# Supporting Information

## Anderson et al. 10.1073/pnas.1421853112

### Proof of Theorem 1

***Proof:*** Let $(\mathbf{y},\mathbf{z})$ be the point for which we must determine whether the constrain in Eq. **7** is satisfied. Following the well-known procedure for PC-TSP, first we form a directed weighted graph $\overline{G}=(\overline{V},\overline{E},\overline{\mathbf{w}})$ where $\overline{V}=\{s\}\cup V$ where $s$ is an extra node, $\overline{E}=E\cup\{(s,n)|n\in N\}$, and weights $\overline{w}_e$ for $e\in\overline{E}$ are given by

$$\overline{w}_e=\begin{cases} y_e & e\in E,\\ 1 & \text{otherwise},\end{cases}$$

(the edges with $\overline{w}_e=1$ each go from the super source to a node in $N$).

Then, for every $v\in P$ where $f_v^i>0$ we solve the max flow min cut problem with source $s$ and sink $v$. If we find a cut of weight less than $f_v^i$, then by taking $S$ to be the set of nodes on the sink side of the cut we have found a violated constraint. As we are optimizing over all cuts separating $v$ from the super source and then checking all $v$ we in fact check all of the constraints in Eq. **7**.

***Proof of Theorem 2:*** Before proving the result, we introduce two auxiliary IP formulations.

### Subtour Elimination Formulation

We propose an alternative formulation on the same set of variables as the PC-TSP formulation, called the subtour formulation. The name is derived from the subtour elimination formulation of the TSP, as in ref. 1. In the subtour elimination formulation, all of the constraints are the same as the PC-TSP formulation except that Eq. **7** is replaced by

$$\sum_{e\in E(S)} y_e+\sum_{\substack{D\in\mathcal{C}_k\\ V(D)\subseteq S}}(|D|-1)z_D + \sum_{\substack{D\in\mathcal{C}_k\\ V(D)\nsubseteq S}}|D\cap E(S)|z_D\le |S|-1\quad S\subset P. \qquad \textbf{[S1]}$$

The idea of the constraint is that subset $S$ of the nodes, the number of edges used to make chains, should be at most $|S|-1$. The IP formulation would still be correct without the second and third sums; however, their addition strengths the inequality.

### Cycles Formulation

We propose another alternative formulation on the same set of variables as the PC-TSP and subtour formulations, the cycles formulation. In the cycles formulation all of the constraints are the same as in the PC-TSP and subtour formulations except that Eq. **2** is replaced by

$$\sum_{e\in C} y_e+\sum_{\substack{D\in\mathcal{C}_k\\ D\neq C}}|D\cap C|z_D\le |C|-1\quad C\in\mathcal{C}. \qquad \textbf{[S2]}$$

The idea of the constraint is to prevent the edges of $C$ from forming a cycle, unless the variable $z_C$ is used (should the variable exist). Obviously, if $y_e=1$ for $e\in C$, they would form a cycle, but the constraint prevents this. Again, the IP formulation would still be correct without the second sum in the constraint, but including this sum makes the formulation stronger. In Fig. S6 we give an example of an instance where, if the second sum were not included, the cycle formulation would be weaker than the recursive formulation.

### Proof of Result

Let $Z_{\text{cyc}}$ and $Z_{\text{sub}}$ be the value of the LP relaxation for the cycle and subtour formulations of the KEP, respectively. Let $P_{\text{rec}}$, $P_{\text{cyc}}$, $P_{\text{sub}}$, and $P_{\text{tsp}}$ be the polyhedrons for the LP relaxations of each formulation. We will now show

$$Z_{\text{tsp}}\prec Z_{\text{sub}}\prec Z_{\text{cyc}}\preceq Z_{\text{rec}}.$$

Because the relations above are transitive, this will imply *Theorem 2*.

***Proof:*** First, we show that $P_{\text{tsp}}\subseteq P_{\text{sub}}$, which immediately implies that $Z_{\text{tsp}}\preceq Z_{\text{sub}}$, because the two formulations share the same objective function. It suffices to show that each of the subtour elimination constraints from Eq. **8** are implied by the entire cut set formulation. Fix $S\subseteq P$ and assume that $\mathbf{y}$ is feasible for the cut set formulation. Fix some $u\in S$. First, we claim that

$$\sum_{\substack{D\in\mathcal{C}_k\\ V(D)\subseteq S}}(|D|-1)z_D+\sum_{\substack{D\in\mathcal{C}_k\\ V(D)\nsubseteq S}}|D\cap E(S)|z_D \le \sum_{\substack{D\in\mathcal{C}_k\\ V(D)\cap S\neq\emptyset}}(|V(D)\cap S|-1)z_D \qquad \textbf{[S3]}$$

$$\le\sum_{\substack{v\in S\\ v\neq u}}\sum_{D\in\mathcal{C}_k(v)} z_D. \qquad \textbf{[S4]}$$

To justify Eq. **S3**, observe that for cycles $D$ such that $V(D)\subseteq S$ we immediately have $|D|=|V(D)|=|V(D)\cap S|$, so for these $z_D$ terms, we have $|D|-1=|V(D)\cap S|-1$. For $D$ such that $V(D)\nsubseteq S$, we have two cases:

- If $V(D)\cap S=\emptyset$, then $D\cap E(S)=\emptyset$ as well, so these terms can be dropped.
- If $D$ has $\ell$ vertices in $S$, where $0<\ell<|D|$, then at most $\ell-1$ of the edges of $D$ will have both endpoints in $S$.

Thus, Eq. **S3** has been shown. To justify Eq. **S4**, by a simple counting argument we have the following:

- If $u\notin V(D)$, then the term $z_D$ will appear $|V(D)\cap S|$ times in Eq. **S4.**
- If $u\in V(D)$, then the term $z_D$ will appear $|V(D)\cap S|-1$ times in Eq. **S4.**

Thus, Eq. **S4** has been shown. Applying this inequality, we now have

$$\begin{aligned}&\sum_{e\in E(S)} y_e+\sum_{\substack{D\in\mathcal{C}_k\\ V(D)\subseteq S}}(|D|-1)z_D+\sum_{\substack{D\in\mathcal{C}_k\\ V(D)\nsubseteq S}}|D\cap E(S)|z_D\\ &\le\sum_{e\in E(S)} y_e+\sum_{\substack{v\in S\\ v\neq u}}\sum_{D\in\mathcal{C}_k(v)} z_D\\ &=\sum_{v\in S} f_v^i-\sum_{e\in\delta^-(S)} y_e+\sum_{\substack{v\in S\\ v\neq u}}\sum_{D\in\mathcal{C}_k(v)} z_D \qquad \textbf{[S5]}\\ &=f_u^i-\sum_{e\in\delta^-(S)} y_e+\sum_{\substack{v\in S\\ v\neq u}}\left(f_v^i+\sum_{D\in\mathcal{C}_k(v)} z_D\right)\\ &\le\sum_{\substack{v\in S\\ v\neq u}}\left(f_v^i+\sum_{D\in\mathcal{C}_k(v)} z_D\right) \qquad \textbf{[S6]}\\ &\le|S|-1, \qquad \textbf{[S7]}\end{aligned}$$

where Eq. **S5** follows as for a set of nodes $S$; all edges incoming to a node in $S$ have their other endpoint either in $S$ or outside of $S$. Eq. **S6** follows from applying Eq. **S2** (multiplied by $-1$) for the set $S$ and the vertex $u$, and Eq. **S7** follows from applying the upper bound from the flow constraint in Eq. **S6** $|S|-1$ times.

Next, we show that $P_{\text{sub}} \subseteq P_{\text{cyc}}$ and thus $Z_{\text{sub}} \leq Z_{\text{cyc}}$. It suffices to show that for any cycle $C$ Eq. **S9** is directly implied by Eq. **S8** taking $S = V(C)$. To bound the first term of the left-hand side of Eq. **S9**, we have

$$\sum_{e \in C} y_e \leq \sum_{e \in E(S)} y_e.$$

For the second term, we will partition $D \in \mathcal{C}_k, D \neq C$ into two sets, those where $V(D) \subseteq S$ and $D \neq C$, or those where $V(D) \nsubseteq S$, that is,

$$\sum_{\substack{D \in \mathcal{C}_k \\ D \neq C}} |D \cap C| z_D = \sum_{\substack{D \in \mathcal{C}_k \\ V(D) \subseteq S \\ D \neq C}} |D \cap C| z_D + \sum_{\substack{D \in \mathcal{C}_k \\ V(D) \nsubseteq S}} |D \cap C| z_D.$$

For the first sum, we have $|D \cap C| \leq |D| - 1$, because $D \neq C$ (and $D \not\subset C$ because both $D$ and $C$ are simple cycles). Thus,

$$\sum_{\substack{D \in \mathcal{C}_k \\ V(D) \subseteq S \\ D \neq C}} |D \cap C| z_D \leq \sum_{\substack{D \in \mathcal{C}_k \\ V(D) \subseteq S \\ D \neq C}} (|D| - 1) z_D \leq \sum_{\substack{D \in \mathcal{C}_k \\ V(D) \subseteq S}} (|D| - 1) z_D.$$

For the second sum, because $C \subset E(S)$, we have $|D \cap C| \leq |D \cap E(S)|$ for all $D \in \mathcal{C}_k$, and thus

$$\sum_{\substack{D \in \mathcal{C}_k \\ V(D) \nsubseteq S}} |D \cap C| z_D \leq \sum_{\substack{D \in \mathcal{C}_k \\ V(D) \nsubseteq S}} |D \cap E(S)| z_D.$$

Putting everything together then applying Eq. **S8** we have

$$\begin{aligned} &\sum_{e \in C} y_e + \sum_{\substack{D \in \mathcal{C}_k \\ D \neq C}} |D \cap C| z_D \\ &\quad \leq \sum_{e \in E(S)} y_e + \sum_{\substack{D \in \mathcal{C}_k \\ V(D) \subseteq S}} (|D| - 1) z_D + \sum_{\substack{D \in \mathcal{C}_k \\ V(D) \nsubseteq S}} |D \cap E(S)| z_D \\ &\quad \leq |S| - 1 = |C| - 1, \end{aligned}$$

showing the claim.

To show that $Z_{\text{cyc}} \preceq Z_{\text{rec}}$, consider $(\mathbf{y}^*, \mathbf{z}^*) \in P_{\text{cyc}}$ that is optimal for the cycle formulation (the values of $f_v^i$ and $f_v^o$ are implied by $\mathbf{y}^*$). If we let

$$x_e = y_e^* + \sum_{C \in \mathcal{C}_k,\ e \in C} z_C^*,$$

then we claim that $\mathbf{x} \in P_{\text{rec}}$ (again with the values of the flow variables being determined by $\mathbf{x}$). To show this, it suffices to verify Eqs. **S2–S4** hold for $\mathbf{x}$. To obtain Eq. **S2**

$$\sum_{e \in \delta^+(v)} x_e = \sum_{e \in \delta^+(v)} \left( y_e^* + \sum_{C \in \mathcal{C}_k,\ e \in C} z_C^* \right) \quad \textbf{[S8]}$$

$$= \sum_{e \in \delta^+(v)} y_e^* + \sum_{C \in \mathcal{C}_k(v)} z_C^* \quad \textbf{[S9]}$$

where in Eq. **S8** we applied the definition of $x_e$, and in Eq. **S9** we used that $\mathcal{C}_k(v)$, the set of cycles hitting $v$, is equal to the disjoint union over all $e$ going out of $v$ of the set of cycles containing $e$ (the union is disjoint as each cycle contains exactly one edge out of $v$). Likewise, we have

$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^-(v)} y_e^* + \sum_{C \in \mathcal{C}_k(v)} z_C^*.$$

Thus, Eq. **S6** from the cycles formulation implies Eq. **S2** in the recursive formulation. An analogous argument immediately gives us Eq. **S3** as well. Finally, to obtain Eq. **S4** we have for any cycle $C$ with $|C| > k$,

$$\begin{aligned} \sum_{e \in C} x_e &= \sum_{e \in C} \left( y_e^* + \sum_{\substack{D \in \mathcal{C}_k \\ e \in D}} z_e^* \right) \\ &= \sum_{e \in C} y_e^* + \sum_{\substack{D \in \mathcal{C}_k \\ D \neq C}} |D \cap C| z_D \end{aligned} \quad \textbf{[S10]}$$

$$\leq |C| - 1, \quad \textbf{[S11]}$$

where in Eq. **S10** we are counting, and using that $|C| > k$ implies that there is no $D \in \mathcal{C}_k$ such that $D = C$, and in Eq. **S11** we are applying Eq. **S9**. Thus, we conclude that $\mathbf{x}$ is feasible. Using feasibility, we can obtain the result as follows:

$$\begin{aligned} Z_{\text{rec}} &\geq \sum_{e \in E} c_e x_e \\ &= \sum_{e \in E} c_e \left( y_e^* + \sum_{\substack{C \in \mathcal{C}_k \\ e \in C}} z_e^* \right) \\ &= \sum_{e \in E} c_e y_e^* + \sum_{C \in \mathcal{C}_k} c_C z_e^* \\ &= Z_{\text{cyc}}. \end{aligned}$$

In Fig. S7 we give a family of problem instances where $Z_{\text{tsp}} < Z_{\text{sub}}$. In Fig. S8 we give an instance where $Z_{\text{sub}} < Z_{\text{cyc}}$.

**The KEP with Bounded Chain Lengths.** We show how to adapt the PC-TSP formulation to allow for a maximum chain length $\ell$, although the technique would work for any of the four formulations presented (the adaptation is trivial for the cycle and subtour formulations). For each NDD $n \in N$ and each edge $e \in E$, we introduce auxiliary edge variables $y_e^n$ and likewise $f_v^{i,n}$ and $f_v^{o,n}$ indicating flow that must begin at $n$. The formulation becomes

$$\max \quad \sum_{e \in E} w_e y_e + \sum_{C \in \mathcal{C}_k} w_C z_C$$

$$(\mathbf{y}, \mathbf{z}, \mathbf{f}^i, \mathbf{f}^o) \in P_{\text{tsp}} \quad \textbf{[S12]}$$

$$\sum_{n \in N} y_e^n = y_e \quad e \in E$$

$$\sum_{e \in E} y_e^n \leq \ell \quad n \in N \quad \textbf{[S13]}$$

$$\sum_{e \in \delta^-(v)} y_e^n = f_v^{i,n} \quad v \in V,\ n \in N \quad \textbf{[S14]}$$

$$\sum_{e \in \delta^+(v)} y_e^n = f_v^{o,n} \quad v \in V,\ n \in N \quad \textbf{[S15]}$$

$$f_v^{o,n} \leq f^{i,v} \leq 1 \quad v \in V,\ n \in N \quad \textbf{[S16]}$$

$$y_e \in \{0,1\} \quad e \in E$$
$$z_C \in \{0,1\} \quad C \in \mathcal{C}_k$$
$$y_e^n \in \{0,1\} \quad e \in E, \quad n \in N.$$

The new constraints are briefly explained as follows. From Eq. **S12** we have that each edge used ($y_e$) must be part of a chain beginning at some NDD $n$. From Eq. **S13** we obtain that each chain can use at most $\ell$ edges, thus giving the maximum chain length. In Eqs. **S14** and **S15** we just define auxiliary variables denoting whether an edge used in a chain starting at n comes into/out of $v$. Finally, in Eq. **S16** we enforce that the edges used in the chain starting at $n$ are consecutive. The remaining constraints are exactly the same as the PC-TSP constraints with no maximum chain length.

**Stochastic Optimization for the KEP.** Here we present a general framework for dealing with the possibility that after an edge is selected it might become ineligible for the matching, an event we refer to as an "edge failure." Edge failures occur commonly in practice for a variety of reasons (e.g., a donor backs out, a patient dies, or a biological incompatibility is discovered).

We propose a two-phase system for planning exchanges that anticipates edge failures occurring at random and plans to maximize the number of transplants performed once the failed edges have been identified and removed. In the first phase, a subset of the edges in the graph are selected to be tested for edge failures. Operational constraints restrict this set, where the basic idea is that it is not practical to check all of the edges. Some natural examples of phase-one edge sets to test include the following:

- Use at most $m$ edges in phase one.
- Each node has in-degree at most $m_i$ and out-degree at most $m_o$.
- The edges used in phase one must be a feasible solution to the KEP.

The only restriction on the rule used to select phase one edges is that there exists a polyhedron $P$ such that $\mathbf{y} \in P \cap \mathbb{Z}^{|E|}$ iff $\mathbf{y}$ corresponds to a valid set of phase one edges (i.e., the set of phase-one edges must be describable as a mixed-integer program). After the phase-one selections are made, we learn which of the edges that we tested in phase one failed, and in phase two we solve the regular KEP using only edges that we checked and that did not fail in phase one. Because we do not know which edges will fail before we make our phase-one decision, we use the objective of maximizing the expected weight of our phase-two KEP solution when picking our phase-one solution. Next, we describe the probabilistic framework we use for edge failures, and then the computational technique used to compute our phase-one solution.

We assume that there is a family of random variables $X_e$ for $e \in E$, taking the value one if the edge $e$ can be used in the matching and zero otherwise. We make no assumptions about the independence structure of the variables $X_e$. However, we do assume that we can jointly sample the vector of $X_e$ variables.

We now define a two-stage stochastic integer optimization problem. We have decision variables $y_e$ for $e \in E$ that indicate the edges we wish to test in stage one. In stage two, we observe our realization $\omega \in \Omega$ of $X_e(\omega)$ for the edges where $y_e = 1$ (the edges we tested), and then we form an optimal cycle packing using only edges that we tested in phase one and where $X_e(\omega) = 1$. We select our phase-one edges $\mathbf{y}$, integer and in $P$, to maximize the expected size of the phase-two packing.

This problem can be solved using the method of sample average approximation, as described and mathematically justified in refs. 2–4. Suppose that we sample the vector of $X_e$ jointly $n$ times, and let $x_e^j$ for $j = 1, \ldots, n$ be the realization of $X_e$ in the $j$th sample. Let $y_e^j$ be one if we use edge $e$ in realization $j$ and zero otherwise, and likewise let $z_C^j$ be one if we use cycle $C$ in the $j$th realization. Let $P_{\text{tsp}}^j$ be the cut set polyhedron on the variables $y_e^j$ and $z_C^j$. Our formulation is then as follows:

$$\max \quad \sum_{j=1}^{n} \left( \sum_{e \in E} c_e y_e^j + \sum_{C \in \mathcal{C}_k} c_C z_C^j \right) \qquad \textbf{[S17]}$$

$$\begin{aligned}
\text{s.t.} \quad & \mathbf{y} \in P, \\
& (\mathbf{y}^j, \mathbf{z}^j) \in P_{\text{tsp}}^j, \\
& y_e^j \le y_e \qquad e \in E, j = 1, \ldots, n, \\
& y_e^j \le x_e^j \qquad e \in E, j = 1, \ldots, n, \\
& z_C^j \le y_e \qquad C \in \mathcal{C}_k, e \in C, j = 1, \ldots, n, \\
& z_C^j \le x_e^j \qquad C \in \mathcal{C}_k, e \in C, j = 1, \ldots, n, \\
& y_e \in \{0,1\} \qquad e \in E, \\
& y_e^j \in \{0,1\} \qquad e \in E, j = 1, \ldots, n, \\
& z_C^j \in \{0,1\} \qquad C \in \mathcal{C}_k, j = 1, \ldots, n.
\end{aligned}$$

This model has a few very attractive features. First, it allows for a general probabilistic model for edge failures, which in practice should be much more accurate than simply independently and identically distributed edge failures. For example:

- If an edge failed because the donor or receiver became ill or backed out, then all edges involving that donor/receiver would be ruled out simultaneously.
- If an edge failed because a receiver developed a new HLA antibody, then all edges from donors with that HLA antigen would fail simultaneously.
- If an edge failed because a doctor or transplant center deemed a donor to be of inadequate quality for the recipient (e.g., too old), then possibly other edges pointing to the same doctor/transplant center would fail, but not necessarily all of them, because a highly sensitized recipient may have lower standards than a standard recipient.

Clearly, a very sophisticated model could be made to predict edge failures. Further, it will likely be easier to draw samples from such a model than to explicitly work out the joint distribution of edge failures.

Another good feature of this model is that we have a great deal of flexibility in choosing $P$ (the set of edges we are allowed to pick in phase one). Our flexibility in choosing $P$ allows us to adapt to various operational constraints of actually running a kidney exchange. Additionally, we can use $P$ to try and influence "agents" (e.g., donors, recipients, doctors, hospitals, and transplant centers) into taking actions that maximize global welfare. For example, if we select more than one incoming edge to a node in phase one, then the receiver, the doctor, the hospital, and the transplant center may be incentivized to reject the worse of the two edges to try and get a higher quality donor. One very simple fix is to restrict the edges tested in phase one to give each node an in-degree of at most one. Then as no one will receive multiple offers, no one will be incentivized to turn down a kidney they otherwise would have accepted.

Finally, note that it is at times desirable to add additional decision variables to the phase-one problem. For example, if we were to restrict our phase-one solution to be a feasible solution to the KEP, while we could take $P = P_{\text{rec}}$, it is computationally more efficient to use the PC-TSP formulation instead. One way of accomplishing this is as follows: Add decision variables $\tilde{y}_e$ for $e \in E$ and $\tilde{z}_C$ for each cycle $C \in \mathcal{C}_k$, let

$$y_e = \tilde{y}_e + \sum_{\substack{C \in \mathcal{C}_k \\ e \in C}} \tilde{z}_C,$$

and then take $P$ to be the PC-TSP polyhedron applied to $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$, along with the constraint above relating $\mathbf{y}$ to $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$. Further, note that the cut set constraints for the $P^j_{\text{tsp}}$ polyhedrons would automatically be implied by the cut set constraints from $P_{\text{tsp}}$ on $(\tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ and thus could be eliminated.

Our model and can easily solve problems with up to 30 scenarios on a desktop computer. Various heuristics designed for the sample average approximation approach can provide close to optimal results for larger problems. One may also consider solving a fully stochastic and dynamic optimization model; however, such a model is not tractable because it would include an infinite horizon stochastic dynamic program with a state for every possible graph and a large decision space for every state. One interesting way to tackle this is through approximate dynamic programming (see e.g., ref. 5). Previous studies have shown that a large class of dynamic algorithms do not improve outcomes significantly beyond greedy algorithms (see, e.g., ref. 6 and references therein), and thus solving the one-shot optimization problems is an important challenge.


1. Bertsimas D, Weismantel R (2005) *Optimization over Integers* (Dynamic Ideas, Belmont, MA), Vol 13.
2. Swamy C, Shmoys DB (2005) Sampling-based approximation algorithms for multi-stage stochastic optimization. *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, New York), pp 357–366.
3. Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM J Optim* 12(2):479–502.
4. Ahmed S, Shapiro A (2002) The sample average approximation method for stochastic programs with integer recourse. *SIAM J Optim* 12:479–502.
5. Hutter F, Hoos HH, Leyton-Brown K, Stützle T (2009) ParamILS: An automatic algorithm configuration framework. *J Artif Intell Res* 36(1):267–306.
6. Anderson R, Ashlagi I, Gamarnik D, Kanoria Y (2013) A dynamic model for barter exchange. ACM technical report (Assoc Computing Machinery, New York).


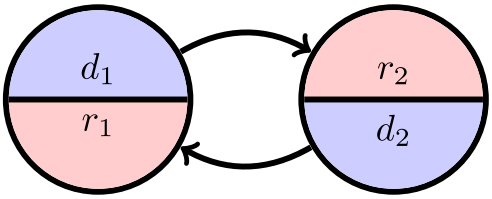


**Fig. S1.** A cyclic exchange involving two patient–donor pairs. Each pair is represented by a node, where the blue half of the node represents the donor and the red half represents the patient.

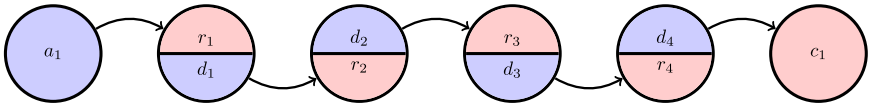


**Fig. S2.** A chain exchange involving an altruistic donor, $d_0$, four patient–donor pairs, and a patient with no donor $p_5$. Each pair is represented by a node, where the blue half of the node represents the donor and the red half represents the patient.

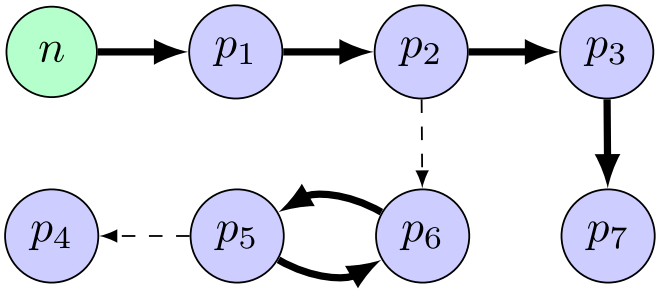


**Fig. S3.** An example of a KEP instance. The node labeled $n$ is an NDD, and the remaining nodes $p_1$ through $p_7$ correspond to patient–donor pairs. Edges indicate possible transplants from the donor in the source node to the patient in the target node. In the optimal solution for this instance, indicated by the bold edges, we form the chain $n$, $p_1$, $p_2$, $p_3$, $p_7$, and the two cycle with $p_5$ and $p_6$, leaving $p_4$ unmatched.

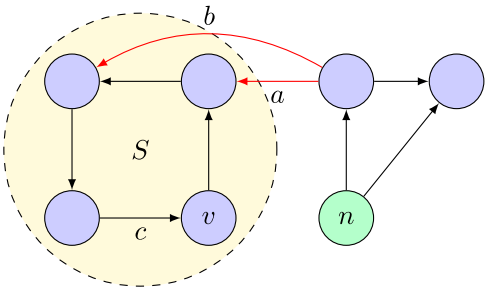


**Fig. S4.** An example of a cut set constraint. The graph contains a single NDD in green, labeled *n*. Observe that if node *v* is to be involved in any chain (i.e., $f_v^i = 1$), then we must use at least one of the edges *a* or *b* that go across the cut separating *S* from the remaining nodes and NDD.

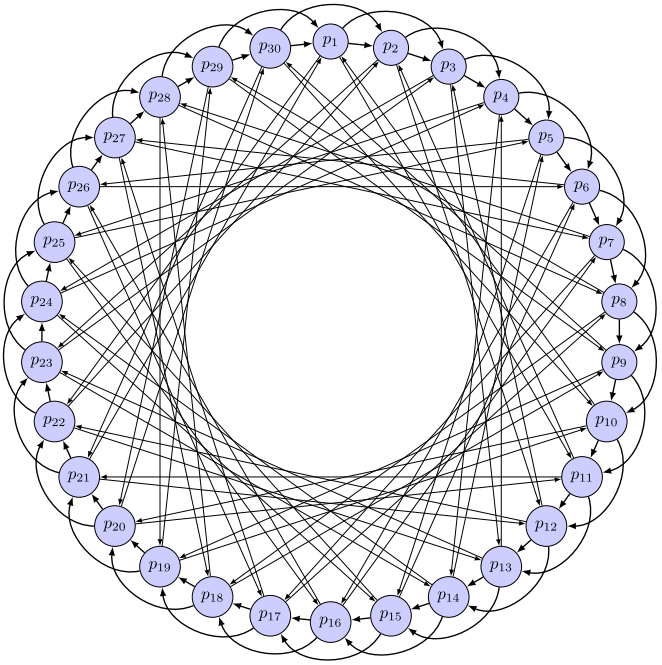


**Fig. S5.** A pathological instance of the KEP that is very difficult for the direct formulation but is solved trivially by the TSP formulation. The optimal solution is zero.

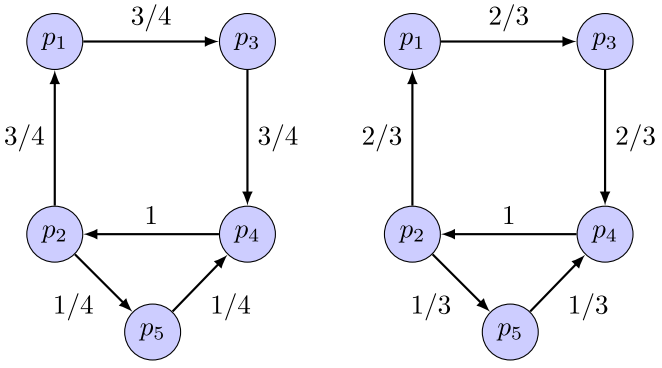


**Fig. S6.** The figure represents two fractional solutions to demonstrate the necessity of the second sum from the left-hand side of inequality **S9**. In this instance, $P = \{p_1, \ldots, p_5\}$, $N = \emptyset$, the edges are as indicated in the figure above, and all edges have weight one. The numbers next to the edges indicate fractional solutions, namely $y_e$ for the recursive formulation, and $y_e + \sum_{C \in \mathcal{C}_k, e \in C} z_C$ for the cycle formulation. Observe that the solution on the left has greater weight than the solution on the right. The solution on the left is infeasible for the recursive formulation, because the constraint on the cycle $\{(p_1,p_2),(p_2,p_3),(p_3,p_4),(p_4,p_1)\}$ is violated. The solution on the right is optimal for the recursive formulation. For the cycles formulation, letting the cycle $D = \{(3,4),(4,5),(5,3)\}$, without the second sum from the left-hand side of **S9**, we could take $z_D = 1/4$ and $y_e = 3/4$ for $e = (1,2),(2,3),(3,4),(4,1)$ and then fractional solution on the left would be feasible. This would break the result that $Z_{\text{cyc}} \preceq Z_{\text{rec}}$. However, by including the variable $z_D$ in the constraint against the four cycles, we again have that the solution on the right is optimal for the cycles formulation.

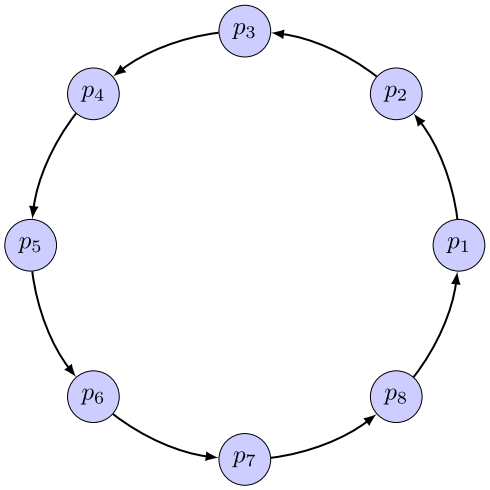


**Fig. S7.** Consider the family of problem instances on $n \geq 4$ nodes where $P = \{p_1, \ldots, p_n\}$, $N = \emptyset$, there are $n$ edges forming a single cycle of length $n$, and $w_e = 1$ for every edge. Above is the instance where $n = 8$. The optimal solution for the IP and the PC-TSP LP relaxation are both zero, but the subtour elimination LP relaxation has an optimal solution $n - 1$ [each node has $y_e = (n-1)/n$].

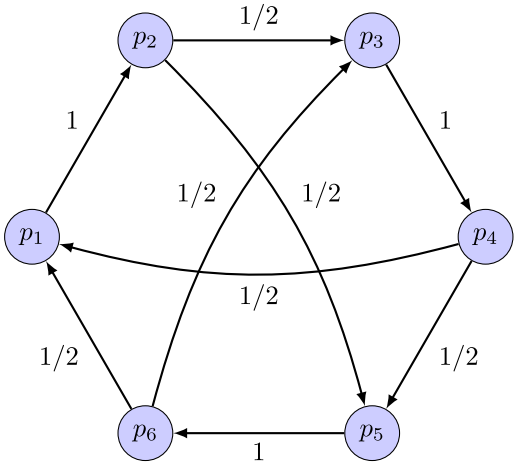


**Fig. S8.** In the instance on six nodes above, where $k = 3$, $N = \emptyset$, $P = \{p_1, \ldots, p_6\}$, and each edge has weight one, the IP optimum is zero. Taking $y_e$ to be the edge labels in the figure above, we get a feasible solution to the LP relaxation of $Z_{cyc} = 6$. However, the LP optimum for the subtour formulation is $Z_{sub} = 5$. We can attain this value by taking $y_{(i,i+1)} = 5/6$ and $y_{(6,1)} = 5/6$. To show that 5 is optimal, we apply constraints **S8** taking $S = P$, to obtain that $\sum_{e \in E(P)} y_e \leq 5$, and then observe that $\sum_{e \in E(P)} y_e$ is equal to the objective function.

**Table S1. Average number of chains of size $k$ ($k = 3, 4, 5, 6$) in random pools of various sizes and a single altruistic donor**

| Nodes | $k = 3$ | $k = 4$ | $k = 5$ | $k = 6$ |
|---|---|---|---|---|
| 150 | 4,520 | 69,780 | 1,063,727 | 16,116,117 |
| 200 | 5,147 | 99,046 | 1,884,160 | 35,304,432 |
| 250 | 15,407 | 370,071 | 8,807,015 | 207,347,121 |

**Table S2. Additional patients matched for incremental increases in the maximum chain length**

| Measures | (3,∞), % | (4,∞), % | (5,∞), % | (6,∞), % |
|---|---|---|---|---|
| Additional highly sensitized (PRA >95) matched | 35 | 27 | 21 | 16 |
| Additional patients matched | 21 | 17 | 14 | 12 |
| Instances with more highly sensitized matched | 35 | 32 | 25 | 23 |

**Table S3. Performance of the recursive and TSP algorithms for "difficult" real-data KEP instances**

| | | | Running time, s | |
|---|---|---|---|---|
| NDDs | Patient–donor pairs | Edges | Recursive | TSP |
| 3 | 202 | 4,706 | 0.148 | 0.031 |
| 10 | 156 | 1,109 | 13.093 | 0.022 |
| 6 | 263 | 8,939 | 59.158 | 1.655 |
| 5 | 284 | 10,126 | 71.066 | 0.807 |
| 6 | 324 | 13,175 | 418.27 | 0.981 |
| 6 | 328 | 13,711 | 474.947 | 1.947 |
| 6 | 312 | 13,045 | 1,200* | 0.157* |
| 10 | 152 | 1,125 | 48.56 | 0.054 |
| 3 | 269 | 2,642 | 40.506 | 0.134 |
| 10 | 257 | 2,461 | 67.783 | 0.258 |
| 7 | 255 | 2,390 | 85.475 | 0.268 |
| 6 | 215 | 6,145 | 248.46 | 0.532 |
| 10 | 255 | 2,550 | 216.48 | 0.126 |
| 1 | 310 | 4,463 | 721.66 | 0.555 |
| 11 | 257 | 2,502 | 1,039.105 | 0.125 |
| 6 | 261 | 8,915 | 1,200 | 4.435 |
| 10 | 256 | 2,411 | 587.238 | 0.114 |
| 6 | 330 | 13,399 | 1,200 | 1.621 |
| 10 | 256 | 2,347 | 1,200 | 0.305 |
| 7 | 291 | 3,771 | 1,200* | 0.221 |
| 8 | 275 | 3,158 | 1,200* | 0.224 |
| 4 | 289 | 3,499 | 1,200* | 0.2 |
| 3 | 199 | 2,581 | 1,200* | 0.041 |
| 7 | 198 | 4,882 | 1,200* | 8.204 |
| 2 | 389 | 8,346 | 1,200* | 0.096 |

Timeouts (optimal solution not found) are indicated by an asterisk.

**Table S4. KEP on very large historical datasets with the recursive and TSP algorithms**

| | | | | Recursive algorithm | | TSP algorithm | |
|---|---|---|---|---|---|---|---|
| Instance | NDDs | Patient–donor pairs | Edges | Running time, s | RAM, GB | Running time, s | RAM, GB |
| APD | 47 | 931 | 190,820 | 1.79 | 1 | 104 | 25 |
| NKR | 162 | 1,179 | 346,608 | 3.074 | 1 | 314 | 37 |