Published in final edited form as:

Stat Biosci. 2017 December ; 9(2): 453–469. doi:10.1007/s12561-016-9155-y.

A Look-Ahead Strategy for Non-Directed Donors in Kidney Paired Donation

Wen Wang, Mathieu Bray, Peter X.-K. Song, and John D. Kalbfleisch

Department of Biostatistics, University of Michigan, Ann Arbor, MI, Kidney Epidemiology and Cost Center, University of Michigan, Ann Arbor, MI

Abstract

While there is a growing need for kidney transplants to treat end stage kidney disease, the supply of transplantable kidneys is in serious shortage. Kidney paired donation (KPD) programs serve as platforms for candidates with willing but incompatible donors to assess the possibility of exchanging donors, thus opening up new transplant opportunities for these candidates. In recent years, non-directed (or altruistic) donors (NDDs) have been incorporated into KPD programs beginning chains of transplants that benefit many candidates. In such programs, making optimal decisions in transplant exchange selection is of critical importance. With the aim of improving the selection of chains beginning with an NDD, this paper introduces a look-ahead multiple decision strategy to select chains, that are easy to extend in the future. Simulation studies are adopted to assess performance of this strategy. Taking into account the extensibility of chains increases the number of realized transplants.

Keywords

altruistic donor; bridge donor; fallback; multiple decisions; non-directed donor

1 Introduction

Kidney transplants are preferred over dialysis for treatment of end stage kidney disease (ESKD), both as a means to improve expected survival and to provide a better quality of life (Port et al. 1993). Based on data from the Organ Procurement and Transplantation Network (OPTN), US demand for kidney transplants greatly exceeds supply, and the gap is increasing (Organ Procurement and Transplantation Network 2015), as shown in Figure 1. The proportion of transplants from living donors is about 37% overall; both the proportion and the number of living donor transplants has decreased slightly over the last ten years, 2005–2014.

One of the major hurdles preventing kidney transplants is biological incompatibility, either due to ABO blood type mismatch¹ or the candidate's sensitivity to human leukocyte antigens (HLA) of the donor. These sensitivities can arise through previous blood

¹There are four principal blood types: A, B, AB and O. Type O can donate to any of the four blood types. Type B can donate to type B and AB, type A can donate to A or AB, and type AB can only donate to type AB. Candidates of blood type O tend to be disadvantaged since they can receive only from an O donor, but O donors can give to any blood type.

transfusions, pregnancy, or other reasons and typically prohibit transplantation since they can lead to rejection of the kidney (Ashlagi et al. 2011). Compatibility of a donor and a candidate is first assessed by virtual crossmatch, a computerized assessment of ABO blood type compatibility and candidate sensitivities to donor HLA antigens, which later must be confirmed by a lab crossmatch to guarantee compatibility prior to transplantation. Patients having a willing but incompatible donor cannot proceed with transplantation. To overcome such incompatibilities, kidney paired donation (KPD) was proposed as an ethical and practical solution (Park et al. 1999; Rapaport 1986; Ross et al. 1997). Now there are at least three national KPD programs and at least ten center-specific and regional KPD programs (Ellison 2014) in U.S..

A KPD program (or KPD pool) consists of pairs, each of which is comprised of a candidate in need of a kidney transplant and her/his incompatible donor, as well as non-directed donors (NDDs) who are willing to donate to a candidate in the pool, but with no specific associated candidate. Due to the nature of organ exchanges, transplants in a KPD pool are organized either in cycles of incompatible pairs, or in chains initiated by NDDs. Figure 2a displays a three-way cycle involving three candidate-donor pairs, in which pairs can be arranged in a cyclic sequence and the donor in each pair is compatible with the candidate in the next pair. As illustrated in Figure 2b, a chain of length 2 is initiated by an NDD followed by a sequence of two pairs, in which the NDD is compatible with the candidate in the first pair, and the donor in each subsequent pair is compatible with the candidate in the following pair. At the end of this chain, the donor of the last pair is referred to as a bridge donor (BD) and plays the role of an NDD in creating a new chain sometime in the future. Because of this continuation, the total chain in the KPD program may be built up by a sequence of shorter chains, or chain segments, linked together through the BDs. In practice, short chain segments are preferred given hospital and surgical limitations and inherent failure probabilities discussed below.

In a KPD program, match runs based on virtual crossmatches are typically conducted on a regular basis with the aim of selecting disjoint chains and cycles from the pairs and NDDs currently in the pool (e.g. Fumo et al. 2015). In this paper, we concentrate on the identification of suitable chains, utilizing the NDDs (and BDs) available at the start of the match run. Potential transplants selected in a match run can fail to proceed to transplant typically for any of the following three reasons. First, as noted above, the compatibility of the proposed transplant must be confirmed by a lab crossmatch and in addition, both the donor and the recipient must agree to the proposal. Second, donors or candidates may be unavailable due to sickness, pregnancy, business or family obligations, other transplant opportunities or death. We refer to a proposed transplant as viable if the compatibility is confirmed by lab crossmatch and the donor and candidate involved in the transplant are available. Third, a transplant can fail to proceed because one of the other transplants in the same cycle or chain segment cannot proceed to transplant for either of the first two reasons. For instance, in Figure 2b, the transplant from D_1 to C_2 falls apart if the transplant from the NDD to C_1 fails. In our study, a potential chain is selected and transplanted if and only if all involved transplants are viable. That is, a given chain will not be transplanted if any transplant in the chain is not viable.

In practice, only relatively shorter chains, under a certain length, are considered in each match run due to computational limitations in the optimization algorithms, but more importantly, due to the high failure probability of long chains. Our study objective concerns the assessment of strategies to select chains initiated by NDDs in a KPD pool. Specifically, we consider selecting chains of a given length where the value of each potential chain is assessed by taking account of the value of the BD to the KPD at the end of the selected chain. This value is obtained by a “look-ahead” strategy (Li et al. 2014) which considers the potential for the BD to generate future transplants by evaluating outcomes over a horizon of given length. Thus, we consider combinations of chain segment lengths (referred to as decision number) and various look-ahead horizons and evaluate these strategies in a simulation study.

This paper is organized as follows. Section 2 describes a graphic representation of KPD programs and a brief review of BD utility calculation (Li et al. 2014). Section 3 introduces a look-ahead multiple decision strategy. Section 4 describes a simulation study to explore performance of the strategy in Section 3. Section 5 presents results of the simulation study. Section 6 gives concluding remarks.

2 Formulation

2.1 Graphic Representation

$a \cup \mathcal{V}_p$ where $\mathcal{V}_a$ (“a” stands for “altruistic”) is the set of m NDDs and $\mathcal{V}_p$ is the set of n incompatible pairs. The edge set, $\mathcal{E}$, is a set of ordered pairs of vertices in $\mathcal{V}$. Specifically, for any two vertices v_1 and v_2 in $\mathcal{V}$, $(v_1, v_2) \in \mathcal{E}$ if and only if the donor in pair (or NDD) v_1 is predicted by virtual crossmatch to be compatible with the candidate in pair v_2 . For the rest of this article, “vertex” and “pair or NDD” are used interchangeably, and so are “edge” and “potential transplant”. A chain of length d is a vertex sequence, denoted by $c = [v_0, v_1, \dots, v_d]$, in which $v_0, v_1, \dots, v_d$ are distinct and $v_0 \in \mathcal{V}_a$, $v_i \in \mathcal{V}_p$ and $(v_{i-1}, v_i) \in \mathcal{E}$ for $i = 1, 2, \dots, d$. As a matter of notation, we sometimes denote a longer chain $c' = [v_0, v_1, \dots, v_d, v_{d+1}]$ as $c' = [c, v_{d+1}]$ where $c = [v_0, v_1, \dots, v_d]$, and the set of all vertices involved in a chain c is denoted by $\{c\}$.

To address the issue of transplant failure preventing a proposed transplant from proceeding, we define probabilities of success on the vertices and on the edges. The edge probability function, $p^e: \mathcal{E} \rightarrow [0,1]$, is the probability that a negative (compatible) virtual crossmatch results in a negative lab crossmatch and that the potential transplant is acceptable to the associated donor and recipient. To incorporate the chance of (temporary) unavailability of pairs and NDDs before transplantation, we also consider a vertex probability function, $p^v: \mathcal{V} \rightarrow [0,1]$, where $p^v(v_i)$ is the probability that a selected vertex v_i is found to be available for a transplant in a match run. Unavailability is taken as a temporary condition and the pair or NDD will again be present in the next match run unless the pair or NDD withdraws. The probability that a vertex permanently withdraws from the KPD program before each match run is set as p^{wp} for pairs and p^{wa} for NDDs. Failure of any edge, temporary unavailability and permanent withdrawal are assumed to be mutually independent.

To complete the specification of the KPD, we also consider an edge utility function $u^e: \mathcal{E} \rightarrow [0, \infty)$, where $u^e(v_i, v_j)$ quantifies the value assigned to a transplant of the donor kidney in pair or NDD v_i to the candidate in pair v_j . In general terms, the aim of an algorithm for selection of transplants in a KPD is to maximize the total realized utility of the transplants actually performed. Assigning the value of 1 to each potential transplant is equivalent to maximizing the total number of transplants. In some cases, however, it is useful to allow the utility to depend on characteristics of the candidate and/or the donor. Thus some KPDs (e.g. Alliance for Paired Donation (APD)) use a point scheme to prioritize possible transplants, where precedence is given to transplants involving, for example, candidates with high PRA, long waiting time in the KPD, or shorter physical distance between the donor and the candidate (Fumo et al. 2015). In other cases, the utility may reflect the estimated graft survival of the transplant, so that the aim of the algorithm is to maximize the number of transplants for which the graft survives for, say, five years.

2.2 Calculation of Chain Utilities

To compare chains, a utility is assigned to each chain representing its value. A naive chain utility is the sum of the edge utilities of the transplants involved in the chain. However, this naive chain utility does not take account of the degree to which the BD can be expected to be useful to the KPD program in future match runs. For example, if a chain ends with a BD of blood type AB or with HLAs against which the majority of the population of candidates has antibodies, he or she will be incompatible with the majority of current and future candidates in the KPD pool.

To overcome this shortsightedness, and following ideas in previous literature (Li et al. 2014), we propose evaluation of an *s-step look-ahead BD utility* for a given chain c ,

$$u_{BD}^s(c): \mathcal{C} \rightarrow [0, \infty),$$

where $\mathcal{C}$ is the set of all chains and s , a non-negative integer, corresponds to the number of additional transplants being considered after the completion of the chain c . For $s = 0$, $u_{BD}^0 \equiv 0$. For $s = 1$, the *1-step BD utility* is defined as the expected maximum value among utilities of viable edges stemming from the BD associated with c . To elucidate this definition, we use a toy example in Figure 3. First, the BD of the chain $c = [v_0, v_1, v_2]$ is the donor in vertex v_2 . Initiated by v_2 , all possible immediately subsequent transplants of the chain $c = [v_0, v_1, v_2]$ are (v_2, v_3) and (v_2, v_4) , shown by dashed arrows. We denote $u^e(v_i, v_j)$ by u_{ij}^e for $i \neq j$, and without loss of generality, assume that $u_{23}^e \geq u_{24}^e$. Then the *1-step look-ahead BD utility* of v_2 in the chain $c = [v_0, v_1, v_2]$ is

$$u_{BD}^1(c) = p_{23}^e p_3^v u_{23}^e + (1 - p_{23}^e p_3^v) p_{24}^e p_4^v u_{24}^e, \quad (1)$$

²For clarity, functions introduced in Section 2.1 are denoted by f_X (or f_{XY}) instead of $f(x)$ (or $f(x, y)$) in Section 2.2.

which calculates the expected utility obtained in one future step from the BD, v_2 . Proceeding in this manner, we can further take account of the utility of the BDs in the potential chains continuing from v_2 ; thus we might consider accounting for the *1-step look-ahead BD utilities* of v_3 and v_4 . Given expanded chains $[c, v_3]$ and $[c, v_4]$, their *1-step look-ahead BD utilities* $u_{BD}^1([c, v_3])$ and $u_{BD}^1([c, v_4])$ are calculated by equation (1). Again, without loss of generality, we assume that $u_{23}^e + u_{BD}^1([c, v_3]) \geq u_{24}^e + u_{BD}^1([c, v_4])$. The *2-step look-ahead BD utility* is

$$u_{BD}^2(c) = p_{23}^e p_3^v \{u_{23}^e + u_{BD}^1([c, v_3])\} + (1 - p_{23}^e p_3^v) p_{24}^e p_4^v \{u_{24}^e + u_{BD}^1([c, v_4])\},$$

which represents the expected total edge utility obtained in two future steps emanating from the BD, v_2 .

Evidently, we may generalize this calculation to obtain the *s-step look-ahead BD utility* ($s \geq 1$). In effect, this calculation can be defined and obtained iteratively for a general chain $c = [v_0, v_1, \dots, v_d]$ of length d with v_d being the BD. Specifically, the *s-step look-ahead BD utility* is

$$u_{BD}^s(c) = \sum_{i=1}^L \left[\left\{ \prod_{j=1}^{i-1} (1 - p_{v_d w_j}^e p_{w_j}^v) \right\} p_{v_d w_i}^e p_{w_i}^v \{u_{v_d w_i}^e + u_{BD}^{s-1}([c, w_i])\} \right], s \geq 1, \quad (2)$$

where $\{w_l\}_{l=1}^L$ is the set of all vertices that are immediate offspring of v_d in $\mathcal{V}_p \setminus \{c\}$. Further, the vertices are labeled so that $u_{v_d w_l}^e + u_{BD}^{s-1}([c, w_l])$, $l = 1, \dots, L$ are in descending order of magnitude. Note that the domain of u_{BD}^s is $\mathcal{C}$ as opposed to $\mathcal{V}_p$ because to calculate the bridge donor utility for the BD of a chain, vertices involved in the chain must be excluded from immediate offspring of the BD.

Finally, we define $u_C^s(c)$, the *s-step look-ahead chain utility* of $c = [v_0, \dots, v_d]$ as the sum of the naive chain utility and the *s-step look-ahead BD utility* in equation (2). Thus, we write

$$u_C^s(c) = \sum_{i=1}^d u_{v_{i-1} v_i}^e + u_{BD}^s(c). \quad (3)$$

3 Look-Ahead Multiple Decision Strategy

This section presents a look-ahead multiple decision strategy that considers the optimal selection of chains emanating from the NDDs in a match run, while taking account of the utility of the bridge donors that would conclude the selected chain. This general strategy is

determined by a positive integer d (the maximum chain length) and a non-negative integer s (the number of future steps to look ahead).

The approach is described as follows. We focus on chains in a set $\tilde{C}_d$ comprising all chains of length d together with chains, c' , of length less than d , whose BDs can no longer donate within the KPD pool, $\mathcal{V}_p \setminus \{c'\}$. In each match run, we consider the optimal selection of chains in $\tilde{C}_d$ where the utility of each chain is given by its s -step look-ahead chain utility in

(3). Note that, for each chain $c' = [v_0, \dots, v_d] \in \tilde{C}_d$ with $d' < d$, $u_C^s(c') = \sum_{i=1}^{d'} u_{v_{i-1}v_i}^e$ since $v_{d'}$ has no immediate offspring in $\mathcal{V}_p \setminus \{c'\}$. In implementing the strategy, the chains in $\tilde{C}_d$ are ranked in a descending order according to $u_C^s(c)$, $c \in \tilde{C}_d$. Once lab crossmatches and availability are ascertained for all chains in $\tilde{C}_d$ the viable chains are sent sequentially to implementation in order of their utility u_C^s . Once a chain is implemented, its edges and vertices (except the BD) are removed and some subsequent chains are no longer viable. Note that when a KPD has multiple NDDs, chains sent to implementation in the strategy mentioned above may not maximize the sum of the chain utilities of disjoint chains.

Each choice of s and d provides a separate strategy that can be evaluated in terms of the total utility that we expect it to generate. The greater the expected realized utility, the better the performance of the strategy. For a given pool with n pairs and m NDDs, the computational complexity of these algorithms depends on $d+s$, which we refer to as the total depth of the strategy. Depending on computational capacity and the size of n and m , the strategies contemplated would typically be limited to relatively smaller values of $d+s$. A question of some interest is how s and d are best selected for a given total depth, $d+s$, and how much the realized utility varies as $d+s$ increased. Our (partial) answers to these questions are based on an extensive simulation study as described in the subsequent section. In following sections, we refer the look-ahead multiple decision strategy with d and s as the (d,s) look-ahead strategy.

4 Simulation Study

To evaluate the effect of different d and s combinations on the performance of the (d,s) look-ahead strategy, we conduct extensive simulation studies. To limit the computational burden, the total depth is constrained to $d+s \leq 6$ in all simulation experiments. In virtual crossmatches, determination of candidate sensitivities to donor HLA antigens (specifically HLA A, B, BW4, BW6, CW, DQ, DR, DR51, DR52 and DR53) is made according to the HLA equivalences table in (Organ Procurement and Transplantation Network 2010). Following (Ashlagi et al. 2011), the edge probability function p^e is taken to depend on the panel reactive antibody (PRA) level of the candidate; see Table 1. Note that PRA is a measure of the extent to which the candidate is known to be sensitized against HLA antigens in the general population. For example, if candidate's PRA is 0.75, then the candidate is sensitized to 75% of the population to which the candidate being tested. We consider two edge utility functions: the first simply assigns to each edge a constant, $u^e \equiv 1$, which essentially asserts that all transplants are equally valuable. As such, our aim is to optimize the number of transplants. The second, $u^e(v_1, v_2)$, $(v_1, v_2) \in \mathcal{E}$, is generated from a uniform $U(10,30)$ distribution so that the utility is allowed to vary among edges. In all simulations

vertex probabilities $p^v(v_1)$, $v_1 \in \mathcal{V}$ are generated independently according to a uniform $U(0.75, 1.00)$ distribution. In addition, we specify withdrawal probabilities, $p^{wp} = 0.02$ for pairs, $p^{wa} = 0.01$ for NDDs. Furthermore, a long chain of length more than 3 is implemented sequentially in two shorter subchains. We give the donor at the break point a probability of 0.005 to permanently withdraw from the KPD pool. When this happens, the second subchain fails to materialize.

KPD pools are simulated by randomly drawing NDDs and pairs with replacement from a combined database of data on pairs and NDDs from the APD and the University of Michigan (UM) KPD programs. There are 585 pairs and 56 NDDs in the APD data and 281 pairs and 7 NDDs in the University of Michigan KPD program. In both data sources, blood type, PRA level and anti-HLA antibodies of each candidate and blood type and HLA antigens of each donor are used. We evaluated the strategies with simulations in a static and a dynamic KPD pool, as described in the next two subsections.

4.1 Evaluation of Static KPD Program

Static KPD pools represent a kind of idealized KPD program whereby an initial pool is created at random and a series of match runs applied, with no new pairs or NDDs entering the pool over the time of the match runs. Such pools have been examined in previous literature, e.g. (Abraham et al. 2007). In our experiments, static pools are simulated to see how long a chain is obtained on average as a result of each (d, s) look-ahead strategy.

In the static simulations, 1000 KPD programs with 1 NDD and 80 pairs are generated at random. In each simulated KPD program, edges are built by virtual cross-matches and edge probabilities and utilities, p^e and u^e , as well as vertex probabilities p^v are assigned as described above. Based on the probabilities, p^e , lab crossmatch results are generated for all edges and recorded. Selections of edges in each match run are made without knowledge of the results of the lab crossmatches unless the edge has been selected and checked on a previous match run. Match runs are conducted until the NDD (or the BD of the last implemented chain segment) can no longer be extended within the KPD pool. Each of the 1000 simulated KPDs is analyzed using each (d, s) look-ahead strategy.

The details of the implementations are as follows. For a given (d, s) combination, each match run involves the following steps:

1. Obtain the set $\tilde{C}_d$, calculate $u_C^s(c)$ for each $c \in \tilde{C}_d$ and rank the chains in $\tilde{C}_d$ in a descending order.
2. Ascertain lab crossmatch results of edges and temporary availability of vertices in all chains $c \in \tilde{C}_d$ and remove from the graph those edges with positive lab crossmatch results.
3. Implement the highest ranked viable chain. As noted, potential chains of length 4 or more are assumed to be performed in two subchains and the second subchain fails to materialize with a small probability. Then remove all vertices of this chain (except the BD) from the KPD pool. The BD will be treated as an NDD in

the next match run. All unused vertices, including those that are temporarily unavailable, are returned to the pool for the next match run.

4. Remove each pair and the NDD permanently from the pool according to the probabilities p^{wp} and p^{wa} , respectively. The pool for the next match run is formed to include remaining vertices and any edges connecting them that have not been observed with positive lab crossmatch results.

The above steps 1–4 are repeated until no further extension of the chain is possible.

4.2 Evaluation of Dynamic KPD Program

A dynamic KPD pool is a more realistic model of a KPD program with new pairs and NDDs entering the pool for each match run. They are also simulated to evaluate (d,s) look-ahead strategies in a manner similar to the static KPD pool. Since new vertices enter the KPD programs for each match run, we limit the number of match runs as opposed to running match runs exhaustively until depletion of chains, as in the static KPD case in Section 4.1. For each KPD program, 8 match runs are scheduled. To make fair comparisons among strategies, in each match run every strategy is allowed to schedule up to 6 transplants, if possible, for each NDD. With $d=1$ and any s , for example, each match run considers six successive implementations of the $(1,s)$ lookahead strategy. With $d=2$ or $d=3$ and any s , each match run consists of three or two successive implementations respectively. With $d=4$, a $(4,s)$ look-ahead strategy is implemented, followed by a $(2,s)$ strategy in each of the eight match runs. Similarly for the case, $d=5$, a $(5,s)$ look-ahead strategy is followed by a $(1,s)$ strategy.

In simulations for the dynamic KPD pool, 500 replications are performed. Prior to each of the eight match runs, one NDD and 30 pairs are added to the KPD pool; these are selected at random with replacement from the combined database. For each given (d,s) , match runs proceed as follows:

1. Create a set of new vertices and a set of new edges by running virtual cross-matches among the new vertices and among old and new vertices. Then assign edge probabilities and utilities as well as vertex probabilities on the elements in both sets as before.
2. Set the Remaining Decision Number (RDN) to 6.
3. Set $d^* = \min\{d, \text{RDN}\}$ and carry out the (d^*,s) look-ahead strategy on the remaining pool. Carry out viable chain(s) in $\tilde{C}_d$ in descending order of s -step lookahead chain utility. Remove those vertices (except for the BDs) and edges in those viable chains from the pool. Also remove any edges that were determined by lab crossmatch to be unviable. At the end, set $\text{RDN} = \text{RDN} - d^*$.
4. Repeat step 3 if RDN is not 0.
5. Remove a pair or an NDD permanently from the pool according to the probabilities p^{wa} and p^{wp} , respectively. The pool for the next match run is formed to be the remaining vertices and any edges connecting them that have not been observed with positive lab crossmatch results.

5 Results

We only report results for the (d,s) *look-ahead strategies* obtained from the dynamic KPD simulations with $u^e \equiv 1$ under the parameter settings described above. The results of the static KPD simulations and simulations with variable utilities are similar to those of the dynamic KPD simulations with constant utilities. Thus they are omitted here.

5.1 Runtime

Before presenting results, we summarize in Table 2 the computation time to carry out the simulation for each (d,s) strategy in the dynamic case. As mentioned in Section 3, the computational complexity of a (d,s) *look-ahead strategy* increases as the total depth increases. And the computational complexities of (d,s) *look-ahead strategies* with the same total depth are of similar order of magnitude.

5.2 Number of Transplants

Figure 4 shows the mean number of transplants for 8 match runs and empirical 95% confidence intervals for the dynamic KPD program simulations with $u^e \equiv 1$. As the total depth $d + s$ increases, the maximum mean number of transplants increases, but with diminishing returns. When the total depth is fixed, the strategy with $s = 0$, i.e. without looking ahead, performs the worst, while those strategies with $d \approx s$ perform the best. This suggests that to obtain a larger realized utility, one should consider a larger total depth given computational feasibility. Under a fixed total depth, a strategy that takes a balance between benefit at present (d) and benefit in the future (s) is preferred.

To gain more insight as to why the best strategy with a fixed total depth is given by the choice of $d \approx s$, static KPD program simulations as described in Section 4.1 with $u^e \equiv 1$, $p^v \equiv 1$ and p^e fixed at 0.2, 0.4, 0.6, 0.8 and 1 respectively were conducted. Similar static KPD program simulations were conducted with $u^e \equiv 1$, $p^e \equiv 1$ and p^v fixed at 0.2, 0.4, 0.6, 0.8 and 1 respectively. As shown in Figure 5a and Figure 5b, when p^e and p^v are increasing respectively, the best strategy with a fixed total depth changes from $d = 1$ to $d \approx s$. This suggests that when the probability of a successful transplant is very low, strategies making fewer decisions and looking more steps ahead are advantageous among those with the fixed total depth. When the probability of a successful transplant is moderate to high, however, strategies with $d \approx s$ are advantageous among those with a fixed total depth.

5.3 Blood Type Distribution

Figure 6 shows the blood type distribution among the candidate population of the combined database and among the candidates who receive transplants in simulated (d,s) *look-ahead strategies*. Almost 60 percent of the candidate population of the combined database is blood type O, the blood type that is most difficult to match. With a fixed total depth, the strategy with d slightly larger than a half of the total depth shows the largest gain to transplant candidates with blood type O. As the total depth increases, the proportion of type O candidates receiving transplants also increases, although with diminishing returns. It is interesting to note that for a fixed $d + s$, the best strategy with $d \approx s$ in terms of number of transplants also admits a higher percentage of type O transplanted candidates.

5.4 PRA level Distribution

Figure 7 shows the PRA distribution of the candidate population of the combined database and candidates who receive transplants in simulated (d,s) look-ahead strategies. Almost 50 percent of the candidate population of the combined database has a PRA level of 75 or higher. These are candidates who are sensitized to the HLA antigens of at least 75% of the US population; the higher a candidate's PRA, the more difficult he/she is to match. With a fixed total depth, the strategy with $d \approx s$ shows the largest gain in transplanted candidates with very high PRA. As the total depth increases, the proportion of candidates with very high PRA receiving transplants also increases, although with diminishing returns. Again, for a fixed $d + s$, the strategy with $d \approx s$ tends to optimize both the number of transplants and the percentage of transplanted candidates with very high PRA.

5.5 Number of Lab Crossmatches

To examine the efficiency of the strategies, we display in Figure 8 and Figure 9 the mean number of lab crossmatches in 8 match runs and the number of lab cross-matches per realized transplant in the dynamic KPD simulations respectively. The mean number of lab crossmatches in strategies with the same d is of similar magnitude. As d increases, the mean number of lab crossmatches increases. With the d being fixed, as s increases, the number of lab crossmatches per realized transplant decreases. With a fixed total depth, the number of lab crossmatches per unit realized transplant increases as d increases. In other words, the strategies with $d = 1$ are the most cost-efficient, but generate relatively fewer transplants as seen in Figure 4. As d becomes larger (e.g. $d > 3$) the number of required lab crossmatches increases very rapidly, but as noted above, with little gain in transplants achieved.

6 Concluding Remarks

A good dynamic decision-making strategy ought to consider both current and future benefits, and a balance of these two is important; the aim is to achieve a near optimal solution at the present time while attempting to assure future opportunities. The KPD is a complex dynamical system with many hard-to-match candidates and a limited number of easy-to-match donors. Studying and comparing different strategies for using existing and future resources can provide a useful approach to their management. In this paper, we focus on the BD utility in selecting chains initiated by NDDs; this provides a needed balance between current gains and potential future gains and, as our simulations show, this approach can achieve better results than one which ignores future outcomes ($s = 0$). It should be noted that this conclusion refers to relatively large KPD programs, as are considered in the simulations. For example, in a very small KPD pool with all chains of size $d + s$ or less, there would be no advantage to look-ahead strategies ($s > 0$) since the optimal decision is easily identified.

We have conducted extensive simulation experiments to understand the pros and cons concerning possible (d,s) look-ahead strategies. We considered the number of realized transplants as the primary objective for the measure of current benefit; secondary objectives, such as the number of transplants for candidates of blood type O, the number of transplants for candidates with high PRA level, and cost-effectiveness are also considered. Our

simulation results suggest that increasing the total depth $d + s$ results in more transplants on average, although the gains beyond $d + s = 4$ seem to be relatively small. In comparing various strategies with $d + s$ fixed, our simulation results suggest that a desirable strategy should strike a balance with $d \approx s$. It is worth pointing out that the strategies most commonly used in KPD programs do not consider any look-ahead steps, and effectively take $s = 0$. In our simulation studies, this approach performed poorly for both primary and secondary objectives for optimization.

In our simulated implementation of the (d, s) look-ahead strategy, we checked for viability of all potential chains in each match run; in applications, this approach would be financially and administratively burdensome for larger values of d and a more efficient approach is possible. For example, lab crossmatches and vertex availabilities may be ascertained for a small number of selected chains with the highest chain utilities. Once this is completed, associated vertices and edges could be removed from the remaining selected chains and KPD pool. If the optimal chain has been identified, we carry out the associated transplants; otherwise, we proceed with additional tests as needed by assessing the next small number of chains. We have carried out some preliminary evaluations of such strategies and these seem promising in reducing costs. In our simulations, we considered all crossmatches as being done since we were interested in comparing (d, s) and $(d + s, 0)$ look-ahead strategies with regard to transplants obtained, without introducing constraints. The results show little advantage to $(d + s, 0)$ look-ahead strategy even with the extensive crossmatches and suggest concentration on that of (d, s) .

In an actual KPD program, there will be new pairs entering the program between match runs, so taking that into account in assessing the value of a BD is natural. It would be of interest to do simulations in which the value of a BD depends not only on the population in the current pool but also on a simulated population entering the KPD program in the future.

It has been the experience of many KPDs that long chains are in fact built up through a sequence of shorter chain segments connected together through the bridge donor. A recent paper (Fumo et al. 2015) expands on this issue in the context of the experience of the APD, and other papers (Bray et al. 2015; Li et al. 2014) on matching strategies have found that using shorter chain segments is as efficient as long chains, even in the context of the usual match run approach.

In practice, both chains and cycles are possible in KPD programs, although in recent times chains have become more frequently used (Fumo et al. 2015). A useful extension of the proposed utility-based method is to incorporate cycles into the lookahead multiple decision strategy. In most current implementations, disjoint chains and cycles are selected to optimize the total expected utility by solving an integer programming problem in each match run (Ashlagi et al. 2011; Bray et al. 2015; Li et al. 2014). Another approach would be to alternate between chains and cycles in match runs, using look-ahead strategies for the chains and traditional optimization approaches for the cycles. A comparison of these two approaches would be of interest.

Acknowledgments

This research was supported by the National Institutes of Diabetes and Digestive and Kidney Diseases (NIDDK) through grant number 1R01-DK093513. We thank Dr Michael A. Rees and Alan B. Leichtman for helpful discussions and for making available program data from the APD and the UM KPD respectively.

References

- Abraham, DJ., Blum, A., Sandholm, T. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. *Proceedings of the 8th ACM conference on Electronic commerce*; ACM; 2007. p. 295-304.
- Ashlagi I, Gilchrist DS, Roth AE, Rees MA. Nonsimultaneous chains and dominos in kidney-paired donation revisited. *American Journal of transplantation*. 2011; 11(5):984–994. [PubMed: 21521469]
- Bray M, Wang W, Song PK, Leichtman A, Rees M, Ashby V, Eikstadt R, Goulding A, Kalbfleisch J. Planning for uncertainty and fallbacks can increase the number of transplants in a kidney-paired donation program. *American Journal of Transplantation*. 2015; 15(10):2636–2645. [PubMed: 26372837]
- Ellison B. A systematic review of kidney paired donation: Applying lessons from historic and contemporary case studies to improve the US model. *Wharton Research Scholars Journal*. 2014; (107)
- Fumo D, Kapoor V, Reece L, Stepkowski S, Kopke J, Rees S, Smith C, Roth A, Leichtman A, Rees M. Historical matching strategies in kidney paired donation: The 7-year evolution of a web-based virtual matching system. *American Journal of Transplantation*. 2015
- Li Y, Song P, K, Leichtman AB, Rees MA, Kalbfleisch JD. Decision making in kidney paired donation programs with altruistic donors. *SORT (Barcelona)*. 2014; 38(1):53. [PubMed: 25309603]
- Organ Procurement and Transplantation Network. Proposal to update HLA equivalences tables. 2010. http://optn.transplant.hrsa.gov/PublicComment/pubcommentPropSub_261.pdf
- Organ Procurement and Transplantation Network. Organ procurement and transplantation network. 2015. <http://optn.transplant.hrsa.gov>
- Park K, Moon JI, Kim SI, Kim YS. Exchange donor program in kidney trans-plantation1. *Transplantation*. 1999; 67(2):336–338. [PubMed: 10075605]
- Port FK, Wolfe RA, Mauger EA, Berling DP, Jiang K. Comparison of survival probabilities for dialysis patients vs cadaveric renal transplant recipients. *Jama*. 1993; 270(11):1339–1343. [PubMed: 8360969]
- Rapaport FT. The case for a living emotionally related international kidney donor exchange registry. *Transplantation proceedings*. 1986; 18:5–9.
- Ross LF, Rubin DT, Siegler M, Josephson MA, Thistlethwaite JR, Woodle ES. Ethics of a paired-kidney-exchange program. *The New England Journal of Medicine*. 1997; 336(24):1752–1755. [PubMed: 9180096]

Fig. 1. Annual performed kidney transplants, annual performed deceased donor kidney transplants, annual performed living donor kidney transplants and annual newly registered candidates for kidney transplants in the US from 1995 to 2014.

Fig. 2.
Examples of a 3-way cycle and a chain of length 2 (i.e. two potential transplants).
Compatibility is indicated by directed edges pointing from donors to candidates.

Fig. 3.

An example of BD utility calculation for a chain $c = [v_0, v_1, v_2]$ of length 2. Empty circles denote an NDD and two pairs in the chain, and each solid arrow directed from v_i to v_j denotes a possible transplant from pair v_i to pair v_j . Solid circles and dotted arrows denote future pairs and transplants that could be initiated by the BD in pair v_2 .

Fig. 4.
Mean realized utility of dynamic pools with $u^e \equiv 1$ over 8 match runs.

(a) $u^e \equiv 1$ and constant p^e fixed at various values.

(b) $u^v \equiv 1$ and constant p^v fixed at various values.

Fig. 5. Mean realized utility of static pool simulations with constant edge/vertex probabilities over 8 match runs.

Fig. 6. Blood type distribution of transplanted candidates of dynamic pools with $u^e \equiv 1$ over 8 match runs.

Fig. 7.
PRA level distribution of transplanted candidates of dynamic pools with $u^e \equiv 1$ over 8 match runs.

Fig. 8.

Mean number of lab crossmatches in all match runs of dynamic pools with $u^e \equiv 1$.

Fig. 9.
Number of lab crossmatches per unit realized utility of dynamic pools with $u^e \equiv 1$.

Table 1

Negative lab crossmatch probabilities, conditional on negative virtual cross-match, according to the highest value of previously measured panel reactive antibody (PRA) level of candidates.

PRA level	PRA level	Probability of match success
very high	75–100	0.50
high	50–74	0.65
moderate	25–49	0.80
low	0–24	0.95

Table 2
Mean runtime in millisecond of dynamic pool simulations with $\mathcal{H}^p \equiv 1$ over 8 match runs.

$\frac{s}{d}$	0	1	2	3	4	5
1	3.69×10^0	2.83×10^1	7.76×10^2	1.55×10^3	8.88×10^3	6.70×10^4
2	1.77×10^1	1.02×10^2	5.67×10^2	3.13×10^3	2.71×10^4	-
3	1.08×10^2	5.56×10^2	3.23×10^3	2.70×10^4	-	-
4	1.20×10^3	2.74×10^3	1.98×10^4	-	-	-
5	1.07×10^4	2.37×10^4	-	-	-	-
6	1.12×10^5	-	-	-	-	-