices in G_k^2, and every pair of bits in $\binom{[k]}{2}$ appears in exactly one patient vector of a vertex in G_k^1, each donor vector in G_k^2 must be the all-ones vector of length k.

Consider vertex $i \in [k]$ in G_k^2. It has an incoming edge from each vertex $u \in V(G_k^1)$ such that $i \in S_u$ and it is missing the $\binom{k-1}{2}$ other possible incoming edges from G_k^1 (note that the labeling of the vertices, as well as the choices of the sets S_u, are made without any knowledge of the bit-vectors associated with the vertices). We next show that $\left|\cap_{u \in N_p(i)} Q_d(u)\right| = 1$. That this quantity is at most 1 is clear, as $Q_d(u)$ and $Q_d(v)$ intersect in at most one conflict bit for all $u, v \in V(G_k^1)$, $u \neq v$. If this quantity were 0, then for some $u, v \in N_p(i)$, $Q_d(u) \cap Q_d(v) = \emptyset$. But then at least two zeroes would appear in $Q_p(i)$, which is a contradiction as it implies that i would have more than k incoming edges. Thus, the patient vector $\mathbf{p}_i$ for $i \in V(G_k^2)$ has exactly one zero and ones elsewhere. Moreover, since $N_p(i) \neq N_p(j)$ for any distinct $i, j \in [k]$, it follows that $\mathbf{p}_i \neq \mathbf{p}_j$, so each patient vector is distinct and the position of its only zero is unique. $\square$

Lemma 2. *Consider a digraph G having G_k as a subgraph and an additional vertex $x \notin V(G_k)$. We use the compatibility function f_{thresh}^1 and seek to find a $(k, 1)$-representation for the induced subgraph $G[V(G_k) \cup \{x\}]$. Let $U \subseteq V(G_k^1)$ having that property that if $v \in V(G_k^1)$ with $Q_d(v) \subseteq \cup_{u \in U} Q_d(u)$, then $v \in U$. Let $U' = \{u \in V(G_k^1) : u + 1 \in U\}$. Let $Q = \cup_{u \in U} Q_d(u)$.*

If $N_p(x) = V(G_1^k) \setminus U$, then $Q_p(x) = Q$. If $N_d(x) = V(G_1^k) \setminus U'$, then $Q_d(x) = Q$.

Proof. We use the fact that there are exactly two bits set to one in the donor and patient vectors of each vertex in G_k in any $(k, 1)$-representation. For the first statement, since x has no edge from $u \in U$, $Q_p(x) \supseteq Q_d(u)$. Thus $Q_p(x) \supseteq Q$. Now let $v \in V(G_k^1) \setminus U$ and $q_v \in Q_d(v) \setminus Q$. If $q_v \in Q_p(x)$, then for each $q \in Q$, there exists a vertex w in G_k^1 with $Q_d(w) = \{q, q_v\}$, so that $\{w, x\}$ would also not be an edge of G, a contradiction. Hence, $Q_p(x) = Q$. The second statement follows analogously. $\square$

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

Proof. Consider a 3SAT formula on n variables and with m clauses. Set $k = 2n + 2$, and build the following graph on $2 + n + m + \binom{k}{2} + k$ vertices. The first two vertices are labeled v and u. Then there is a vertex v_i for each variable $i \in [n]$, a vertex c for each clause $c \in [m]$. Call the subgraph induced

by these $2 + n + m$ vertices G'. The last vertices come from the gadget G_k.

The vertices in G_k^2 ground the k bits used in each donor and patient vector.. We think of the k bits, in order, as corresponding to the n positive literals, then their n negations, followed by two "extra" bits. Then the index of literal x_i will be i, and the index of literal $\bar{x}_i$ will be $n + i$. For i and j distinct in $V(G_k^2)$, $|N_p(i) \cap N_p(j)| = 1$ within G_k. Denote this vertex of G_k^1 by $v(i, j)$, and without loss of generality we can assume that $Q_d(v(i, j)) = \{i, j\}$.

The edges among vertices in the induced subgraph G_k are already defined; we define (a subset) of the rest of the edges. Together, these comprise precisely the subset F of the edges and non-edges specified as an input the instance we are creating of (k, t)-REPRESENTATION WITH IGNORED EDGES.

Vertex v has no incoming edges, and the only outgoing edges from v to $V(G')$ are to every variable vertex v_i, $i \in [n]$. The rest of the vertices that are not in G_k have no outgoing edges at all, to either $V(G')$ or $V(G_k)$, and the only incoming edges are from vertices of G_k^1. Vertex u has an incoming edge from every vertex of G_k^1 except $v(2n + 1, 2n + 2)$. For each variable vertex v_i, $i \in [n]$, it has an incoming edge from every vertex in $V(G_k^1)$ except $v(i, n+i)$. For each clause $c \in [m]$, let $\{c_1, c_2, c_3\}$ be the indices of the three literals that appear in c. Let $C \subset V(G_k^1)$ be $\{v(c_1, c_2), v(c_1, c_3), v(c_2, c_3), v(c_1, k), v(c_2, k), v(c_3, k)\}$. Then the vertex corresponding to c has an incoming edge from every vertex in $V(G_k^1) \setminus C$.

Every vertex of $V(G')$ except for v will have a donor vector with every bit set to one because there are no outgoing edges to any vertex of G_k^1, and v will have an all-ones patient vector because it has no incoming edges from G_k^1. By Lemma 2, in any $(k, 1)$-representation of G, vertex u will have $Q_p(u) = \{2n + 1, 2n + 2\}$. Variable vertex v_i, $i \in [n]$, will have $Q_p(v_i) = \{i, n + i\}$. Clause vertex $c \in [m]$ will have $Q_p(c) = \{c_1, c_2, c_3, 2n + 2\}$.

Since the graph does not have an edge from v to u, $\{2n + 1, 2n + 2\} \subseteq Q_d(v)$ (these are the only two conflict bits in $Q_p(u)$ and the threshold is 1). Since the graph has an edge from v to each variable vertex v_i, $i \in [n]$, $Q_d(v)$ must contain at most one of the indices corresponding to the variable or its negation (there are no conflicts from the extra bits, which are set to 0 in the patient vector of v_i). Since the graph does not have an edge from v to any of the clause vertices, it has to have at least one conflict bit in a position corresponding to one of the three literals in the clause (the other conflict comes from the extra bit $2n + 2$).

Thus, finding a suitable $(k, 1)$-representation that satisfies the adjacencies of edges that appear in F would involve finding an appropriate set $Q_d(v)$, which we have shown corresponds to choosing at most one value for each x_i, as well as choosing at least one literal that appears in each clause. This is the same as the problem of finding a satisfying formula for the initial instance of 3SAT.

As an example, consider the 3SAT formula $x_1 \vee \bar{x}_2 \vee x_3$. Figure 6 shows the most relevant part of the graph used in the reduction. One possible $(k, 1)$-representation may have $Q_d(v) = \{1, 7, 8\}$, indicating $x_1 = 1$ and the rest of the

variables are arbitrary. Another example of a possible representation is $Q_d(v) = \{1, 3, 5, 7, 8\}$, meaning $x_1 = 1$, $x_2 = 0$ (index 5 appears), and $x_3 = 1$. $\qquad\square$

$v(x_1, x_3)$	10100000
$v(x_1, \bar{x}_2)$	10001000
$v(x_3, \bar{x}_2)$	00101000
$v(x_1, 8)$	10000001
$v(x_3, 8)$	00100001
$v(\bar{x}_2, 8)$	00001001
$v(x_1, \bar{x}_1)$	10010000
$v(x_2, \bar{x}_2)$	01001000
$v(x_3, \bar{x}_3)$	00100100
$v(7, 8)$	00000011
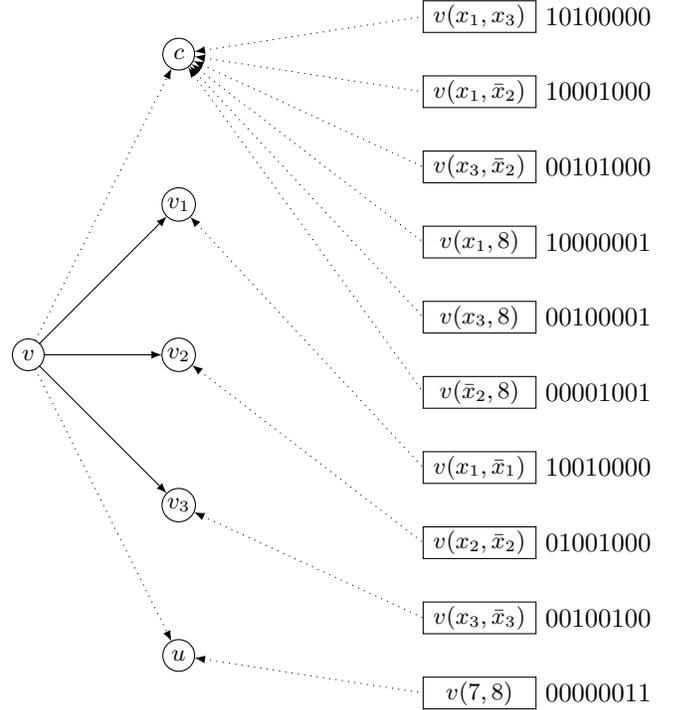
Figure 6: Example of 3SAT reduction to (k, t)-representation.