Appendix to "Market Failure in Kidney Exchange"

Nikhil Agarwal, Itai Ashlagi, Eduardo Azevedo,
Clayton Featherstone, and Ömer Karaduman

May 21, 2018

A Proofs

A.1 Preliminary results

We begin with a lemma showing that hospital supply can be aggregated, so that hospitals behave as a single hospital that takes the aggregate cost curve into account. This is similar to standard aggregation results in neoclassical firm theory (Mas-Colell et al., 1995 p. 148).

Lemma A.1. *Fix a vector of rewards $\boldsymbol{p}$. Aggregate supply $\boldsymbol{S}(\boldsymbol{p})$ is the Minkowski sum of individual supply $\boldsymbol{S}^h(\boldsymbol{p})$ for all hospitals. Moreover, if there is a set of individual supply vectors $(\boldsymbol{q}^h)_{h=1}^H$ with each $\boldsymbol{q}^h \in \boldsymbol{S}^h(\boldsymbol{p})$, then*

$$\sum_{h=1}^{H} C^h(\boldsymbol{q}^h) = C\left(\sum_{h=1}^{H} \boldsymbol{q}^h\right). \tag{A1}$$

Proof. Note that

$$\begin{aligned}
\max_{(q^h)_{h=1}^H} \boldsymbol{p}\cdot\left(\sum_{h=1}^{H}\boldsymbol{q}^h\right) - \sum_{h=1}^{H} C^h(\boldsymbol{q}^h) \quad &= \quad \max_{\boldsymbol{q}\in\mathbb{R}_+^I} \max_{(\boldsymbol{q}^h)_{h=1}^H} \boldsymbol{p}\cdot\left(\sum_{h=1}^{H}\boldsymbol{q}^h\right) - \sum_{h=1}^{H} C^h(\boldsymbol{q}^h), \\
&\qquad\qquad \text{s.t.} \quad \sum_{h=1}^{H}\boldsymbol{q}^h = \boldsymbol{q}, \\
&= \quad \max_{\boldsymbol{q}\in\mathbb{R}_+^I} \boldsymbol{p}\cdot\boldsymbol{q} - C(\boldsymbol{q}).
\end{aligned} \tag{A2}$$

Consider $\boldsymbol{q}_0 = \sum_{h=1}^H \boldsymbol{q}_0^h$ with each $\boldsymbol{q}_0^h$ in $\boldsymbol{S}^h(\boldsymbol{p})$. By the optimality of aggregate cost, we have that

$$\boldsymbol{p}\cdot\boldsymbol{q}_0 - C(\boldsymbol{q}_0) \geq \boldsymbol{p}\cdot\left(\sum_{h=1}^{H}\boldsymbol{q}_0^h\right) - \sum_{h=1}^{H} C^h(\boldsymbol{q}_0^h). \tag{A3}$$

Optimality of the $\boldsymbol{q}_0^h$ implies that the right-hand side of this inequality attains the maximum in the left-hand side of equation (A2). Hence,

$$\boldsymbol{p}\cdot\boldsymbol{q}_0 - C(\boldsymbol{q}_0) \geq \max_{\boldsymbol{q}\in\mathbb{R}_+^I} \boldsymbol{p}\cdot\boldsymbol{q} - C(\boldsymbol{q}),$$

so that $\boldsymbol{q}_0$ is in $\boldsymbol{S}(\boldsymbol{p})$. Inequality (A3) holds as an equality, which implies (A1) as desired.

Conversely, consider $\boldsymbol{q}_0$ in $\boldsymbol{S}(\boldsymbol{p})$. Equation (A2) implies that

$$\boldsymbol{p}\cdot\boldsymbol{q}_0 - C(\boldsymbol{q}_0) = \max_{(\boldsymbol{q}^h)_{h=1}^H} \boldsymbol{p}\cdot\left(\sum_{h=1}^{H}\boldsymbol{q}^h\right) - \sum_{h=1}^{H} C^h(\boldsymbol{q}^h).$$

Let $(\boldsymbol{q}_0^h)$ be supply vectors that minimize the sum of costs conditional on total supply being $\boldsymbol{q}_0$. Then

$$\boldsymbol{p}\cdot\boldsymbol{q}_0 - C(\boldsymbol{q}_0) = \boldsymbol{p}\cdot\left(\sum_{h=1}^{H}\boldsymbol{q}_0^h\right) - \sum_{h=1}^{H} C^h(\boldsymbol{q}_0^h).$$

Therefore, each $\boldsymbol{q}_0^h$ is in $\boldsymbol{S}^h(\boldsymbol{q}^h)$.

□

Given the assumptions in the body of the paper, we can treat aggregate supply and aggregate inverse supply as functions rather than correspondences.

Lemma A.2. *The aggregate supply $\boldsymbol{S}(\boldsymbol{p})$ is a single-valued correspondence.*

Proof. Since we assume that the maximum of each hospital's objective is attained for some quantity for every vector of rewards, Lemma A.1 implies that $\boldsymbol{S}(\boldsymbol{p})$ is non-empty for any rewards vector $\boldsymbol{p}$. Because $C(\boldsymbol{q})$ is strictly convex, $\boldsymbol{S}(\boldsymbol{p})$ is the unique maximizer of the function $\boldsymbol{p}\cdot\boldsymbol{q} - C(\boldsymbol{q})$. □

Lemma A.3. *For any strictly positive $\boldsymbol{q}$, $\boldsymbol{P}_S(\boldsymbol{q})$ is single-valued and $\boldsymbol{P}_S(\boldsymbol{q}) = \boldsymbol{\nabla C}(\boldsymbol{q})$.*

Proof. Since $\boldsymbol{q}$ is interior and $C(\boldsymbol{q})$ is smooth and strictly convex, the first-order necessary conditions are also sufficient for a maximum of the aggregate supply program. □

A.2 Proof of the Main Theorem

Proof of Theorem 1. Let $(\boldsymbol{p}^*, (\boldsymbol{q}^{h*})_{h=1,\ldots,H})$ maximize hospital welfare subject to all hospitals choosing $\boldsymbol{q}^h$ in $S^h(\boldsymbol{p})$ given $\boldsymbol{p}$ and subject to allocating the number of transplants that are produced. Mathematically, the tuple $(\boldsymbol{p}^*, \boldsymbol{q}^*, (\boldsymbol{q}^{h*})_{h=1,\ldots,H})$ in $\mathbb{R}^I \times \mathbb{R}^I_+ \times \mathbb{R}^{IH}_+$ maximizes

$$f(\boldsymbol{q}) - \sum_{h=1}^{h} C^h(\boldsymbol{q}^h)$$

subject to

$$\boldsymbol{q} = \sum_{h=1}^{H}\boldsymbol{q}^h, \tag{A4}$$

to each h

$$\boldsymbol{q}^h \in \boldsymbol{S}^h(\boldsymbol{p}), \tag{A5}$$

and to

$$\boldsymbol{p}\cdot\boldsymbol{q} = f(\boldsymbol{q}). \tag{A6}$$

Lemma A.1 implies that this maximization problem is equivalent to finding a pair $(\boldsymbol{p}^*, \boldsymbol{q}^*)$ in $\mathbb{R}^I \times \mathbb{R}^I_+$ that maximizes

$$f(\boldsymbol{q}) - C(\boldsymbol{q}) \tag{A7}$$

subject to

$$\boldsymbol{q} \in \boldsymbol{S}(\boldsymbol{p}), \tag{A8}$$

and to (A6).

By Lemma A.3, constraint (A8) is equivalent to $\boldsymbol{p} = \boldsymbol{\nabla C}(\boldsymbol{q})$. Thus, $\boldsymbol{q}^*$ maximizes (A7) in $\mathbb{R}^I_{++}$ subject to

$$\boldsymbol{\nabla C}(\boldsymbol{q}) \cdot \boldsymbol{q} = f(\boldsymbol{q}).$$

The production function and aggregate cost function are smooth, and $\mathbb{R}^I_{++}$ is an open set. Therefore, the Lagrange multiplier theorem implies that there exists λ such that $\boldsymbol{q}^*$ maximizes

$$f(\boldsymbol{q}) - C(\boldsymbol{q}) + \lambda \cdot \{f(\boldsymbol{q}) - \boldsymbol{\nabla C}(\boldsymbol{q}) \cdot \boldsymbol{q}\}.$$

Setting the derivative equal to zero, we have

$$\boldsymbol{\nabla f}(\boldsymbol{q}) - \boldsymbol{\nabla C}(\boldsymbol{q}) = \frac{\lambda}{1+\lambda} \boldsymbol{q'D^2C}(\boldsymbol{q}).$$

To solve for the Lagrange multiplier, we multiply by $\boldsymbol{q}$ on the right, and use the equality $\boldsymbol{\nabla C} \cdot \boldsymbol{q} = f$. We have

$$\boldsymbol{\nabla f}(\boldsymbol{q}) \cdot \boldsymbol{q} - f(\boldsymbol{q}) = \frac{\lambda}{1+\lambda} \boldsymbol{q'D^2C}(\boldsymbol{q})\boldsymbol{q}.$$

Therefore,

$$\frac{\boldsymbol{\nabla f}(\boldsymbol{q}) \cdot \boldsymbol{q} - f(\boldsymbol{q})}{\boldsymbol{q'D^2C}(\boldsymbol{q})\boldsymbol{q}} = \frac{\lambda}{1+\lambda}.$$

Substituting λ we get the final formula,

$$\boldsymbol{\nabla C}(\boldsymbol{q}) = \boldsymbol{\nabla f}(\boldsymbol{q}) - \left(\frac{\boldsymbol{\nabla f}(\boldsymbol{q}) \cdot \boldsymbol{q} - f(\boldsymbol{q})}{\boldsymbol{q'D^2C}(\boldsymbol{q})\boldsymbol{q}} \right) \cdot (\boldsymbol{q'D^2C}(\boldsymbol{q})).$$

Lemma A.3 allows us to replace $\boldsymbol{D}^2 C(\boldsymbol{q})$ with $\boldsymbol{DP}_S(\boldsymbol{q})$.

Part 1 of the theorem follows by substituting $\boldsymbol{p} = \boldsymbol{\nabla C}(\boldsymbol{q})$. Part 2 follows because, with constant returns to scale, $\boldsymbol{\nabla f}(\boldsymbol{q}) \cdot \boldsymbol{q} - f(\boldsymbol{q}) = 0$ so that $\lambda = 0$. Part 3 follows because, with no agency problems, welfare and social welfare coincide.

□

A.3 Additional Proofs

Proof of Proposition 1. In what follows, we are considering an asymptotic where $\boldsymbol{q}_0 \to \boldsymbol{q}^*$. Note that, by Lemma (A.1) and the fact that $\boldsymbol{q}^*$ is strictly positive, $\boldsymbol{p}^* = \boldsymbol{\nabla C}(\boldsymbol{q}^*)$. Further, since $\boldsymbol{q}_0$ is positive, we have that $\boldsymbol{p}_0 = \boldsymbol{\nabla C}(\boldsymbol{q}_0)$.

Part 1: Approximation of $\boldsymbol{q}^* - \boldsymbol{q}_0$:

Theorem (1) shows that

$$\boldsymbol{\nabla f}(\boldsymbol{q}^*) - \boldsymbol{\nabla C}(\boldsymbol{q}^*) = \mathbf{0}.$$

Taking a Taylor expansion of the left-hand side about $\boldsymbol{q_0}$, we have

$$\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{\nabla C}(\boldsymbol{q}_0) + (\boldsymbol{q}^* - \boldsymbol{q}_0)' \left[\boldsymbol{D^2 f}(\boldsymbol{q}_0) - \boldsymbol{D^2 C}(\boldsymbol{q}_0)\right] + \boldsymbol{\varepsilon_1} = \mathbf{0}, \tag{A9}$$

where $\|\boldsymbol{\varepsilon_1}\|$ is $o(\|\boldsymbol{q}^* - \boldsymbol{q}_0\|)$. Therefore,

$$\boldsymbol{q}^* - \boldsymbol{q}_0 = -\left[\boldsymbol{D^2 f}(\boldsymbol{q}_0) - \boldsymbol{D^2 C}(\boldsymbol{q}_0)\right]^{-1} \cdot \left[\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0\right]' + \boldsymbol{\varepsilon}_2, \tag{A10}$$

where the error term

$$\boldsymbol{\varepsilon}_2 = -\left[\boldsymbol{D^2 f}(\boldsymbol{q}_0) - \boldsymbol{D^2 C}(\boldsymbol{q}_0)\right]^{-1} \cdot \boldsymbol{\varepsilon_1'}$$

has a magnitude that is $o(\|\boldsymbol{q}^* - \boldsymbol{q}_0\|)$.

Part 2: First approximation of the deadweight loss:

The deadweight loss at $\boldsymbol{q}_0$ is given by

$$DWL = f(\boldsymbol{q}^*) - C(\boldsymbol{q}^*) - [f(\boldsymbol{q}_0) - C(\boldsymbol{q}_0)].$$

A second-order Taylor expansion of $f(\boldsymbol{q}^*) - C(\boldsymbol{q}^*)$ around $\boldsymbol{q}_0$ yields that

$$DWL = [\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{\nabla C}(\boldsymbol{q}_0)] \cdot (\boldsymbol{q}^* - \boldsymbol{q}_0) + \frac{1}{2} \cdot (\boldsymbol{q}^* - \boldsymbol{q}_0)' \cdot \left[\boldsymbol{D^2 f}(\boldsymbol{q}_0) - \boldsymbol{D^2 C}(\boldsymbol{q}_0)\right] \cdot (\boldsymbol{q}^* - \boldsymbol{q}_0) + \varepsilon_3,$$

where ε_3 is $o(\|\boldsymbol{q}^* - \boldsymbol{q}_0\|^2)$. We can substitute

$$(\boldsymbol{q}^* - \boldsymbol{q}_0)' \left[\boldsymbol{D^2 f}(\boldsymbol{q}_0) - \boldsymbol{D^2 C}(\boldsymbol{q}_0)\right] = -\left[\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0\right] - \boldsymbol{\varepsilon_1}$$

using equation (A9). This yields

$$DWL = \frac{1}{2} \cdot [\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0] \cdot (\boldsymbol{q}^* - \boldsymbol{q}_0) + \varepsilon_4,$$

where $\varepsilon_4 = -\boldsymbol{\varepsilon_1} \cdot (\boldsymbol{q}^* - \boldsymbol{q}_0) + \varepsilon_3$ is $o(\|\boldsymbol{q}^* - \boldsymbol{q}_0\|^2)$. This establishes the first approximation formula, as Lemma A.3 lets us replace $\boldsymbol{D^2 C}(\boldsymbol{q}_0)$ with $\boldsymbol{DP_S}(\boldsymbol{q}_0)$.

Part 3: Second approximation of the deadweight loss:

To establish the second approximation formula we substitute $\boldsymbol{q}^* - \boldsymbol{q}_0$ from equation (A10) into the first approximation to get

$$DWL = -\frac{1}{2} \cdot [\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0] \cdot \left[\boldsymbol{D^2 f}(\boldsymbol{q}_0) - \boldsymbol{D^2 C}(\boldsymbol{q}_0)\right]^{-1} \cdot [\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0]' + \frac{1}{2} \cdot [\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0] \cdot \boldsymbol{\varepsilon_2} + \varepsilon_4, \tag{A11}$$

where ε_4 is $o(\|\boldsymbol{q}^* - \boldsymbol{q}_0\|^2)$ and $\|\boldsymbol{\varepsilon_2}\|$ is $o(\|\boldsymbol{q}^* - \boldsymbol{q}_0\|)$. Since equation (A9) shows that $\|\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0\|$ is $\mathcal{O}(\|\boldsymbol{q^*} - \boldsymbol{q_0}\|)$, and the product of a $o(\|\boldsymbol{q}^* - \boldsymbol{q}_0\|)$ term and a $\mathcal{O}(\|\boldsymbol{q^*} - \boldsymbol{q_0}\|)$ term is $o\left(\|\boldsymbol{q^*} - \boldsymbol{q_0}\|^2\right)$, we conclude that $\varepsilon_5 = \frac{1}{2} \cdot [\boldsymbol{\nabla f}(\boldsymbol{q}_0) - \boldsymbol{p}_0] \cdot \boldsymbol{\varepsilon_2} + \varepsilon_4$ is $o\left(\|\boldsymbol{q^*} - \boldsymbol{q_0}\|^2\right)$. Lemma A.3 lets us replace $\boldsymbol{D^2 C}(\boldsymbol{q}_0)$ with $\boldsymbol{DP_S}(\boldsymbol{q}_0)$.

□

Proposition 2. Let $(\boldsymbol{p}^*, (\boldsymbol{q}^{h*})_{h=1,\ldots,H})$ maximize social welfare subject to all hospitals choosing supply optimally given $\boldsymbol{p}$ (breaking ties by looking at social cost) and subject to not promising more transplants than are produced. Mathematically, the tuple $(\boldsymbol{p}^*, (\boldsymbol{q}^{h*})_{h=1,\ldots,H})$, in $\mathbb{R}^I \times \mathbb{R}_+^{IH}$ maximizes

$$f(\boldsymbol{q}) - \sum_{h=1}^{H} SC^h\left(\boldsymbol{q}^h\right)$$

subject to constraints (A4), (A5), and (A6). By Lemma A.1 , the assumption that the maximum is strictly positive, and the assumption that C is smooth and convex, the maximization problem is equivalent to finding $\boldsymbol{q}^*$ in $\mathbb{R}_{++}^I$ that maximizes

$$f(\boldsymbol{q}) - SC(\boldsymbol{q})$$

subject to

$$\boldsymbol{P}_S(\boldsymbol{q}) \cdot \boldsymbol{q} = f(\boldsymbol{q}).$$

By the Lagrange multiplier theorem, there exists a multiplier λ^{SW} such that, at the optimum,

$$\boldsymbol{\nabla f}(\boldsymbol{q}) - \boldsymbol{\nabla SC}(\boldsymbol{q}) + \lambda^{SW}(\boldsymbol{\nabla f}(\boldsymbol{q}) - \boldsymbol{P}_S(\boldsymbol{q}) - \boldsymbol{q'DP_S}(\boldsymbol{q})) = \boldsymbol{0}.$$

If we use the fact that $\boldsymbol{\nabla C}(\boldsymbol{q}) = \boldsymbol{P}_S(\boldsymbol{q})$, this becomes

$$\boldsymbol{0} = \boldsymbol{\nabla f}(\boldsymbol{q}) - \boldsymbol{P}_S(\boldsymbol{q}) + (\boldsymbol{\nabla C}(\boldsymbol{q}) - \boldsymbol{\nabla SC}(\boldsymbol{q})) + \lambda^{SW}(\boldsymbol{\nabla f}(\boldsymbol{q}) - \boldsymbol{P}_S(\boldsymbol{q}) - \boldsymbol{q'DP_S}(\boldsymbol{q})),$$

$$\boldsymbol{P}_S(\boldsymbol{q}) - \boldsymbol{\nabla f}(\boldsymbol{q}) = \frac{1}{1+\lambda^{SW}}\left(\boldsymbol{\nabla C}(\boldsymbol{q}) - \boldsymbol{\nabla SC}(\boldsymbol{q})\right) - \frac{\lambda^{SW}}{1+\lambda^{SW}}\boldsymbol{q'DP_S}(\boldsymbol{q}),$$

The right-hand side of the last expression is the adjustment term $\boldsymbol{A}$. To obtain the formula for the Lagrange multiplier we multiply on the right by $\boldsymbol{q}^*$ and use the fact that $\boldsymbol{\nabla f}(\boldsymbol{q}^*) \cdot \boldsymbol{q}^* = \boldsymbol{p}^* \cdot \boldsymbol{q}^* = f(\boldsymbol{q}^*)$:

$$\lambda^{SW} = \frac{\left(\boldsymbol{\nabla C}(\boldsymbol{q}) - \boldsymbol{\nabla SC}(\boldsymbol{q})\right)\boldsymbol{q}}{\boldsymbol{q'DP_S}(\boldsymbol{q})\boldsymbol{q}}.$$

The second and third parts of the proposition follow from the formula for the Lagrange multiplier.

□

Proposition 3. The platform chooses $\boldsymbol{q}$ in $\mathbb{R}_+^I$ to maximize

$$f(\boldsymbol{q})$$

subject to

$$f(\boldsymbol{q}) = \boldsymbol{P}_{RS}(\boldsymbol{q}) \cdot \boldsymbol{q}.$$

Because the solution is interior, there exists a Lagrange multiplier λ such that

$$\boldsymbol{\nabla f}(\boldsymbol{q}) + \lambda(\boldsymbol{\nabla f}(\boldsymbol{q}) - \boldsymbol{P}_{RS}(\boldsymbol{q}) - \boldsymbol{q'DP}_{RS}(\boldsymbol{q})) = \boldsymbol{0}.$$

Substituting that the optimal rewards $\boldsymbol{p} = \boldsymbol{P}_{RS}(\boldsymbol{q})$, we obtain

$$\boldsymbol{p} = \boldsymbol{\nabla f}(\boldsymbol{q}) + \frac{1}{\lambda}\boldsymbol{\nabla f}(\boldsymbol{q}) - \boldsymbol{q'DP}_{RS}(\boldsymbol{q}).$$

To calculate the Lagrange multiplier, we right multiply by $\boldsymbol{q}$ and use $\boldsymbol{p}\cdot\boldsymbol{q} = \boldsymbol{\nabla f}(\boldsymbol{q})\cdot\boldsymbol{q} = f(\boldsymbol{q})$ to obtain

$$\lambda = \frac{f(\boldsymbol{q})}{\boldsymbol{q'DP}_{RS}(\boldsymbol{q})\boldsymbol{q}}.$$

These two formulas imply part 1 of the proposition statement. The observation in part 2 follows directly from the formula in part 1.

□

B Data Appendix

This study used five main anonymized data sets: a database of all kidney exchange transplants done in the US from January 1, 2008 through December 4, 2014 (the OPTN transplant data), databases of all kidney exchange transplants organized by each of the three largest multi-hospital platforms in the US (the NKR, APD, and UNOS transplant data), and a database of all patient and donor registrations to the largest of those platforms (the NKR registration data).

B.1 Transplant data

In order to document the kidney exchange market, we merged the OPTN transplant data with the transplant data from NKR, APD, and UNOS. In what follows, we will describe these data and the merge procedure we used.

Obtaining the datasets The OPTN provided us with a dataset on all transplants conducted in the US, known as the Standard Transplant Analysis and Research (STAR) dataset. The STAR dataset by itself lacks two key pieces of information: the transplant hospitals where the kidney was put into the patient and removed from the donor (which we use to determine whether a transplant is internal or external) and the unacceptable antigens for the patient (which we need to measure sensitization). These supplemental pieces of information are also available from the OPTN on request. Merging is done by using OPTN identifiers. The OPTN database contains records on 4377 kidney exchange transplants.

We obtained each of the platform datasets directly from the platform. The platform datasets contain records on 1400 kidney exchange transplants in total: 1193 from NKR, 100 from APD, and 107 from UNOS.

Dataset merge algorithm In order to identify which transplants in the comprehensive OPTN database were organized by each of the three platforms, we matched records in the platform files to records in the OPTN file. However, because all datasets are anonymized, the merge must be done on the biological characteristics of each transplant's recipient and donor ans logistical information on the transplant itself.

Fortunately, we can also use the fact that the OPTN database is comprehensive while the platform database is not.[1] This implies that the transplants in the platform data are a subset of transplants in the OPTN data if record-keeping is perfect. However, without perfect record-keeping, we should be wary of false matches. Our matching procedure is designed to limit false matches by keeping only the highest quality match that meets a minimum threshold. We formalize this idea below.

Let the set of records in the platform data be D_p and the set in the OPTN data be D_o. Let the **universe** of acceptable matches of one platform record $r_p \in D_p$ to one OPTN record $r_o \in D_o$ be represented by $U \subseteq D_p \times D_o$. Define the set of **collisions** a match (r_p, r_o) has relative to some set X to be $\kappa(r_p, r_o, X) \equiv \left\{\left(r'_p, r'_o\right) \in X \mid r'_p = r_p \text{ or } r'_o = r_o\right\} \setminus \{(r_p, r_o)\}$. A set of matches $M \subseteq U$ is called a **merge** if none its matches have any collisions relative to the merge, that is, for each $(r_p, r_o) \in M$, we have $\kappa(r_p, r_o, M) = \emptyset$. Any record that is not part of a match in merge M should be interpreted as unmatched.

Now, we can discuss what makes an individual match good. Define a **ranked criterion** to be a series of N subsets of the universe $C = (C^n)_{n=0}^N$ such that $C^n \subseteq C^{n+1}$, $C^0 = \emptyset$, and $C^N = U$. The ranked criterion codifies a hierarchy of match quality levels. Further, define the **rank** of a match (r_p, r_o) to be $\rho(r_p, r_o) \equiv \min\{n \mid (r_p, r_o) \in C^n\}$.[2]

Finally, we are ready to state our notion of merge quality. Recall that the main idea is that a match of rank n should be in the merge if and only if it is the unique match of rank n or better whose component records aren't part of a better match in the merge. For a given set X and rank level n, define the **universe excluding records that matched better than n in X** to be

$$U^n(X) \equiv U \setminus \left[\bigcup_{(r_p,r_o)\in X\cap C^{n-1}} \kappa(r_p, r_o, U)\right].$$

Then, a set of matches $M \subseteq U$ is **stable relative to ranked criterion C** if it satisfies

$$(r_p, r_o) \in M \Leftrightarrow \left\{\begin{array}{c}(r_p, r_o) \in U^{\rho(r_p,r_o)}(M) \\ \text{and} \\ \kappa\left(r_p, r_o, C^{\rho(r_p,r_o)} \cap U^{\rho(r_p,r_o)}(M)\right) = \emptyset.\end{array}\right\}.$$

The first condition on the right-hand side ensures that (r_p, r_o) doesn't collide with a strictly better match in M, while the second condition ensures that (r_p, r_o) is the unique such match

[1] The OPTN is required by federal administrative law to "[m]aintain records of all transplant candidates, all organ donors and all transplant recipients" (42 C.F.R. § 121.11(a)(1)(ii)).

[2] Note that we could equivalently start with a rank function and construct C to be the lower contour sets of that function.

of its rank. Note that any set M that is stable relative to ranked criterion C is necessarily a merge (see Lemma B.4 below).

Finding a stable merge relative to some ranked criterion C is a simple matter of following Algorithm 1. It is also true that for a ranked criterion C, the stable merge is unique, so we need not worry that there is some other stable merge that we might prefer on different grounds. Proofs of these claims (which are summarized in Proposition B.1) can be found at the bottom of this subsection.

Algorithm 1. *Initialize by setting* $M^0 = \emptyset$. *Then for* $n \in \{1, \ldots, N\}$,

$$\begin{aligned} X^n &= C^n \cap U^n\left(M^{n-1}\right) \\ M^n &= M^{n-1} \cup \left\{(r_p, r_o) \in X^n \mid \kappa\left(r_p, r_o, X^n\right) = \emptyset\right\}. \end{aligned}$$

The output merge is M^N.

Basically, in Step n, the algorithm adds any match to the output set that is the unique rank-n or better match whose records are not involved in a rank $n-1$ or better match that is already in the output set.

Proposition B.1. *Algorithm 1 yields the unique set that is stable relative to ranked criterion* C. *Furthermore, this set is a merge.*

Quality of the merge We merged the OPTN and platform transplant databases using Algorithm 1. The set of acceptable matches, U, was the set of all potential matches where the transplant dates are within 31 days of each other and the ages of the donor and the recipient are each within 10 years of each other. The ranked criterion we used is defined as follows.

- C^1 is the set of all acceptable matches where the donor and recipient each match exactly on blood type, sex, the hospital where the transplant was conducted, and all six major HLA alleles (two each on the HLA-A, HLA-B, and HLA-DR loci).
- For $n \in \{2, \ldots, 5\}$, C^n is the set of all acceptable matches where the donor and recipient each match on either blood type or sex and also each match on at least $7 - n$ out of the six major HLA alleles.

Given the ranked criterion described in the previous section, the merge algorithm performed well. The percentage of platform records matched to an OPTN record was 94% overall (94% for NKR, 97% for APD, and 94% for UNOS). Moreover, the matches seem to be high quality: Table B1 reports the percentage of matches that meet various criteria.

Table B1: Agreement Between Matched Records in the Transplant Merge

Platform	Age within 5 years	Transplant date within 1 day	5 or more HLA alleles match	Blood type and gender match	Transplant hospital matches	At most 1 criterion to the left is violated	No criterion to the left is violated
NKR	97.8%	97.4%	94.9%	95.1%	97.2%	97.6%	87.6%
APD	92.8%	95.9%	93.8%	96.9%	95.9%	96.9%	90.7%
UNOS	87.1%	99.0%	96.0%	99.0%	99.0%	99.0%	83.2%
All of the above	96.6%	97.4%	94.9%	95.5%	97.3%	97.6%	89.9%

Confirming the merge is correct It was necessary to use the merge described above to match transplants in the APD and NKR databases to the OPTN transplant database, since all of these datasets are anonymized to different sets of identifiers. However, since UNOS is a contractor for the OPTN, the UNOS and OPTN databases share common identifiers that allow us to see the actual true merge of the UNOS transplant dataset to the OPTN dataset. Comparing true matches to the matches selected by our algorithm, we find that only one UNOS record was incorrectly matched. That is, our algorithm chose the correct match in the OPTN dataset for 99% of the UNOS records it matched. This gives us added confidence that our merge algorithm is working well.

Proofs concerning Algorithm 1 and stable merges To prove Proposition B.1, three lemmas are helpful. First, we show that in Algorithm 1, once a match collides in some step of the algorithm, there is no chance that the match will ever be added to the merge.

Lemma B.1. *In Algorithm 1, if* $(r_p, r_o) \in X^n$*, then any* $\left(r'_p, r'_o\right) \in \kappa\left(r_p, r_o, X^n\right)$ *cannot be part of* M^N*.*

Proof. Consider the base case $n = N$. By the definition of collision, $\left(r'_p, r'_o\right) \in X^n$, and by the definition of the algorithm, $\left(r'_p, r'_o\right) \notin M^N$.

Now, we prove the induction step: for $n < N$, if $(r_p, r_o) \in X^n$ and $\left(r'_p, r'_o\right) \in \kappa\left(r_p, r_o, X^n\right)$, then $(r_p, r_o) \in X^{n+1}$ and $\left(r'_p, r'_o\right) \in \kappa\left(r_p, r_o, X^{n+1}\right)$. By the definition of collision, both $\left(r'_p, r'_o\right) \in X^n$ and $(r_p, r_o) \in \kappa\left(r'_p, r'_o, X^n\right)$ must hold, and hence by the definition of the algorithm, $\left\{(r_p, r_o), \left(r'_p, r'_o\right)\right\} \cap M^n = \emptyset$. Now, clearly $\left\{(r_p, r_o), \left(r'_p, r'_o\right)\right\} \subseteq C^{n+1}$ since $C^n \subseteq C^{n+1}$, so to show that $\left\{(r_p, r_o), \left(r'_p, r'_o\right)\right\} \subseteq X^{n+1}$, we just need to establish that $\left\{(r_p, r_o), \left(r'_p, r'_o\right)\right\} \subseteq U^{n+1}\left(M^n\right)$.

By way of contradiction, assume otherwise. Then, there must exist some $\left(r''_p, r''_o\right) \in M^n \cap C^n$ such that either $(r_p, r_o) \in \kappa\left(r''_p, r''_o, U\right)$ or $\left(r'_p, r'_o\right) \in \kappa\left(r''_p, r''_o, U\right)$. Without loss of generality, let $(r_p, r_o) \in \kappa\left(r''_p, r''_o, U\right)$. Since $(r_p, r_o) \in X^n$, we know that $\left(r''_p, r''_o\right) \notin M^{n-1}$. Then, for $\left(r''_p, r''_o\right) \in M^n$ to hold, it must be that $\left(r''_p, r''_o\right) \in X^n$. But then, $(r_p, r_o) \in \kappa\left(r''_p, r''_o, X^n\right)$,

which contradicts $\left(r_p'', r_o''\right) \in M^n$. Hence, we have shown that $\left\{\left(r_p, r_o\right), \left(r_p', r_o'\right)\right\} \subseteq X^{n+1}$. Clearly then, $\left(r_p', r_o'\right) \in \kappa\left(r_p, r_o, X^{n+1}\right)$. Thus we have proved the result via induction. □

Next, we show that in Algorithm 1, any match is either added to the merge in the step equal to its rank, or it is never added at all.

Lemma B.2. *In Algorithm 1, if (r_p, r_o) isn't in $M^{\rho(r_p, r_o)}$, then it is not in M^N.*

Proof. The statement is trivially true if $\rho\left(r_p, r_o\right) = N$, so we consider $\rho\left(r_p, r_o\right) = n < N$. By the definition of the algorithm, Step n is the first step at which (r_p, r_o) could join the merge.

Now, if $(r_p, r_o) \notin M^n$ because $(r_p, r_o) \notin X^n$, then for any $n' > n$, $(r_p, r_o) \notin M^n$. To see this, note that since $(r_p, r_o) \in C^n$, if $(r_p, r_o) \notin X^n$, then it must be that $(r_p, r_o) \notin U^n\left(M^{n-1}\right)$. By definition, we know that $U^{n'}\left(X'\right) \subseteq U^n\left(X\right)$, for any $n' > n$ and $X \subseteq X'$. Hence, $U^{n'}\left(M^{n'-1}\right) \subseteq U^n\left(M^{n-1}\right)$ for any $n' > n$ and hence $(r_p, r_o) \notin X^{n'}$. From this, it follows that for any $n' > n$, $(r_p, r_o) \notin M^{n'}$and hence $(r_p, r_o) \notin M^N$.

Now, if (r_p, r_o) fails to join the merge at Step n and $(r_p, r_o) \in X^n$, then it must be that there exists some $\left(r_p', r_o'\right) \in \kappa\left(r_p, r_o, X^N\right)$. By Lemma B.1, we conclude that $\left\{\left(r_p, r_o\right), \left(r_p, r_o\right)\right\} \cap M^N = \emptyset$. □

Then, we establish the existence of a set that is stable relative to ranked criterion C by showing that Algorithm 1 produces it.

Lemma B.3. *Algorithm 1 outputs a set that is stable relative to ranked criterion C.*

Proof. To prove that the algorithm's output, M, is stable relative to C, we must consider both the forward and backward implication in the definition. First, consider the forward implication, which requires that for any $(r_p, r_o) \in M$ both $\kappa\left(r_p, r_o, C^n \cap U^n\left(M\right)\right) = \emptyset$ and $(r_p, r_o) \in U^n\left(M\right)$, where $n = \rho\left(r_p, r_o\right)$. We will prove both implications by contradiction.

By way of contradiction, assume that there exists $\left(r_p', r_o'\right) \in \kappa\left(r_p, r_o, C^n \cap U^n\left(M\right)\right)$. By Lemma B.2, (r_p, r_o) must join the merge in Step n. For this to be true, it must be that $(r_p, r_o) \in X^n$. But since $\left(r_p', r_o'\right) \in \kappa\left(r_p, r_o, C^n \cap U^n\left(M\right)\right)$ and $U^n\left(M\right) \subseteq U^n\left(M^{n-1}\right)$, it must also be that $\left(r_p', r_o'\right) \in \kappa\left(r_p, r_o, X^n\right)$, which by Lemma B.1 means that $(r_p, r_o) \notin M$, providing the required contradiction.

Now, by way of contradiction, assume that $(r_p, r_o) \notin U^n\left(M\right)$. Since Lemma B.2 tells us that $M^{n-1} = M \cap C^{n-1}$, we can conclude that $U^n\left(M\right) = U^n\left(M^{n-1}\right)$. This then tells us that $(r_p, r_o) \notin X^n$, and hence that $(r_p, r_o) \notin M^n$, providing the required contradiction.

Now, consider the backward implication. By way of contradiction, assume that $\kappa\left(r_p, r_o, C^n \cap U^n\left(M\right)\right) = \emptyset$ and $(r_p, r_o) \in U^n\left(M\right)$, but $(r_p, r_o) \notin M$, where $n = \rho\left(r_p, r_o\right)$. It must be that $(r_p, r_o) \in X^n$, since $U^n\left(M\right) \subseteq U^n\left(M^{n-1}\right)$. Hence, $(r_p, r_o) \notin M$ requires that there exists $\left(r_p', r_o'\right) \in \kappa\left(r_p, r_o, C^n \cap U^n\left(M^{n-1}\right)\right)$. But, Lemma B.2 tells us that $M^{n-1} = M \cap C^{n-1}$, so $U^n\left(M^{n-1}\right) = U^n\left(M\right)$. Hence, $\left(r_p', r_o'\right) \in \kappa\left(r_p, r_o, C^n \cap U^n\left(M\right)\right)$, which provides the required contradiction. □

We also need to show that any set that is stable with respect to ranked criterion C is a merge.

Lemma B.4. *Any set $M \subseteq U$ that is stable with respect to ranked criterion C must be a merge.*

Proof. By way of contradiction, assume that M is stable with respect to ranked criterion C, but is not a merge. Then, there exists $(r_p, r_o) \in M$ such that $\kappa(r_p, r_o, M) \neq \emptyset$. Let $(r'_p, r'_o) \in \kappa(r_p, r_o, M)$. Now, if $\rho(r_p, r_o) \neq \rho(r'_p, r'_o)$, then without loss of generality, let $\rho(r_p, r_o) > \rho(r'_p, r'_o)$. Then, it must be that $(r_p, r_o) \notin U^{\rho(r_p, r_o)}(M)$, which contradicts the definition of stable with respect to ranked criterion C.

So, assume that $\rho(r_p, r_o) = \rho(r'_p, r'_o)$. Further, assume that $\{(r_p, r_o), (r'_p, r'_o)\} \subseteq U^{\rho(r_p, r_o)}(M)$; otherwise we have already contradicted the definition of stable. Then, $\{(r_p, r_o), (r'_p, r'_o)\} \subseteq C^{\rho(r_p, r_o)} \cap U^{\rho(r_p, r_o)}(M)$, and hence $\kappa\left(r_p, r_o, C^{\rho(r_p, r_o)} \cap U^{\rho(r_p, r_o)}(M)\right) \neq \emptyset$, contradicting the definition of stable with respect to ranked criterion C. □

Finally, we are ready to prove that the stable match relative to C produced by Algorithm 1 is in fact the unique stable merge relative to C.

Proof of Proposition B.1. Lemma B.3 shows that Algorithm 1 outputs a stable match. Uniqueness can be established by induction on rank. For the base case, consider some $(r_p, r_o) \in M \cap C^1$. Since M is stable, it must be that $\kappa(r_p, r_o, C^1) = \emptyset$. But this would require that (r_p, r_o) be included in M' as well. So, M and M' must agree on their intersection with C^1.

Now, for the inductive step, assume that $M \cap C^n = M' \cap C^n$. Take a match $(r_p, r_o) \in (M \setminus C^n) \cap C^{n+1}$. Since M is stable, it must be that $\kappa(r_p, r_o, C^{n+1} \cap U^{n+1}(M \cap C^n)) = \emptyset$. But since $M \cap C^n = M' \cap C^n$, it is also true that $\kappa(r_p, r_o, C^{n+1} \cap U^{n+1}(M' \cap C^n)) = \emptyset$, which means that $(r_p, r_o) \in M'$. So, by induction, any two merges must contain the same matches, that is, the stable match is unique. □

B.2 Registration data

In this subsection, we describe how the list of registrations to the NKR was assembled. The NKR provided us with snapshot files of the patient and donor pool between April 1, 2012 and December 4, 2014. These files are typically daily snapshots except for some missing periods, each of which is up to a month in length. Each snapshot corresponds to a different date and includes basic medical records for each patient and donor in the pool, their listing dates, the related patient for each donor (if any), and whether a patient is unpaired. From these snapshots we recover patient and donors departures, which may be due to being transplanted or other undocumented reasons. A small number of patients and donors depart without a transplant during the period of a missing snapshot; for these, we use bounds on the departure time using the two closest available snapshots, before and after the real departure date. These snapshots also include each patient's set of donors that may not be matched (i.e. are blocked) despite being virtually compatible. Some of these blocked donors are due to patient preference (not to match with these donors) and others are due to match failures.

C Simulation Details

We now provide details on the procedure used in Section 5.1.

C.1 Matching Offers

C.1.1 The matching algorithm and examples

We describe solve the linear programming problem subject to cycle length and chain constraints. Because it is computationally burdensome to compute all cycles in a pool with many patients and donors, we first solve the relaxed problem by ignoring the constraint that cycles cannot involve more than three transplants. Specifically, we solve:

$$\begin{array}{lll} \max\limits_{x_{jk}\in\{0,1\}} & \sum\limits_{j\in\mathcal{P}\cup\mathcal{U}}\sum\limits_{k\in\mathcal{P}\cup\mathcal{A}} c_{jk}\, w_{jk}\, x_{jk} & \\ \text{s.t.} & x_{jk} - \sum\limits_{\ell} x_{k\ell} = 0 & \text{for all } k\in\mathcal{P} \\ & \left.\begin{array}{l} \sum\limits_{j} x_{jk} \le 1 \\ \sum\limits_{k} x_{jk} \le 1 \end{array}\right\} & \text{for all } k \end{array}$$

where $\mathcal{A}$ is the set of altruistic donors, $\mathcal{P}$ is the set of pairs, $\mathcal{U}$ is the set of unpaired recipients, $x_{jk} = 1$ denotes a proposed transplant from the donor in $k \in \mathcal{P}\cup\mathcal{A}$ to the patient in $j \in \mathcal{P}\cup\mathcal{U}$, w_{jk} is the weight described in Section C.1.2 below, and $c_{jk} = 1$ if a transplant from k to j is allowed and 0 otherwise. The first constraint ensures that a donor who is part of a pair is only asked to donate an organ if the intended recipient has been proposed a transplant. The second and third constraints ensure that no donor or recipient is involved in more than one transplant.

If the solution to the problem does not involve any long cycles, i.e. there do not exist $j_1, \ldots, j_n \in \mathcal{P}$ such that $x^*_{j_{k+1}j_k} = 1$ for $k \in \{1, \ldots, K-1\}$, $x^*_{j_1 j_K} = 1$, and $K \ge 4$, then it must be that x^* is optimal, given the no long-cycle constraints, and is our desired solution. In 87.2% of simulation days the solution to this relaxed problem yields a feasible match without any further cycle restrictions.

If the solution to this problem contains at least one long cycle, then we proceed as follows. We begin by following the algorithm in Anderson et al. (2015). The algorithm includes a constraint that explicitly prohibits all long cycles in x^*, i.e. for each sequence $j_1, \ldots, j_K \in \mathcal{P}$ such that for $k \in \{1, \ldots, K-1\}$, $x^*_{j_{k+1}j_k} = 1$, $x^*_{j_1 j_K} = 1$, and $K \ge 4$, we include a constraint in the problem above to ensure that $x^*_{j_1 j_K} \prod_{k=1}^{K-1} x^*_{j_{k+1}j_k} = 0$. If the solution to the modified problem also contains long cycles, we modify the problem again to prohibit those cycles. We iterate this procedure up to ten times. This procedure yields a feasible solution in about 50% of the remaining cases (about 7.8% of all cases) with an average of approximately 6.3 iterations.

If the algorithm above does not yield a feasible solution even after 10 repetitions, we proceed to the next phase in which we use Johnson (1977)'s algorithm to compute cycles and explicitly add constraints that prohibit long cycles. This algorithm searches the compatibility graph induced by c to calculate cycles. We enumerate and add a constraint to our program to prohibit any long cycles we have found. The number of cycles is usually small, but sometimes is very large. Therefore, we search for cycles with a time-out of one second. We find a solution to the problem with these additional constraints and terminate our algorithm if the solution is feasible. This procedure is repeated once more, if necessary. At the end of this phase, we are able to find an optimal solution to the full problem in about 99.0% of the simulation days.

For the remaining 1.0% of days in the simulation, our matching algorithm still ends up with long cycles. Whenever this is the case, we attempt to find a solution with the following, alternative algorithm:

$$
\begin{aligned}
\max_{y_\ell \in \{0,1\}} \quad & \sum_{\ell \in C} w_\ell \, y_\ell & & \\
\text{s.t.} \quad & w_\ell = \sum_{i \in \ell}^{|\ell|-1} w_{i,i+1} + w_{|\ell|,1} & & \text{if } \ell \text{ is a cycle,} \\
& w_\ell = \sum_{i \in \ell}^{|\ell|} w_{i,i+1} & & \text{if } \ell \text{ is a chain,} \\
& \ell \cap \ell' = \emptyset & & \text{if } y_\ell \, y_{\ell'} = 1, \\
& c_{i,i+1} = 1,\ c_{|\ell|,1} = 1, \forall i \in \ell & & \text{if } \ell \text{ is a cycle,} \\
& c_{i,i+1} = 1, \forall i \in \ell & & \text{if } \ell \text{ is a chain.}
\end{aligned}
$$

where C is the set of feasible chains and cycles, $y_\ell = 1$ denotes implementing chain or cycle ℓ, w_ℓ is the sum of the weights for each transplant in the chain or cycle indexed by ℓ, and $c_{jk} = 1$ if a transplant from k to j is allowed and 0 otherwise. Denote the number of pairs/altruistic donors/unpaired recipients in a chain or cycle ℓ by $|\ell|$. The first set of constraints defines the total weight w_ℓ for each chain or cycle; the second set ensures no donor or recipient is involved in more than one transplant; the third set ensures that all simultaneously proposed chains and cycles do not overlap. Because it is computationally burdensome to compute all chains in a pool with many patients and donors, we solve the problem chains with up to length of 5. We abandon the match on that "simulation day" if we still cannot find a solution, and all patients and donors are returned to the kidney exchange pool to wait for the next day for the transplant offer. Even in these rare cases, we can find an optimal match within 2.9 days on average.

Figure C1 illustrates a few kidney exchange pools. The left panel shows compatibility as captured by c, feasible transplants, and the optimal match. Blue dots denote patient-donor pairs, and magenta dots denote altruistic donors. We ignore unpaired patients in this illustration for simplicity. A blue arrow depicts a feasible transplant with the origin of the arrow denoting the donor. Red and green arrows depict cycles and chains, respectively, in the optimal match. Given feasible transplants on left, our match algorithm offers the one on the right. The figure shows there may be several feasible transplants, and in these cases, the optimal match may be relatively easy to determine. Figure C2, on the other hand, illustrates

a relatively hard-to-match problem where the optimal match is relatively more difficult to determine.

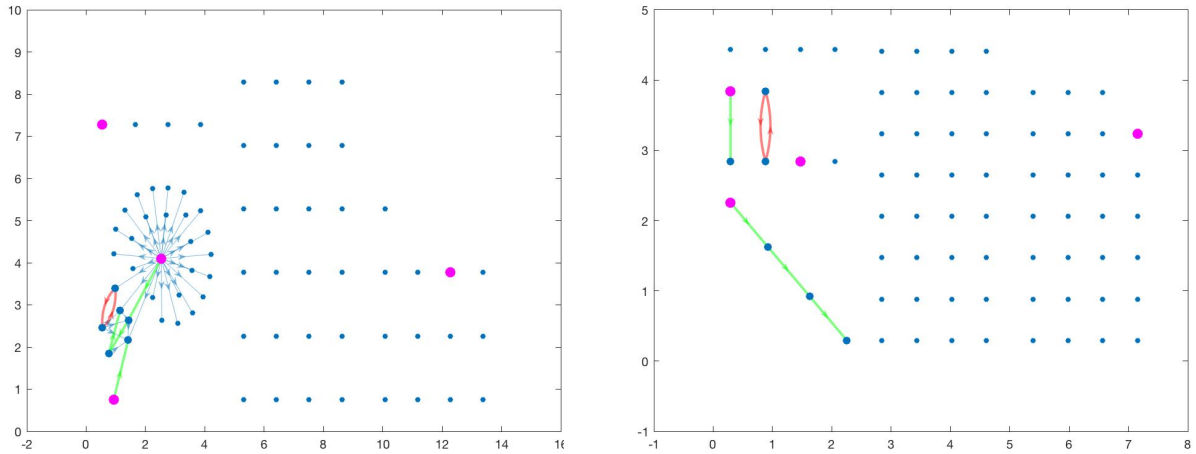


Figure C1: Optimal matches for two simple pools

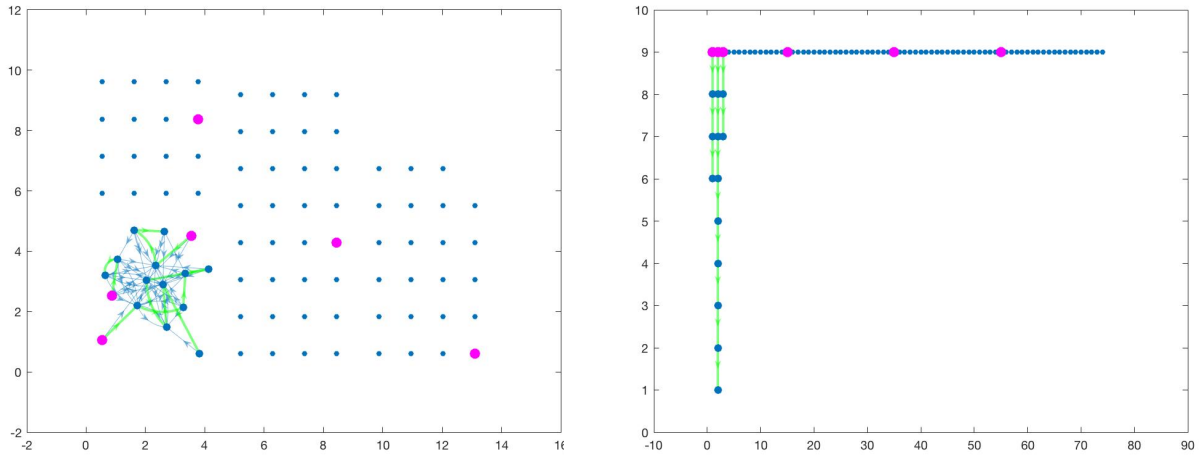


Figure C2: Optimal matches for two more complicated pools

C.1.2 Weights

We attempt to closely match the weights, w_{ij}, on a NKR transplant between patient i and donor j. These weights are designed to favor patients who are highly sensitized, in other words, who are harder to transplant. To define weights, w, NKR first defines a matching power for each submission. Each patient has a Patient Match Power (PMP), a number between 0 and 1, that is a fraction of compatible donors in the NKR pool for that patient. A low PMP for patient i implies that few donors are compatible with patient i. Similarly, the Donor Match Power (DMP) is defined as the fraction of patients in the NKR pool with whom that donor is compatible. Because these quantities and the pool used by the NKR

$WNKR_{ij}$ interval	w_{ij}
$WNKR_{ij} > 70$	1
$25 < WNKR_{ij} \leq 70$	1.01
$5 < WNKR_{ij} \leq 25$	1.2
$WNKR_{ij} \leq 5$	1.5

Table C2: Weights used by the NKR

to compute these match powers are not directly observed in our dataset, we calculate them using our sample.

Given these characteristics, NKR calculates a scaled measure of how likely a feasible transplant can occur between i and j, $WNKR_{ij}$. Specifically,

$$WNKR_{ij} = PMP \times DMP \times 10,200.$$

A low $WNKR_{ij}$ correlates with a transplant between i and j being unlikely. It is important to note that the magnitude of $WNKR_{ij}$ is not related to the success of a transplant if it turns out to be feasible. These weights therefore accord higher priority to hard-to-match patients and donors. Using $WNKR_{ij}$, NKR assigns the weights w_{ij} as follows:

Because these weights are less than 2, they typically maximize the total number of transplants. However, these weights may sometimes result in two transplants, each with weight 1.5, instead of three transplants with weight 1 each.

C.2 Arrival and Departure

C.2.1 Arrival process

We assume the daily number of submissions in the NKR is given by a Poisson distribution with parameter λ, where λ represents the mean daily arrival rate for NKR. We estimate that parameter to be $\lambda = 1.975$. In each period, our simulations draw a number, say n_t, from this distribution. Then we draw n_t submissions with replacement from the entire pool that ever registered in the NKR during the April 2012 to June 2014 sample period.

Figure C3 shows the fit of the arrival per day distributions of NKR and Poisson. Notice that NKR's distribution has more 0 arrivals per days than the poisson distribution. This mass point is explained by weekends, which appear to have a much lower arrival rate.[3] Figure C4 shows the arrival per day distribution of NKR for weekdays and our estimated poisson distribution, which shows a better fit.

[3] Only 40 arrivals in weekends over the course of 140 weeks.

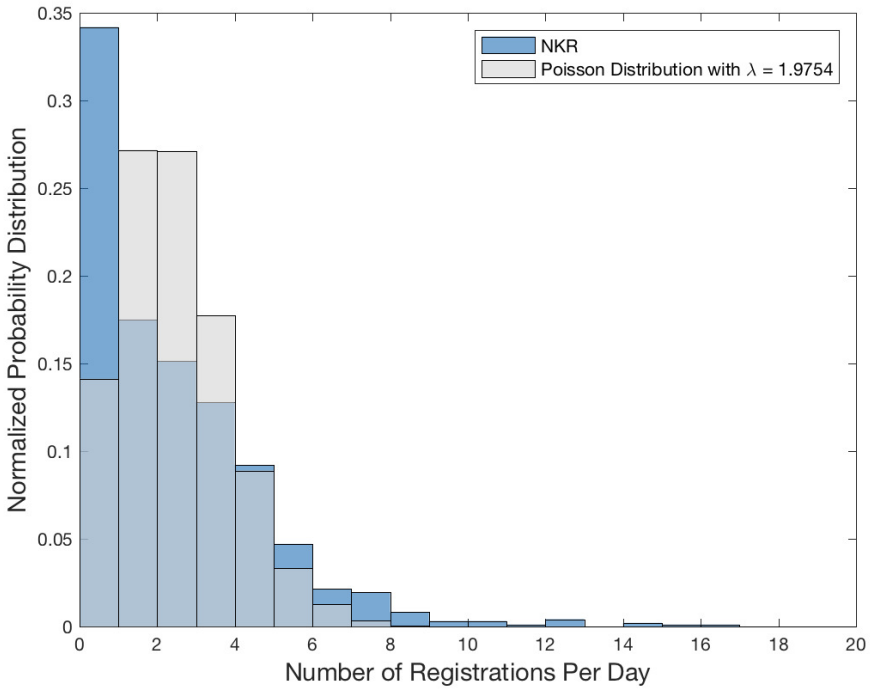


Figure C3: Distribution of NKR and Poisson Number of Submissions Per Day

C.2.2 Departure process

To model departures, we estimate an interval censored hazard model to calculate the rate at which patients and/or donors depart the NKR without a transplant. Specifically, let t_i^a, t_i^*, and t_i^d be the (latent) arrival, transplant, and departure dates for an unpaired patient, donor, or patient-donor pair i. Our dataset records t_{0i} and t_i^* if i was transplanted. Further, if i is transplanted, then we know that $t_i^d > t_i^*$. If i is not transplanted, then in most cases we observe t_i^a, but in some cases, we only know that t_i^a belongs to an interval $\left[t_i^{a-}, t_i^{a+}\right]$ (typically within a week). If i departed without a transplant, we observe t_i^deither exactly or up to a small interval. If i remains in the NKR at the end of our sample, then we know that $t_i^d > T$. Using these observations, we can construct bounds on the duration τ_i that each unit i remains in the NKR without a transplant.

With these observations, we estimate the exponential hazards model. The model is characterized by a survival function

$$S_i(\tau) = \exp(-\lambda_i t),$$

where we use the parametric form

$$\lambda_i = \alpha + z_i \beta.$$

The likelihood of the model for the interval censored survival data is straightforward to derive, and estimation via `intcens` in STATA is straightforward.

Table C3: Departure Hazard Rate Estimates

	(1) Patient-Donor Pairs	(2) Unpaired Patients	(3) Altruistic Donors
Patient Matching Power	1.824***	16.35***	
	(0.244)	(2.446)	
Donor Matching Power	0.0699		0.000167
	(0.137)		(0.00126)
Patient Age	0.994*	1.002	
	(0.00349)	(0.00383)	
Donor Age	1.008*		1.011
	(0.00442)		(0.0140)
AB Blood-type Patient	2.465***	1.557*	
	(0.698)	(0.390)	
A Blood-type Patient	1.184	1.294	
	(0.160)	(0.265)	
B Blood-type Patient	1.077	0.635*	
	(0.172)	(0.158)	
AB Blood-type Donor	0.584**		1.249
	(0.150)		(1.357)
A Blood-type Donor	0.667***		1.562
	(0.0832)		(0.654)
B Blood-type Donor	0.608***		0.764
	(0.0957)		(0.520)
Constant	0.00578***	0.000892***	0.0656
	(0.00565)	(0.000203)	(0.234)
Observations	1,264	498	164

Note: Interval censored exponential hazard model. Patient (Donor) Match Power is the fraction of donors (patient) in the NKR pool over the course of a sample a given patient (donor) is compatible with. Sample restricted to patients and donors that registered after April 2012.

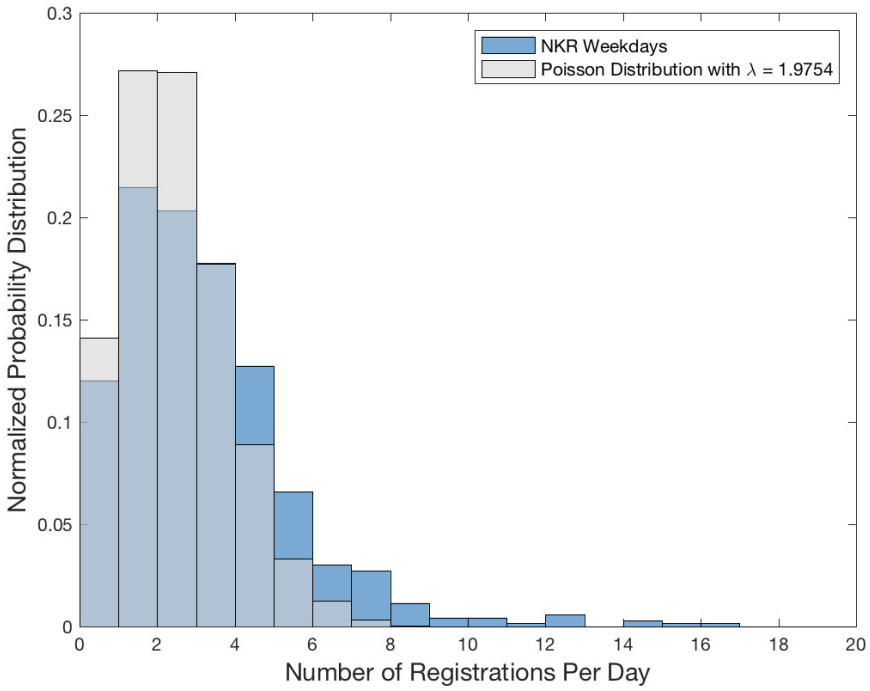


Figure C4: Distribution of NKR and Poisson Number of Submissions Per Day for Weekdays

Table C3 presents the estimates. Note that the hypothesis tests in a hazard model are reported relative to 1, which implies no effect. As can be seen, patients with blood types that are easier to match or who are paired with easier-to-match donors have a higher departure rate. This finding is consistent with patients and patient-donor pairs departing in response to transplantation opportunities elsewhere, either through direct donation, deceased donor transplants, or live-donor exchanges outside the NKR.

C.3 Compatibility and acceptance

To calculate whether donor j is compatible with patient i, we use the blood types of the patient and donor, the tissue type of the donor, and the list of unacceptable antigens listed by the patient. There are three additional ways in which a transplant between a patient and a donor can be prohibited.

First, upon registration, each patient can declare criteria for excluding donors based on a variety of characteristics. These include thresholds for the maximum donor age and minimum donor weight that are acceptable. These criteria are recorded in our dataset.

Second, upon arrival a patient can list as unacceptable any number of specific donors who were in the NKR pool at the time. This rejection can be done for any reason, including known pathologies. Patients can also exclude donors later, but according to our understanding, the

practice is most common during registration. Our dataset includes the anonymized identifiers for these excluded donors.

Third, when a transplant is proposed, a patient may refuse the specific donor. A first phase of refusals is at the patient's discretion (with advise from his/her surgeon). If a patient chooses to proceed after the first phase, a final tissue-type compatibility test is conducted. We refer to this as the second phase.

In our simulations, we initialize $c_{ij} = 1$ if j is compatible with i and if j is not excluded by i. Otherwise, we set $c_{ij} = 0$. If j is offered to i during the simulation and a transplant is ruled out during either the first or second phases in the third type of exclusion, then we set $c_{ij} = 0$ for future "simulation days."

C.4 Burn-in and calculating standard errors

We may start our simulations from any initial state for the NKR because the effect of the chosen initial state fades over time. A convenient choice is to pick an initial pool with no unpaired patients, altruistic donors, or patient-donor pairs. Although the initial pool does not affect long-run averages with enough simulations, it is advisable to discard or burn-in a portion of the initial chain in order to improve the estimates' precision. A burn-in of about 2000 days appears to yield potential scale reduction factors for the number of transplants per day that is close to 1, suggesting that the chain is likely to have converged at that point.

Our simulations produce a series $y_1, \ldots, y_T$ of the transplants that occur on each day after an initial burn period. We estimate $f(\boldsymbol{q})$ as the sample mean of the y_t and calculate the standard errors of this estimate using the non-overlapping batch means estimator by following Chapter 12 in Robert and Casella (2004). The method divides the time series of y_t into batches, calculates the sample mean in each of those batches, and uses the variability in sample means to estimate the standard error of $f(\boldsymbol{q})$. We use the commonly recommended batch size of approximately $\sqrt{T}$. The procedure is a simple and popular method that accounts for autocorrelation of the y_t.

C.5 Calibration

C.5.1 Calibration Procedure

Our simulation procedure is tailored to match the procedures and practices used by the NKR. In most cases, the data or institutional knowledge directly tell us the parameters; e.g., the weights w_{ij} are chosen to match NKR's practices. However, there are a few aspects of the real-world procedures and outcomes in the NKR on which we don't directly have information. We model and parametrize these aspects in our simulation model and calibrate them to match the realized number of matches in the NKR.

There are two main sets of parameters we need to calibrate. First, we do not have direct data on the frictions of translating proposed transplants into surgeries. As mentioned in Section

C.3, the various acceptance phases may result in some transplants not being consummated. Each phase introduces a time-lag between transplants being proposed and finalized as well as the chance of a match being aborted. Roughly speaking, these frictions reduce the number of transplants facilitated by the NKR.

We parametrize these phases and calibrate the parameters to best fit the observed number of transplants by patient type. To do so, we simulate outcomes predicted by our model for various lengths (number of days) and various failure probabilities for both phases. The first phase parameters can be interpreted as controlling the frictions in the system because proposed matches are refused, whereas the second phase parameters govern the frictions due to biological compatibility tests.

Second, as mentioned earlier, when chains are aborted because of a refusal, NKR usually tries to use the donor, called the bridge donor, of the last transplanted patient for a new chain. However, the exchange prefers not to wait too long to start a chain with this donor. If a new chain cannot be found, the donor is offered to a patient without a related donor. Unfortunately, we do not know of a consistent policy rule followed by the NKR. We therefore also experimented with the number of days the NKR tries to match a bridge donor.

In summary, we calibrated five parameters: (i) the number of days a bridge donor can initiate a new chain, (ii) the number of days for consent and the probability of consent for the two phases of match acceptance, and (iii) the number of days taken for testing and the probability of failing the tissue type biological test.

This approach parsimoniously parametrizes the nature of frictions in the NKR. However, these models' single parameter versions are clearly a simplification. Most likely, NKR policies and their ability to translate proposed matches into transplants evolves over time and includes some ad hoc modifications to their basic procedure.

For our calibrations, we conduct our simulations by setting an initial market with the patients and donors who were present in the NKR on 1 April, 2012. This is the date from which we have clear registration data. Then, for each parameter set, we run 100 simulations until December 2014, the last date of the available data.

We calibrated our parameters to match the transplantation probabilities, days in the exchange broken by whether or not the patient/donor was transplanted, the total stock of patients and donors in the exchange, and the total number of transplantations. The best fitted parameters were 14 days of waiting and 80% success rate for each of the two phases.

C.5.2 Calibration Results

Figures *C*5 - *C*8 compare the trends for these statistics across the data and the simulations for our chosen parameters. The dashed lines depict data, darker lines are the mean of 100 simulations, and dashed lines are 95% confidence interval from the simulations.

Green lines show the number of submissions of a given type (altruistic, pair, or unpaired), red lines show the cumulative number of submissions that departed the market without a transplant, and blue lines show cumulative number of submissions that were transplanted.

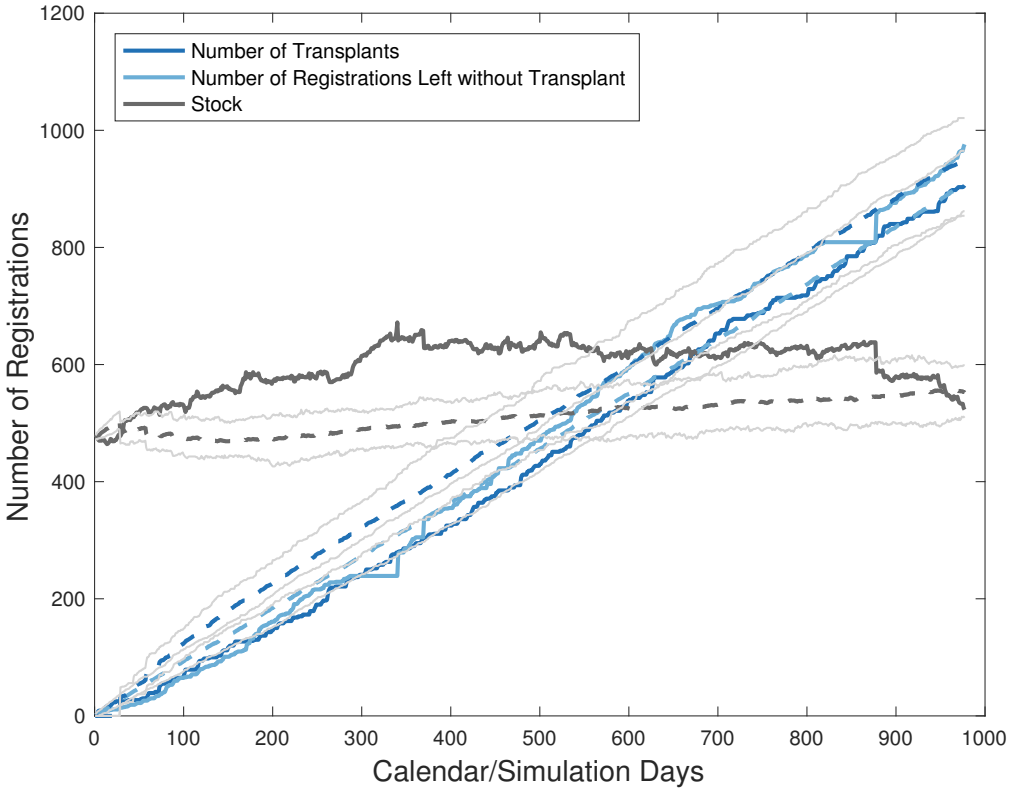


Figure C5: Calibration for All Submissions

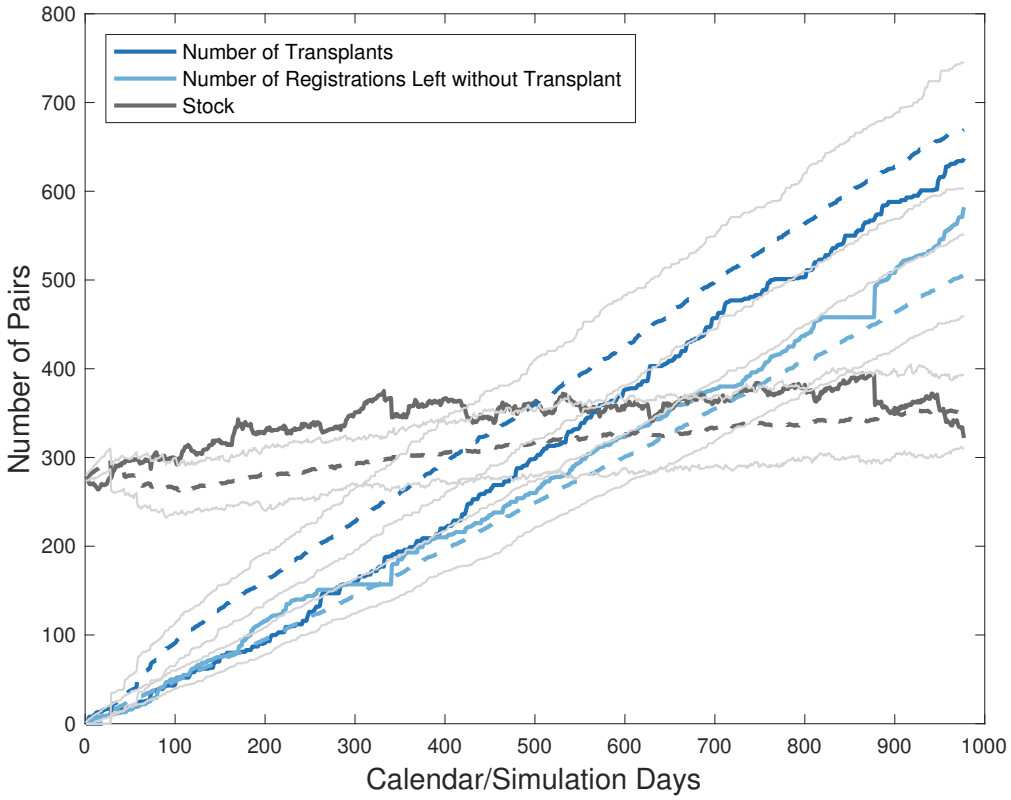


Figure C6: Calibration for Pairs

Note: Observed Quantities and Simulations are shown with solid and dashed lines respectively. Solid grey lines represent 95% confidence intervals.

Figures $C5$ and $C6$ show that, at the calibrated parameters, the model fits the data extremely well. Although the averages are not as well matched in Figures $C8$ and $C7$, the observed quantities are within the 95% confidence intervals for the model.

Table $C4$ presents the summary statistics comparison between simulations and NKR data. Although we calibrated only five parameters, the table shows that the model matches several data moments extremely well.

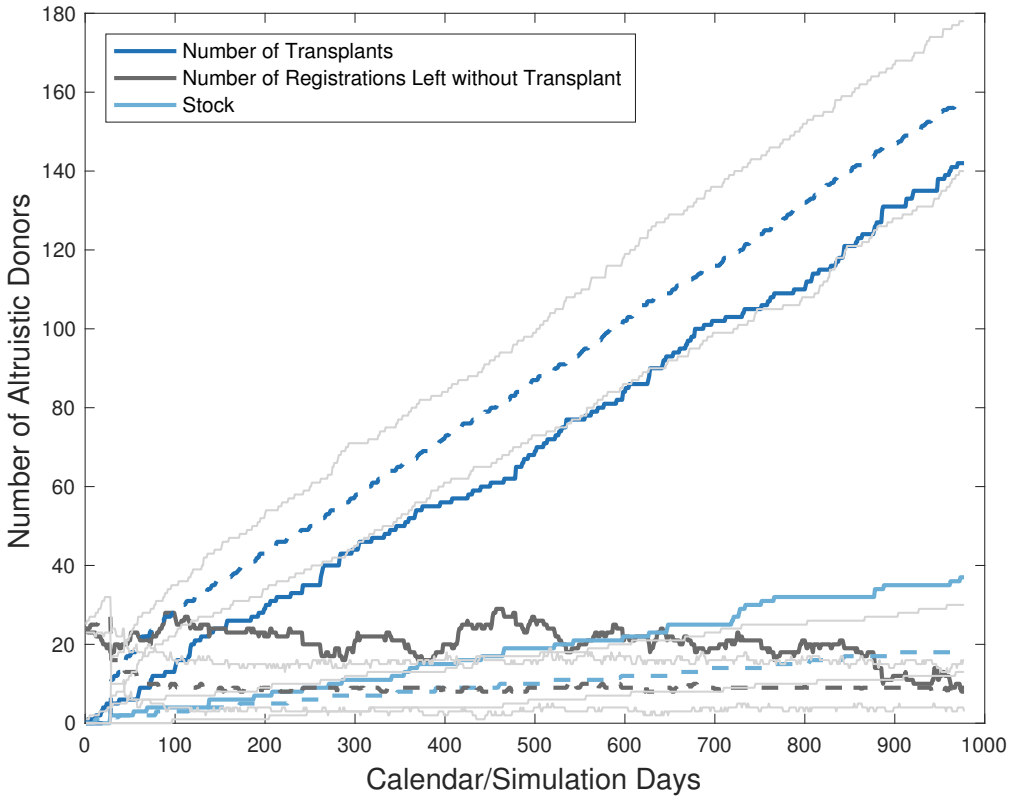


Figure C7: Calibration for Altruistic Donors

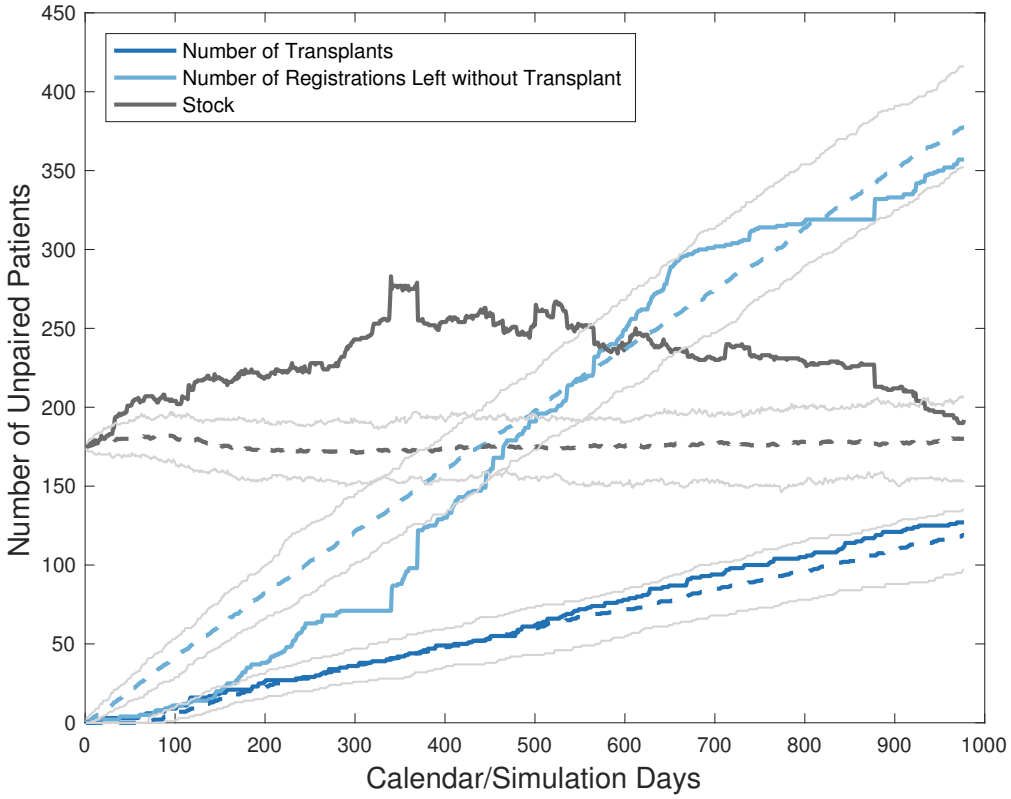


Figure C8: Calibration for Unpaired Patients

Note: Observed Quantities and Simulations are shown with solid and dashed lines respectively. Solid grey lines represent 95% confidence intervals.

Table C4: Model Fit

	Patient Match Probability	Donor Match Probability	Mean Days in the NKR for Transplanted Patients	Mean Days in the NKR for Untransplanted Patients	Mean Days in the NKR for Transplanted Donors	Mean Days in the NKR for Untransplanted Donors
Panel A: Observed Quantities from the National Kidney Registry (NKR)						
Pair	0.56	0.55	146.67	197.58	140.53	189.92
Chip Patients	0.30	-	63.85	135.40	-	-
Altruistic Donors	-	0.79	-	-	105.98	74.91
Under-Demanded	0.41	0.40	195.19	223.27	187.97	215.60
Over-Demanded	0.65	0.66	117.55	129.45	115.37	128.52
Self-Demanded	0.64	0.62	133.09	188.89	126.22	178.06
Panel B: Simulated Quantities						
Pair	0.60	0.57	99.85	183.35	120.00	179.42
Chip Patients	0.27	-	87.87	105.40	-	-
Altruistic Donors	-	0.89	-	-	50.00	47.72
Under-Demanded	0.38	0.36	164.01	211.06	190.64	209.50
Over-Demanded	0.71	0.69	69.62	135.61	78.82	132.20
Self-Demanded	0.73	0.69	84.87	153.87	106.51	148.88

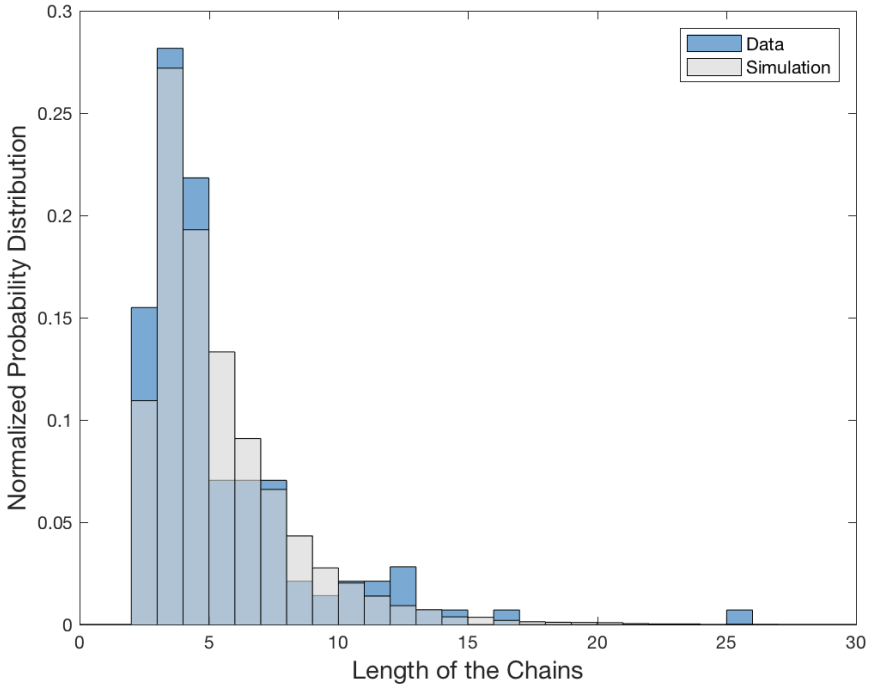


Figure C9: Chain Length Distribution

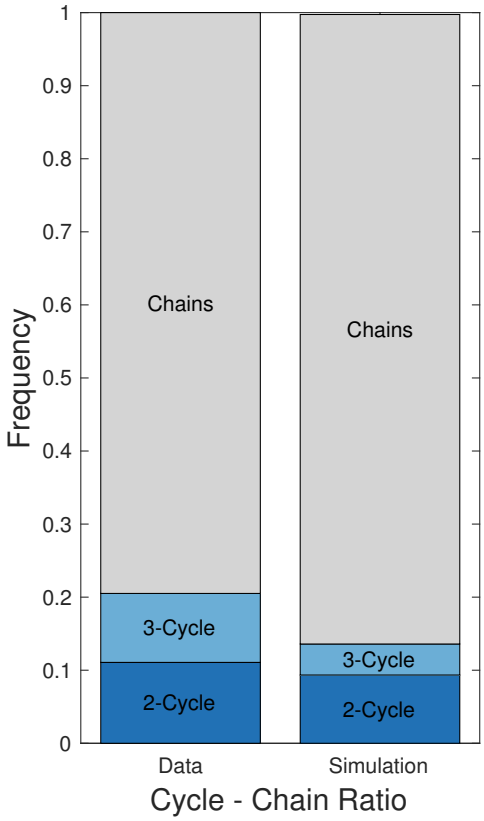


Figure C10: Cycle - Chain Ratio

Table C5: Regression Tree Summary Statistics

	N	Match Probability			Marginal Product		
		Mean	S.E.	Within Category Standard Deviation	Mean	S.E.	Within Category Standard Deviation
				Panel A: Altruistic Donors			
Non-O Donor	102	0.86	(0.01)	0.04	0.84	(0.05)	0.21
O Donor	62	0.94	(0.01)	0.00	1.86	(0.06)	0.17
				Panel B: Patient-Donor Pairs			
O Patient, Non-O Donor	493	0.28	(0.01)	0.16	0.08	(0.02)	0.19
O Patient, O Donor, PRA >= 91%	100	0.30	(0.01)	0.23	0.13	(0.05)	0.23
O Patient, O Donor, PRA < 91%	148	0.81	(0.01)	0.03	0.72	(0.04)	0.22
Non-O Patient, O Donor, PRA >= 96%	124	0.28	(0.01)	0.21	0.09	(0.05)	0.21
Non-O Patient, Non-O Donor, PRA < 96%	292	0.85	(0.01)	0.03	0.69	(0.03)	0.26
Non-O Patient, O Donor, PRA < 96%	108	0.84	(0.01)	0.01	1.44	(0.05)	0.30
				Panel C: Unpaired Patients			
Unpaired Patients	501	0.24	(0.01)	0.13	0.07	(0.02)	0.20

Notes: Categories are determined by regression tree analysis to predict marginal products as a function of whether a submission is a pair or altruist, blood types, and the patient's PRA. Our procedure followed standard recommendations in Friedman et al. (2001). Specifically, we used 10-fold cross-validation to pick the penalty parameter on the number of nodes, required each leaf to have at least 20 observations and pruned a leaf if it does not increase the overall fit by at least 2%. The resulting tree is depicted in Figure 8. Standard errors for the simulations are calculated by following Chapter 12 of Robert and Casella (2004). The within category standard deviation is estimated using shrinkage methods recommended in Morris (1983).

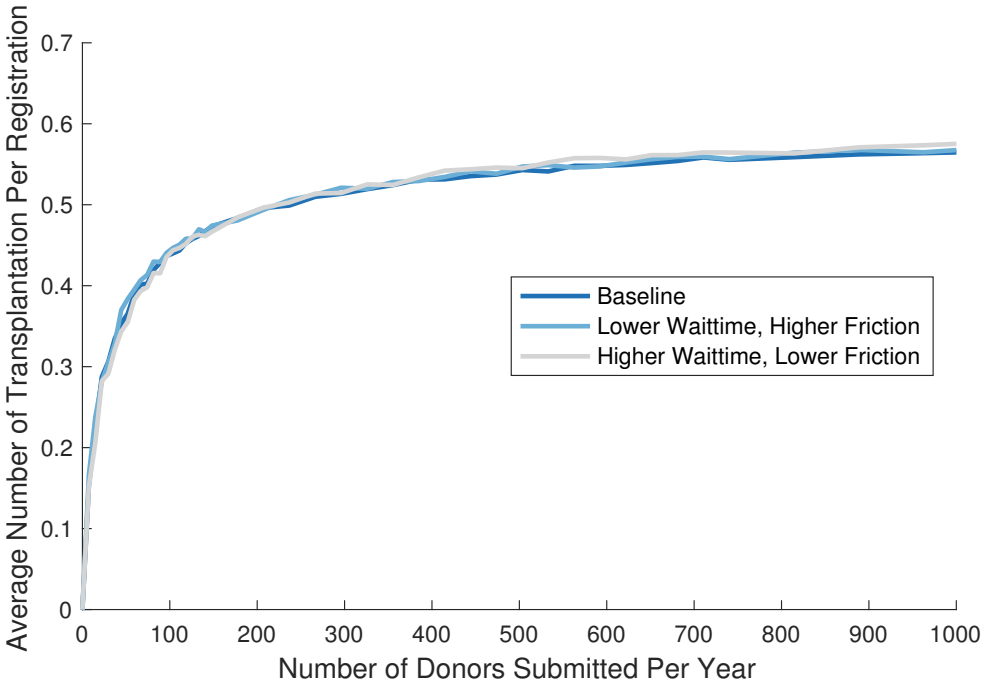


Figure D11: Robustness: Production Function versus Scale

Notes: Constructed as in figure 6.

D Robustness analyses

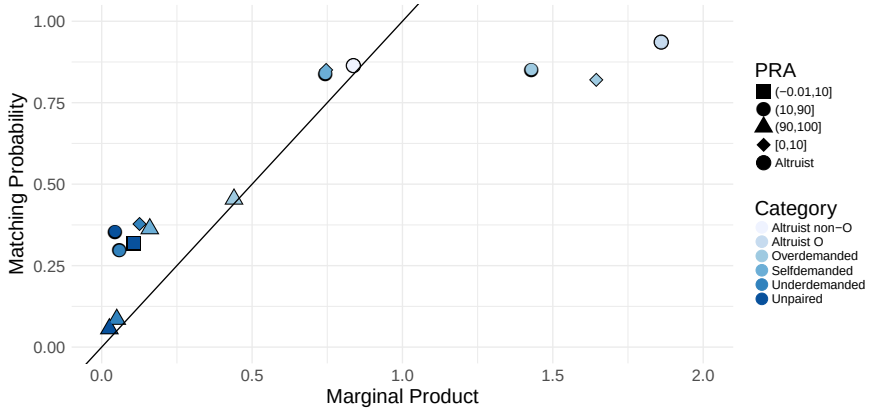
This section assesses the robustness of our results to calibrated parameters. Specifically, frictions in consummating proposed transplants due to longer waiting times but higher approval rates produce similar moments as lower waiting times and lower acceptance rates. Chain lengths, however, are increasing in acceptance rates and are best matched by our baseline parameters. We compare our baseline results with two substantially different parameters. The first, labelled "Higher Wait-time and Lower Frictions," has two weeks and three weeks for each of the two approval periods (approval and biological testing) but increases the acceptance rates in each phase from 80% to 85%. The other, labelled "Lower Wait-time and Higher Friction," uses three days and three weeks for each phase, respectively, but decreases the acceptance rates in each phase from 80% to 75%.

The qualitative and quantitative findings are robust to these alternative parameters. Figure D11 plots average products, as in Figure 6. These alternative parameters yield average product functions that closely follow the baseline. Table D6 shows the inefficiency estimates as in Table 3. The estimated inefficiency is within 5-10% of the baseline. Figure D12 shows marginal product versus matching probability of registrations aggregated by category, as in 7b. These results are also qualitatively similar. Table D7 shows marginal product, matching probability, and point system summary statistics, as in Table C5. Again, the points system under the alternative parameters are similar in magnitude.

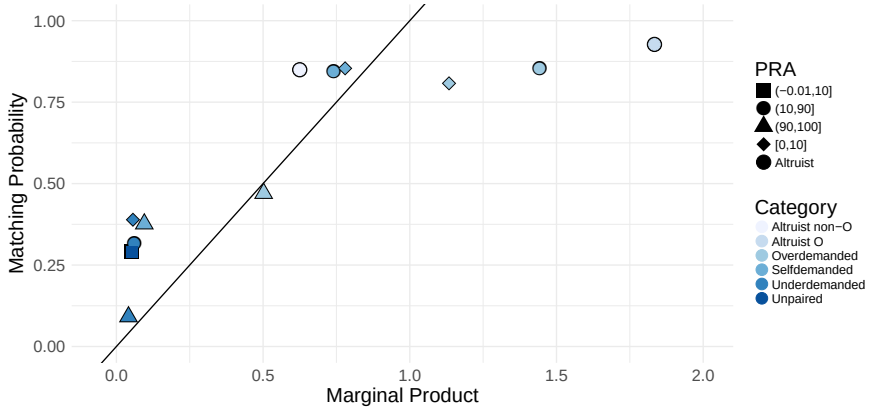
Table D6: Robustness: Total Efficiency Loss

	Number of Hospitals	Efficiency Loss: Additional Kidney Exchange Transplants		
		Base	Higher Waittime Lower Friction	Lower Waittime Higher Friction
		Panel A: All Hospitals		
All Hospitals	164	447.7	454.2	357.1
		Panel B: By hospital size (number of PKEs per year)		
Top Quartile	42	237.5	229.7	238.0
2nd Quartile	48	132.7	126.8	77.4
3rd Quartile	40	57.9	72.1	22.0
Bottom Quartile	34	19.7	25.7	19.7
		Panel C: By Platform Membership		
NKR	68	234.8	231.2	215.6
Only UNOS and APD	45	106.0	107.2	73.4
None	51	106.9	115.8	68.0
		Panel D: By NKR Participation Rate (Fraction of PKEs facilitated through the NKR)		
Top Quartile	17	14.5	15.8	9.0
2nd Quartile	17	44.2	44.2	28.7
3rd Quartile	17	81.5	80.9	78.3
Bottom Quartile	17	94.7	90.3	99.6

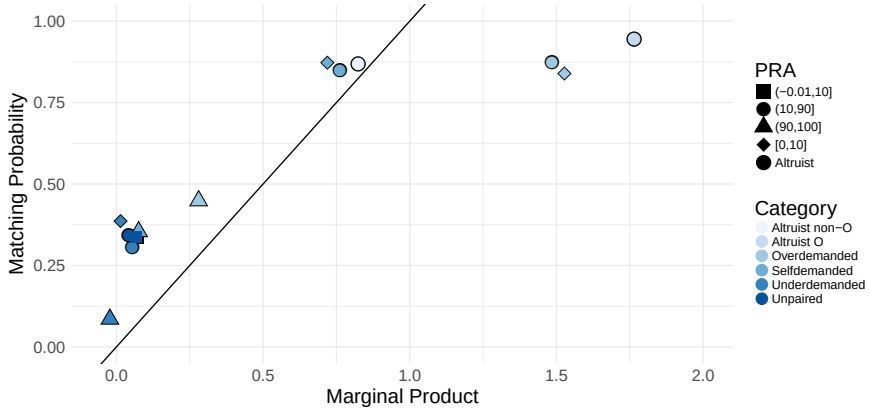
Notes: Constructed as in Table 3.



(a) Baseline



(b) Higher Wait-time Lower Friction



(c) Lower Wait-time Higher Friction

Figure D12: Robustness: Private versus Socially Optimal Rewards for Submission Types

Notes: Constructed as in Figure 7.

Table D7: Robustness: Points System

	Match Probability			Marginal Product			Points per Transplantation		
	Baseline	Higher Waittime Lower Friction	Lower Waittime Higher Friction	Baseline	Higher Waittime Lower Friction	Lower Waittime Higher Friction	Baseline	Higher Waittime Lower Friction	Lower Waittime Higher Friction
Panel A: Altruistic Donors									
Non-O Donor	0.86	0.85	0.87	0.84	0.62	0.82	0.97	0.74	0.95
O Donor	0.94	0.93	0.94	1.86	1.83	1.76	1.99	1.98	1.87
Panel B: Patient-Donor Pairs									
O Patient, Non-O Donor	0.27	0.29	0.28	0.08	0.03	0.00	-0.72	-0.89	-1.00
O Patient, O Donor, PRA >= 91%	0.29	0.30	0.28	0.13	0.10	0.06	-0.55	-0.67	-0.78
O Patient, O Donor, PRA < 91%	0.81	0.81	0.82	0.72	0.72	0.79	-0.11	-0.12	-0.04
Non-O Patient, O Donor, PRA >= 96%	0.28	0.29	0.27	0.09	0.08	0.01	-0.66	-0.73	-0.96
Non-O Patient, Non-O Donor, PRA < 96%	0.85	0.86	0.87	0.69	0.68	0.62	-0.19	-0.21	-0.28
Non-O Patient, O Donor, PRA < 96%	0.84	0.84	0.86	1.44	1.40	1.44	0.72	0.66	0.68
Panel C: Unpaired Patients									
Unpaired Patients	0.24	0.23	0.24	0.07	-0.03	0.02	-0.71	-1.12	-0.92

Notes: Constructed as in Table C5.

References

Anderson, Ross, Itai Ashlagi, David Gamarnik, and Alvin E Roth, "Finding long chains in kidney exchange using the traveling salesman problem.," *Proceedings of the National Academy of Sciences of the United States of America*, jan 2015, *112* (3), 663–8.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani, *The Elements of Statistical Learning*, Vol. 1, Springer series in statistics New York, 2001.

Johnson, Donald B, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM (JACM)*, 1977, *24* (1), 1–13.

Mas-Colell, Andreu, Michael Dennis Whinston, and Jerry R Green, *Microeconomic Theory*, Oxford University Press, 1995.

Morris, Carl N, "Parametric empirical Bayes inference: theory and applications," *Journal of the American Statistical Association*, 1983, *78* (381), 47–55.

Robert, Christian and George Casella, "Monte Carlo Statistical Methods Springer-Verlag," *New York*, 2004.