Optimization and Simulation of Kidney Paired Donation Programs

by

Yijiang Li

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biostatistics)
in The University of Michigan
2012

Doctoral Committee:

Professor John D. Kalbfleisch, Co-Chair
Professor Peter X. Song, Co-Chair
Professor Alan B. Leichtman
Professor Kerby A. Shedden

To my parents and wife for their love

Acknowledgments

My greatest thanks are due to my two advisors, Dr. Jack Kalbfleisch and Dr. Peter Song. On one hand, Dr. Kalbfleisch has deeply influenced my understanding and appreciation of statistics and scientific research in general; to me, he is an incomparable role model. On the other hand, he is simply a very nice Canadian guy named Jack, always accessible and always helpful with great patience. My other advisor, Peter, is an extremely open-minded professor. He is very inspirational to chat with (not limited to research). He cares his students the most, and often puts a student's deadline before his own. I have received so much more from Peter than I could possibly hope for. Jack and Peter, thank you very much for all of your help and advice throughout my doctoral work.

I am grateful to Dr. Alan Leichtman for his constant support. Alan has mentored, encouraged and cared for me for almost three years. His many invaluable suggestions and insights have greatly improved the quality of this dissertation. I would also like to thank Dr. Kerby Shedden for serving on my dissertation committee. He has been very accessible and provided many useful comments on this dissertation.

I also want to express my very special thanks to John Swales and Christine Feak from the English Language Institute at the University of Michigan. I could not have written this dissertation without their tremendous help in improving my English writing skills.

I am thankful to many friends in Ann Arbor and elsewhere, particularly, Kevin He, Rick Ma, Youna Hu, Xuejing Wang, Yun Li, Guangxing Lin, Vince Marshall, Jared Foster, Xiaowei Zhan, Shan Kang, Peng Zhang, Hui Jiang, Jian Kang, Yan Zhou, Ludi Fan, Phil Westgate, Phil Boonstra, Ning, Wei, Jin Tao, Chen Zhang, Yifei Zheng, Qiong Wu, Feng Wang, Yu Zhang, Hongyuan Gao, Xiang Li, for making life more joyful.

Most of all, thank you, Mom, Dad and Fang for your unconditional love. This dissertation is dedicated to you.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

For patients with end stage renal disease (ESRD), kidney transplantation has become a preferred treatment as compared with dialysis for it provides not only a higher survival rate but also a better quality of life (Evans et al., 1985; Russell et al., 1992; Wolfe et al., 1999). Kidney transplantation is also cost-effective as compared to continuing on dialysis (Laupacis et al., 1996). However, due to a growing demand and limited supply of deceased-donor kidneys, many patients who need a transplant must be placed on long waiting lists. According to the Organ Procurement and Transplantation Network (OPTN), as of December 2010, more than 87,000 kidney transplant candidates in the U.S. were on a waiting list; and in 2009, about 34,000 candidates in the U.S. were added to the list, whereas only about 10,000 actually received a kidney transplant from a deceased donor. In response to this shortage, candidates have increasingly undergone living-donor transplants. As a matter of fact, living-donor transplants have the advantage of both short and long term higher graft survival rates than deceased-donor transplants (Terasaki et al., 1995; Hariharan et al., 2000).

A major barrier to living-donor kidney transplantation is the unfortunate circumstance that many patients with kidney failure recruit willing organ donors who, upon evaluation, prove to be ABO blood type or Human Leukocyte Antigens (HLA) incompatible. With regard to blood type compatibility, A and B donors can donate to candidates of the same blood type or of type AB; AB donors can donate only to AB candidates; and O donors, known as universal donors, can donate to candidates of any blood type, but can only receive kidneys from another O donor. The HLA incompatibility results from the presence in the blood of a candidate of pre-existing antibodies against the HLA antigens of a potential donor. Both forms of incompatibility can lead to a rapid rejection of the transplanted organ and thus prohibit transplant.

An evolving strategy in kidney paired donation (KPD) (Rapaport, 1986; Ross et al., 1997) creates a pool of incompatible pairs and seeks to match one donor-candidate pair to another pair with a complementary incompatibility, such that the donor of the first pair

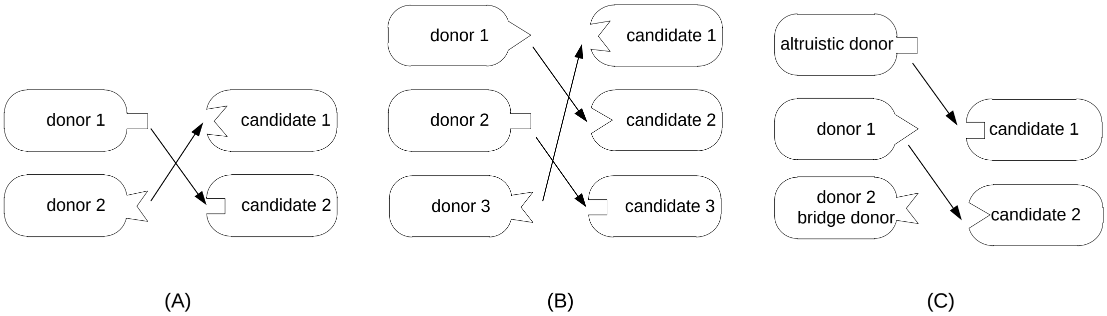

Figure 1.1: (A): A two-way exchange; (B): A three-way exchange; (C): An altruistic donor-initiated chain.

donates to the candidate of the second, and vice versa; see Figure 1.1-A and Figure 1.1-B for illustrations of a two-way exchange and a three-way exchange, respectively. This is called a kidney paired donation program; when these are large, a computerized algorithm is often used to identify appropriate exchanges. Although three-way or higher exchange cycles increase the chance of identifying compatible matches, most individual transplant centers restrict exchanges to at most three ways for two primary reasons. First, all transplants on an exchange cycle must be performed simultaneously to avoid the possibility of reneging — a requirement that would create substantial logistical difficulties of scheduling; for example, eight surgeons and eight operating rooms are required at the same time for a four-way exchange. Second, the greater the length of an exchange cycle, the less likely the potential transplants will actually occur since, if any of the proposed transplants cannot proceed, the whole cycle would collapse.

In recent years, KPD has also been extended to include living non-directed donors, or altruistic donors; an altruistic donor donates to the candidate of an incompatible donor-candidate pair with the understanding that the donor associated with the benefitted candidate in that pair will further donate to the candidate of a second pair, and so on; such a process continues and thus forms an altruistic donor-initiated chain; see Figure 1.1-C for an illustration of such a chain. Since the transplants along the chain are not required to be performed simultaneously, the donor whose candidate received another donor's kidney but has yet to donate could make a donation to a future-arriving candidate; and hence the name bridge donor. In this respect, the chain is open-ended and could potentially increase the chance for a highly sensitized candidate to receive a compatible kidney.

A growing number of KPD programs have recently been established with regional programs in the U.S. including the New England Programs for Kidney Exchange, the Paired Kidney Exchange Program at Johns Hopkins Medical Center, the Alliance for Paired Donation (APD), the National Kidney Registry (NKR) Program, and the University of Michigan Paired Kidney Exchange Program; internationally, KPD programs exist in the Netherlands (Keizer et al., 2005) and South Korea (Park et al., 1999), among other countries.

In this thesis, we propose the following specific aims.

> Aim 1: To develop optimization methods for selecting exchanges among incompatible pairs in a KPD program that take account of utility and uncertainty, and compare these new methods with those currently in use via microsimulation models based on multiple data sources.
>
> Aim 2: To develop optimization frameworks and approaches for decision making in general KPD programs with altruistic donors.
>
> Aim 3: To examine in greater detail some important aspects arising from the practical management of a KPD program, and propose a general mathematical framework based on Markov decision processes (MDPs) to formulate the KPD optimization problem.

The rest of this dissertation is organized as follows. In Chapter 2, we propose novel organ allocation strategies to arrange kidney exchanges under uncertainties with advantages, including (i) allowance for a general utility-based evaluation of potential kidney transplants and an explicit consideration of stochastic features inherent in a KPD program; and (ii) exploitation of possible alternative exchanges when the originally planned allocation cannot be fully executed. This allocation strategy is implemented using an integer programming (IP) formulation, and its implication is assessed via a data-based simulation system by tracking an evolving KPD program over a series of match runs. Extensive simulation studies are provided to illustrate our proposed approach.

In Chapter 3, we propose a strategy to sequentially allocate the altruistic donor (or bridge donor) so as to maximize the expected utility over a certain given number of moves. Analogous to the way a computer plays chess, the idea is to evaluate different allocations for each altruistic donor (or bridge donor) by looking several moves ahead along a derived look-ahead search tree. Simulation studies are provided to illustrate our proposed method.

Chapter 4 synthesizes and builds upon the work from previous chapters. In this chapter, we examine in greater detail two important aspects in the management of a KPD program — robust organ allocations and operational uncertainties. We first extend the idea of using exchange sets for planning organ allocations; the outcome of this extension is a more robust allocation strategy derived from the concept of strongly connected components (SCCs). We then study more carefully the operational uncertainties inherent in the management of KPD programs. We also develop a general mathematical framework based on Markov decision processes (MDPs), which can be use to rigorously and systematically formulate the problem of managing KPD programs in the presence of altruistic donors. Finally, we conclude with some discussion and future directions of this dissertation work.

CHAPTER 2

Kidney Paired Donation (KPD) Programs with Incompatible Pairs

In this chapter, we propose a novel approach to organizing kidney exchanges among incompatible pairs. This approach is implemented using an integer programming (IP) formulation, and its implication is assessed via a data-based simulation system by tracking an evolving KPD program over a series of match runs.

2.1 Introduction

A fundamental problem in managing KPD programs lies in selecting the "optimal" set of kidney exchanges from among the many possible alternatives. This problem has been modeled and analyzed by economists using a game-theoretic approach (Roth et al., 2004), in which they organized donor-candidate pairs as a "housing market", a concept first proposed by Shapley and Scarf (1974), and produced an efficient organ exchange mechanism using Gale's Top Trading Cycles (TTC) algorithm. Roth et al. (2005) and Segev et al. (2005) applied a maximum cardinality matching algorithm (Edmonds, 1965) to select exchanges that allow the maximum number of transplants, in the case where only two-way exchanges are considered. More general approaches have been developed to undertake such a problem by an integer programming (IP) formulation. This novel IP-based formulation was first proposed by Roth et al. (2007), where each potential transplant was assigned equal weight, resulting in an allocation strategy that enables the greatest number of transplants to be potentially implemented. This IP problem can be efficiently solved by finding a maximum weight perfect matching when no restriction is placed on the cycle length. If, however, only cycles of length up to k are considered, this IP problem is NP-hard when k is larger than two but less than the number of participating pairs (Roth et al., 2007). To address this issue, Abraham et al. (2007) have recently developed an algorithm to reduce the computational complexity of managing large KPD programs.

This chapter extends and improves upon the research described above. Our proposed kidney allocation strategy is innovative in several respects. First, it allows for a quality-oriented evaluation of a potential kidney transplant through medical-outcome-based utilities such as post-transplant graft and recipient survival. In addition, it explicitly takes into consideration the probability that a predicted compatible transplant will result in an actual transplant operation. Third, our approach allows for one or more contingency allocations should the originally planned exchanges fail to be executed. We also suggest a data-based simulation system for studying an evolving KPD program to evaluate different kidney allocation strategies, to compare their impact on performance outcomes, and to assess different choices of utility assignments. The knowledge learned from such simulation studies should provide invaluable guidance in implementing an actual clinical KPD allocation system.

The rest of this chapter is organized as follows: in Section 2.2, we introduce the representation and formulation of a KPD program and present a procedure for arranging kidney exchanges according to the maximum utility cycle-based allocation. In Section 2.3, we explore some important issues and features in a KPD program that have not been addressed by previous studies and further propose a kidney allocation strategy based on the maximum expected-utility set-based allocation. In Section 2.4, we present a data-based simulation system for studying an evolving KPD program. Section 2.5 illustrates the application of the proposed simulation system and reports simulation results for comparing different allocation strategies. We conclude with some discussion in Section 2.6.

2.2 Problem formulation and the maximum utility cycle-based allocation

In this section, we first present a graph representation of a KPD program and then describe two IP formulations for organizing kidney exchanges. We also introduce a procedure for arranging organ exchanges according to the maximum utility cycle-based allocation.

2.2.1 Graph representation

We represent a KPD program as a *directed graph*, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the *vertex set*, $\mathcal{V} \equiv \mathcal{V}(\mathcal{G}) = \{1, 2, \cdots, n\}$, is the set of n incompatible donor-candidate pairs, and the *edge set*, $\mathcal{E} \equiv \mathcal{E}(\mathcal{G})$ is a binary relation on $\mathcal{V}$, consisting of ordered pairs of vertices in $\mathcal{V}$. An edge from i to j, denoted as (i, j), indicates that the donor of pair i is predicted to be compatible with the candidate of pair j. This predicted compatibility is based on a *virtual crossmatch*, which involves computer cross-checking for blood type compatibility as well as comparing candidate antibodies against donor HLA antigens. Before a predicted compatible transplant can be considered for an actual surgical operation, it must be confirmed by a labor intensive *laboratory crossmatch* to assure histocompatibility, which involves incubating the serum of a candidate with the white blood cells of a prospective donor. It is worth noting that prior research on KPD has not taken into consideration this uncertainty, and instead has proceeded as though a negative virtual crossmatch would guarantee, if selected, an actual transplant operation.

In such a directed graph, an *exchange cycle* of length k (or a k-way exchange cycle), $k \geq 2$, is defined as a sequence of distinct vertices in $\mathcal{V}$, denoted as $\langle c_1, c_2, \cdots, c_k \rangle$, such that $(c_k, c_1) \in \mathcal{E}$ and $(c_{j-1}, c_j) \in \mathcal{E}, \forall j = 2, 3, \cdots, k$. For an exchange cycle $C = \langle c_1, c_2, \cdots, c_k \rangle$, we denote its *vertex set* as $\mathcal{V}(C) = \{c_1, c_2, \cdots, c_k\}$ and its *edge set* as $\mathcal{E}(C) = \{(c_k, c_1), (c_{j-1}, c_j), j = 2, 3, \cdots, k\}$. In Figure 2.1 are shown one two-way exchange cycle, $\langle 2, 4 \rangle$, and two three-way exchange cycles, $\langle 1, 2, 3 \rangle$ and $\langle 2, 4, 5 \rangle$. Exchange cycles form a *disjoint* collection if their corresponding vertex sets are disjoint. A *cycle-based allocation* for a KPD program $\mathcal{G}$ is defined as a collection of disjoint exchange cycles, and further denoted as $\mathcal{C}(\mathcal{G})$. An alternative *set-based allocation*, denoted as $\mathcal{S}(\mathcal{G})$, will be introduced and discussed in Section 2.3.4.

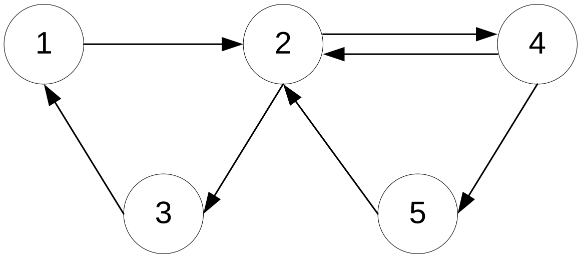

Figure 2.1: Two three-way exchange cycles, $C_1 = \langle 1, 2, 3 \rangle$ and $C_2 = \langle 2, 4, 5 \rangle$, and one two-way exchange cycle, $C_3 = \langle 2, 4 \rangle$; two exchange sets of size three, $S_1 = (\{1, 2, 3\}, \{(1, 2), (2, 3), (3, 1)\})$ and $S_2 = (\{2, 4, 5\}, \{(2, 4), (4, 5), (5, 2), (4, 2)\})$, and one exchange set of size two, $S_3 = (\{2, 4\}, \{(2, 4), (4, 2)\})$.

For an edge $e \equiv (i, j) \in \mathcal{E}$, let u_e or u_{ij} denote an nonnegative *utility* associated with a kidney transplant from the donor in pair i to the candidate in pair j; see more discussion on assigning edge utilities in Section 2.3.1. The utility of an exchange cycle, C, is the sum of its edge utilities, i.e. $U_C = \sum_{e \in \mathcal{E}(C)} u_e$, and the utility of a cycle-based allocation, $\mathcal{C}(\mathcal{G})$, is the sum of the utilities of its cycles, i.e. $\sum_{C \in \mathcal{C}(\mathcal{G})} U_C$. In this setting, the optimal cycle-based allocation which we denote by $\mathcal{C}^*(\mathcal{G})$ is the one with the maximal utility. Roth et al. (2007) proposed two IP formulations to determine $\mathcal{C}^*(\mathcal{G})$. One formulation encodes each exchange cycle as a decision variable, and the other one encodes each edge as a decision variable. We first look at the *cycle formulation* in Section 2.2.2 and then the *edge* *formulation* in Section 2.2.3.

2.2.2 Cycle formulation

Let $\mathcal{C}_k$ be the set of exchange cycles with lengths at most k. For each $i \in \mathcal{V}$, $\mathcal{C}_k(i)$ denotes the exchange cycles in $\mathcal{C}_k$ that involve pair i, i.e. $\mathcal{C}_k(i) = \{C \in \mathcal{C}_k : i \in \mathcal{V}(C)\}$. Define a *decision variable* Y_C for each cycle $C \in \mathcal{C}_k$, such that Y_C is 1 if organ exchanges indicated by C will be arranged, and $Y_C = 0$ otherwise. The problem of selecting $\mathcal{C}^*(\mathcal{G})$ can then be formulated as the following IP problem,

$$\max_{\{Y_C\}} \sum_{C \in \mathcal{C}_k} Y_C U_C, \tag{2.1}$$

$$\text{subject to} \sum_{C \in \mathcal{C}_k(i)} Y_C \leq 1, \forall i \in \mathcal{V}. \tag{2.2}$$

Note that the constraint (2.2) codifies the fact that each pair cannot be allowed in more than one exchange cycle simultaneously. Each feasible solution to this IP problem corresponds to one $\mathcal{C}(\mathcal{G})$, and the optimal solution corresponds to $\mathcal{C}^*(\mathcal{G})$. As a special case in this IP problem, when the utility of an exchange cycle is defined as its length, i.e. $U_C = |\mathcal{V}(C)|$, the resulting objective is to maximize the total number of transplants. Most KPD studies have focused on this simplified utility assignment; see more discussion in Section 2.3.1.

Solving this IP optimization problem in general requires increased computation time as k grows, though in practice most KPD programs restrict k to be three or smaller due to logistical difficulties. Some particular values of k are worth mentioning. When k is equal to two, the optimization problem could be solved in polynomial time using a maximum weighted matching algorithm, which is an extended version of Edmonds' classical maximal cardinality matching algorithm. When k is equal to $|\mathcal{V}|$, i.e. no restriction is placed on the length of an exchange cycle, this problem can be efficiently solved by finding a maximum weight perfect matching. As noted in Roth et al. (2007), this IP problem is NP-hard for k greater than two but less than $|\mathcal{V}|$, which poses associated computational challenges when the number of incompatible pairs is large.

2.2.3 Edge formulation

Let Y_{ij} be a decision variable for each $(i, j) \in \mathcal{E}$, such that Y_{ij} is 1 if edge (i, j) is chosen for a transplant and 0 otherwise. When no restriction is placed on the length of an exchange cycle, we solve the following IP problem:

$$\max_{\{Y_{ij}\}} \sum_{(i,j)\in\mathcal{E}} Y_{ij}u_{ij}, \tag{2.3}$$

$$\text{subject to} \sum_{j:(i,j)\in\mathcal{E}} Y_{ij} \leq 1, \forall i \in \mathcal{V} \tag{2.4}$$

$$\sum_{j:(i,j)\in\mathcal{E}} Y_{ij} = \sum_{j:(j,i)\in\mathcal{E}} Y_{ji}, \forall i \in \mathcal{V}. \tag{2.5}$$

This IP problem could be solved efficiently in polynomial time by finding a maximum weight perfect matching. If the exchange cycle length is restricted to be at most k, an additional set of constraints has to be added, i.e.

$$\sum_{e\in\mathcal{E}(C)} Y_e \leq k, \forall C \in \mathcal{C}_k. \tag{2.6}$$

The number of additional constraints in (2.6), even when $k = 3$, is usually enormously large in a realistic KPD program with several hundred incompatible pairs, which makes it impossible to even store all the constraints in a typical IP solver such as CPLEX [1] or Gurobi [2]. Therefore, the cycle formulation is usually preferred.

2.2.4 The maximum utility cycle-based allocation

We summarize below a procedure for arranging kidney exchanges according to the maximum utility cycle-based allocation. Given a KPD program $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

(i) Define $u : \mathcal{E} \to \mathbb{R}^+$, where u_e is the utility assigned to an edge $e \in \mathcal{E}$.

(ii) Enumerate $\mathcal{C}_k$ and find U_C for each $C \in \mathcal{C}_k$.

(iii) Determine $\mathcal{C}^*(\mathcal{G})$ by solving an IP problem.

(iv) Arrange kidney exchanges according to $\mathcal{C}^*(\mathcal{G})$.

In (ii), we enumerate $\mathcal{C}_k$ by developing an algorithm based on depth-first search (DFS), which essentially prioritizes the direction of search always to offspring vertices and then to sibling vertices.

[1] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[2] http://www.gurobi.com/

2.3 Optimal kidney allocation in a KPD program

In Section 2.2, we have introduced a graph representation of a KPD program and presented an allocation strategy based on $\mathcal{C}^*(\mathcal{G})$. In this section, we focus on issues in managing KPD programs that have not been previously addressed and further explore two alternative organ allocation strategies.

2.3.1 Utilities and uncertainties in a KPD program

Much of the prior work has focused on a simplified edge utility, namely $u_e = 1, \forall e \in \mathcal{E}$. As a result, the utility U_C of an exchange cycle, C, equals its length, and the objective function in the IP problem is the total number of arrangeable transplants. Some recent developments, however, have emerged in assigning rule-based utilities determined by pair attributes, such as degree of candidate sensitization or the age difference between the candidate and donor, so that they better evaluate potential transplants. For example, in an operational guideline recently posted by the U.S. national KPD pilot program, each potential transplant is initially assigned a base utility of 200 points. Extra points are added as bonuses to edges that, say, have zero antigen mismatches, or that involve a donor and a candidate in the same transplant center; on the other hand, a certain number of points are deducted, for example, when a donor has one or more of the candidate's other antibody specificities.

For potential transplants in a KPD program, it is also desirable to examine outcome-based utilities derived from medical outcomes such as the expected graft or patient survival. To this end, we propose to associate with edge, (i, j), a utility quantifying the medical outcome from a potential kidney transplant involving the donor in pair i and the candidate in pair j. This outcome could be, for example, graft survival, post-transplant recipient survival, or the incremental years of recipient life that would accrue with a kidney transplant as opposed to remaining on dialysis. By incorporating this outcome-based utility, we are able to evaluate and compare competing kidney allocations with a quality-oriented view, and therefore provide kidney transplant candidates with organs that are not only compatible, but that could potentially maximize opportunities to achieve good quality of life after transplants.

Prior research on KPD has implicitly assumed that a predicted compatible transplant, if attempted, would yield an actual transplant operation. In reality, predicted compatible transplants have to be confirmed by laboratory crossmatches, and hence may or may not lead to actual transplant operations. This uncertainty is a necessary ingredient since laboratory crossmatches cannot be undertaken on all possibly compatible donors and candidates due to labor and resource limitations. Further, even if the laboratory crossmatch is negative (non-reactive), an actual transplant operation may not occur due to other friction including, for example, pair preference, illness, and death. Throughout the rest of the dissertation, we use the term "*is viable*" to indicate that the edge actually results, if chosen, in a completed transplant operation. An exchange cycle is viable if each of its edges is viable.

Ignoring this uncertainty can result in a situation in which long exchange cycles are evaluated more favorably than short ones, despite the fact that longer cycles are much less likely to be implemented. To partially incorporate this uncertainty into the arrangement of kidney exchanges, the operational guidelines for the national KPD pilot program proposed a deduction of 30 points for a three-way exchange cycle, but not for an exchange cycle that is two-way.

To further address this stochastic feature inherent in a KPD program, we associate with each edge a probability corresponding to the chance of that edge being viable. This probability depends on factors such as the number of mismatched antigens, the donor-candidate travel distance, and age difference. For each $e \in \mathcal{E}$, let X_e be a Bernoulli random variable with $X_e = 1$ if e is viable, and $X_e = 0$ otherwise. By defining p_e as $P(X_e = 1)$, an exchange cycle, C, is then viable with probability equal to $P_C \equiv \prod_{e \in \mathcal{E}(C)} p_e$, under the assumption that edges in an exchange cycle have an independence relationship. More formally, this assumption is regarded as having a collection of independent random variables, $\{X_e, e \in \mathcal{E}(C)\}$. Though this independence relationship can be further relaxed, it is a reasonable assumption when the rate of pair withdrawal (due to factors such as pregnancy, illness, or death) is not high. In the rest of this chapter, we proceed by assuming in general that $X_e, e \in \mathcal{E}$ are independent. It is worth pointing out that a set of common factors could affect both the edge probability and utility.

As we should discuss later in Section 2.3.3, incorporating these probabilities into the management of a KPD program also opens up the opportunity to identify and implement possible alternative fall-back options if the originally planned exchanges cannot be fully executed.

In this chapter, we focus on the development of an allocation strategy for the management of a KPD program with certain desired properties that will be introduced in the remaining of this section. Our attention, however, is not on the estimation of utilities and probabilities associated with potential transplants, although Section 2.3.5 presents a discussion of some aspects of this.

2.3.2 The maximum expected-utility cycle-based allocation

One natural way to evaluate an exchange cycle is to use its *expected utility*, namely $EU_C \equiv U_C P_C$. Utilizing expected utilities in the management of a KPD program provides a way to incorporate uncertainty into the selection of kidney exchanges. For example, expected utility recognizes that a longer exchange cycle has a smaller chance of being viable, which counterbalances the fact that such a cycle might potentially contribute greater utilities and allow for more transplants.

The expected utility of a cycle-based allocation, $\mathcal{C}(\mathcal{G})$, is the sum of the expected utilities of its exchange cycles, i.e. $\sum_{C \in \mathcal{C}(\mathcal{G})} EU_C$. Among all cycle-based allocations, the one with the largest expected utility is the *maximum expected-utility cycle-based* allocation, which we denote as $\overline{\mathcal{C}}^*(\mathcal{G})$. The following procedure generates $\overline{\mathcal{C}}^*(\mathcal{G})$ in a KPD program $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and arranges exchanges accordingly.

(i) Define $u : \mathcal{E} \to \mathbb{R}^+$, where u_e is the utility of an edge $e \in \mathcal{E}$, and $p : \mathcal{E} \to [0, 1]$, where p_e is the probability that e is viable.

(ii) Enumerate $\mathcal{C}_k$ and calculate EU_C for each $C \in \mathcal{C}_k$.

(iii) Find $\overline{\mathcal{C}}^*(\mathcal{G})$ by an IP-based approach as in Section 2.2, with U_C replaced by EU_C.

(iv) Arrange kidney exchanges according to $\overline{\mathcal{C}}^*(\mathcal{G})$.

In (iii), the cycle formulation generalizes in a straightforward manner, but the edge formulation cannot be extended; this is because the edge formulation represents each edge as a separate decision variable and hence cannot describe EU_C. Notice that the above procedure and the one in Section 2.2.4 are both *fixed* in that they do not specify how to proceed if a chosen exchange cycle is not viable.

2.3.3 Contingency plans

Let us begin with a motivating example. In a small KPD program as represented in Figure 2.1, there are two three-way exchange cycles, $C_1 = \langle 1, 2, 3 \rangle$ and $C_2 = \langle 2, 4, 5 \rangle$, and one two-way exchange cycle, $C_3 = \langle 2, 4 \rangle$. We assume that $EU_{C_1} > EU_{C_2} > EU_{C_3}$, and thus conclude $\overline{\mathcal{C}}^*(\mathcal{G})$ is $\{C_1\}$. A further examination, however, reveals that pairs in C_3 are part of the pairs in C_2, i.e. $\mathcal{V}(C_3) = \{2, 4\} \subset \mathcal{V}(C_2) = \{2, 4, 5\}$. This observation might suggest that, when C_2 is selected but could not be completed because of problems in either $(4, 5)$ or $(5, 2)$, the two-way exchange cycle C_3 could still be selected. Therefore, the contribution from this back-up exchange cycle C_3 would add some extra value to the exchange cycle C_2. Does this extra value make $\{C_2\}$ a preferred allocation to $\{C_1\}$? How should we evaluate $\{C_2\}$ so as to correctly recognize $\{C_3\}$ as a possible back-up allocation? To address these questions in this specific example and address other related issues in general, we propose the following.

First, we give two definitions from graph theory: (i) a graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a *subgraph* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if $\mathcal{V}' \subset \mathcal{V}$ and $\mathcal{E}' \subset \mathcal{E}$, and (ii) a graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is an *induced subgraph* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if $\mathcal{G}'$ is a subgraph of $\mathcal{G}$, and in addition, $\mathcal{E}' = \{(u, v) \in \mathcal{E} : u, v \in \mathcal{V}'\}$. In the context of a KPD program, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we then define an *exchange set* as an induced subgraph of $\mathcal{G}$, denoted by $S = (\mathcal{V}', \mathcal{E}')$, with the requirement that it admits at least one exchange cycle of length $|\mathcal{V}'|$, where $|\mathcal{V}'|$ is defined as the *size* of this exchange set. Figure 2.1 illustrates two exchange sets of size three, S_1 and S_2, and one exchange set of size two, S_3.

By definition, an exchange set would allow for one or more cycle-based allocations; if there is indeed more than one and not all of the exchange cycles in the first attempted allocation are viable, we might still have the option to select another allocation, which is called a *contingency*. Therefore, the expected utility generated from arranging exchanges in an exchange set depends on the order in which the possible cycle-based allocations are utilized. Such an ordering defines a more general procedure, which includes the two previously introduced procedures as special cases. For example, in an exchange set, $S = (\mathcal{V}', \mathcal{E}')$, if $\mathcal{C}^*(S)$ is adopted as top priority with no allocation assigned as a contingency, this in fact corresponds to the procedure presented in Section 2.2.4, and in consequence, the corresponding expected utility is $\sum_{C \in \mathcal{C}^*(S)} EU_C$. Similarly, the procedure described in Section 2.3.2 corresponds to selecting $\overline{\mathcal{C}}^*(S)$ as a first priority but again with no contingency at all, and hence generates an expected utility of $\sum_{C \in \overline{\mathcal{C}}^*(S)} EU_C$. According to what we have discussed, both of these two procedures are fixed in the sense that they each select an allocation only for $S = (\mathcal{V}', \mathcal{E}')$, but do not specify how to proceed on $(\mathcal{V}', \mathcal{E}' \setminus \mathcal{E}_f)$, where $\mathcal{E}_f \equiv \{e \in \mathcal{E}' : X_e = 0\}$ is observed when executing that chosen allocation. In contrast, the following "greedy" procedure is *sequential* and generates the largest expected utility.

(i) Find $\mathcal{C}^*(S)$ in $S = (\mathcal{V}', \mathcal{E}')$ by applying the procedure presented in Section 2.2.4.

(ii) If all cycles in $\mathcal{C}^*(S)$ are viable, finish with a claimed utility of $\sum_{C \in \mathcal{C}^*(S)} U_C$; if, however, certain edges, which we denote as $\mathcal{E}_f$, fail to be carried forward to actual operations, abort the original allocation and go back to (i) with $\mathcal{E}' \leftarrow \mathcal{E}' \setminus \mathcal{E}_f$.

Return to Figure 2.1, where both the three-way exchange cycle, $C_2 = \langle 2, 4, 5 \rangle$, and the two-way exchange cycle, $C_3 = \langle 2, 4 \rangle$, could be selected in the exchange set S_2. If U_{C_2} is larger than U_{C_3}, C_2 should be chosen as the first priority. When edge $(4, 5)$ and/or edge

$(5, 2)$ in C_2 are not viable, C_3 could be tried as a contingency plan. On the other hand, if U_{C_2} is less than U_{C_3}, then the cycle C_3, backed up by C_2, should be selected as the first priority.

In light of this and similar examples, we quantify the potential value of selecting an exchange set S by applying the above greedy procedure for S. This potential value actually defines EU_S, the *expected utility* of S. Such an approach takes into consideration the full-potential contributions from all possible back-up allocations. Note that the expected utility of an exchange set can be exactly formulated as follows. For an exchange set, $S = (\mathcal{V}', \mathcal{E}')$, let $2^{\mathcal{E}'}$ denote the collection of all subsets of $\mathcal{E}'$, and for each $\tilde{\mathcal{E}} \in 2^{\mathcal{E}'}$, let

$$P(\tilde{\mathcal{E}}) \equiv \prod_{e \in \tilde{\mathcal{E}}} p_e \prod_{e \in (\mathcal{E}' \setminus \tilde{\mathcal{E}})} (1 - p_e), \tag{2.7}$$

which is the probability that the edges in $\tilde{\mathcal{E}}$ are viable whereas those in $\mathcal{E}' \setminus \tilde{\mathcal{E}}$ are not. Consider $\tilde{S} \equiv (\mathcal{V}', \tilde{\mathcal{E}})$, and let

$$U(\tilde{\mathcal{E}}) \equiv \sum_{C \in \mathcal{C}^*(\tilde{S})} U_C. \tag{2.8}$$

It follows that

$$EU_S = \sum_{\tilde{\mathcal{E}} \in 2^{\mathcal{E}'}} U(\tilde{\mathcal{E}}) P(\tilde{\mathcal{E}}). \tag{2.9}$$

In Figure 2.1, S_2 would be preferred to S_1 if $EU_{S_2} > EU_{S_1}$.

2.3.4 The maximum expected-utility set-based allocation

We have introduced a cycle-based allocation as a collection of disjoint exchange cycles. Among all such allocations, the procedure in Section 2.2.4 arranges kidney exchanges according to the allocation with the maximum utility, while the procedure in Section 2.3.2 arranges kidney exchanges according to the one with the maximum expected utility.

In this subsection, we define a set-based allocation as a collection of disjoint exchange sets, and further denote it as $\mathcal{S}(\mathcal{G})$ for a KPD program $\mathcal{G}$. Following the way in which the expected utility of an exchange set is defined in Section 2.3.3, the expected utility of a set-based allocation $\mathcal{S}(\mathcal{G})$ is, therefore, $\sum_{S \in \mathcal{S}(\mathcal{G})} EU_S$. Among all set-based allocations, the following procedure arranges kidney exchanges according to the maximum expected-utility set-based allocation, which we denote as $\overline{\mathcal{S}}^*(\mathcal{G})$.

(i) Define $u : \mathcal{E} \to \mathbb{R}^+$ and $p : \mathcal{E} \to [0, 1]$ as in Section 2.3.2.

(ii) Enumerate $\mathcal{S}_k$, the set of all exchange sets of size at most k, where $2 \leq k \leq |\mathcal{V}|$.

(iii) For each $S \in \mathcal{S}_k$, calculate its expected utility as $EU_S = \sum_{\tilde{\mathcal{E}} \in 2^{\mathcal{E}'}} U(\tilde{\mathcal{E}})P(\tilde{\mathcal{E}})$.

(iv) Select $\overline{\mathcal{S}}^*(\mathcal{G})$ by forming an IP problem similar to the one discussed in Section 2.2.2.

(v) Apply the aforementioned greedy procedure to each $S \in \overline{\mathcal{S}}^*(\mathcal{G})$.

Arranging kidney exchanges according to the above procedure allows for a more flexible utility-based evaluation of potential transplants, takes into consideration the uncertainties inherent in a KPD program, and provides fall-back options when possible. Several remarks follow. In (ii), a reasonable k, say three or four, is required in practice due to logistical concerns as in Section 4.1. Enumerating $\mathcal{S}_k$ could be accomplished by a DFS-based algorithm similar to the one presented in Section 2.2.4. In (iii), calculating EU_S involves a summation over $|2^{\mathcal{E}'}|$ terms, which poses no computational difficulties in practice for small k.

2.3.5 Estimation of utilities and probabilities

So far in this chapter, we have assumed that we are given a utility function, $u : \mathcal{E} \to \mathbb{R}^+$, and a probability function, $p : \mathcal{E} \to [0, 1]$. In practice, however, these utilities and probabilities are not available and have to be estimated.

In the literature, modeling of outcome-based utility has been considered in deceased-donor kidney transplants by Wolfe et al. (2008), Schaubel et al. (2006), and Schaubel et al. (2009). Such models could be adopted relatively easily to living-donor kidney transplants. On the other hand, the model for probability can be established via a logistic modeling approach, based on clinical data from the Scientific Registry of Transplantation Recipients (SRTR), and multiple KPD programs including the Alliance for Paired Donation (APD) and the University of Michigan Transplant Center. In this logistic model, some primary predictors include, for example, the percentage of current panel reactive antibody (PRA) and the cross reactivity of antibody specificities.

Thus, in practice, all three procedures previously discussed for selecting the cycle-based or set-based allocations can be easily adopted by replacing the u and p with their estimated versions.

2.3.6 A KPD match run and an evolving KPD program

We consider a *match run* as a series of operations on a collection of incompatible pairs $\mathcal{V}$:

(i) Form $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is determined by checking virtual crossmatches on $\mathcal{V}$.

(ii) Assign for each $e \in \mathcal{E}$ an estimated utility $\hat{u}_e$ and an estimated probability $\hat{p}_e$ according to the models discussed in Section 2.3.5.

(iii) Arrange kidney exchanges according to $\mathcal{C}^*(\mathcal{G})$, $\overline{\mathcal{C}}^*(\mathcal{G})$, or $\overline{\mathcal{S}}^*(\mathcal{G})$.

(iv) Recycle any donor-candidate pair that does not proceed to an actual transplant operation back to the KPD pool awaiting future matches.

Every KPD program is constantly evolving in that successfully transplanted pairs leave and new incompatible pairs arrive over time. In addition, existing pairs in the pool could withdraw due to circumstances such as donor or candidate pregnancy, illness, or death. Such an evolving KPD program is managed by repeatedly executing match runs and updating the KPD pool on a regular basis over time. See Figure 2.2 for an illustration.

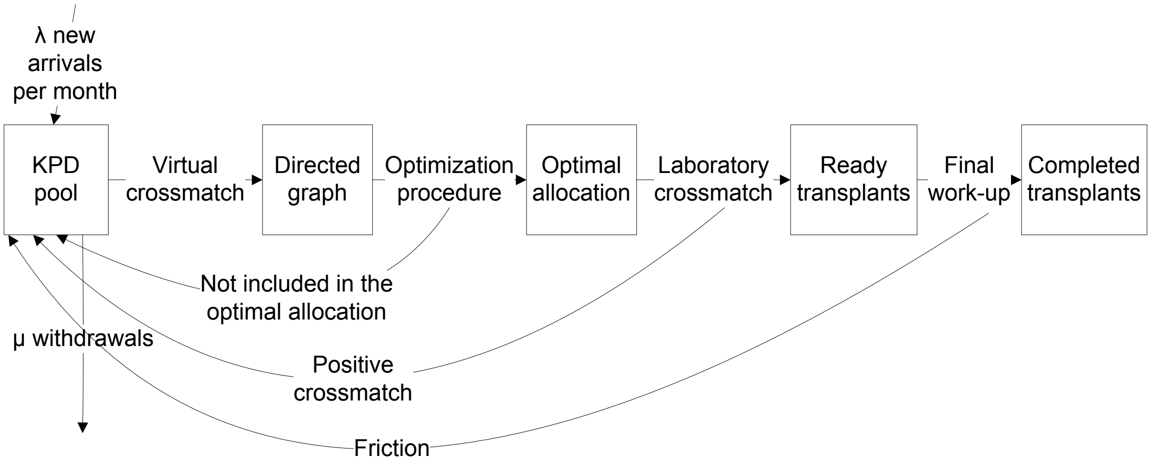

Figure 2.2: A flow diagram of an evolving KPD program

2.4 KPD simulation system

In Sections 2.2 and 2.3, we have presented three organ allocations for a KPD program $\mathcal{G}$, namely $\mathcal{C}^*(\mathcal{G})$, $\overline{\mathcal{C}}^*(\mathcal{G})$, and $\overline{\mathcal{S}}^*(\mathcal{G})$. A comparison among them and other potential ones is certainly of great interest with respect to the practical management of a KPD program. Such a comparison, however, cannot be accomplished by traditional clinical trials due to the nature of kidney transplantation or any transplantation for that matter. Therefore, simulation plays an important role for achieving this purpose. In this section, key ingredients of such a simulation system will be discussed.

2.4.1 Generating incompatible pairs

To create an incompatible pair, we generate its donor and candidate separately according to their own population distributions. Candidates are *sampled at random with replacement* from databases of candidates presenting with a willing but incompatible donor. One such database is derived from the University of Michigan KPD program, in which candidates are described by variables of blood type, PRA, intensity of candidate antibody specificities, and so on. Currently, the UM database consists of 187 incompatible pairs, and additional databases from other KPD programs (as they become available to us through data-usage agreements) will be incorporated for wider variation in candidates. Donors, on the other hand, are generated by separately sampling their blood types and HLA haplotypes. Precisely, blood types are drawn from the U.S. population distribution: O, 44%; A, 42%; B, 10%; and AB, 4% (Stanford Blood Center, 2010). HLA haplotypes are sampled according to their frequencies in the U.S. population, which is derived from an extensive public database on potential bone marrow donors (Maiers et al., 2007).

A simulated donor-candidate pair is regarded as an incompatible pair and hence included in the KPD pool if either their ABO blood types are mismatched, or the donor's HLA haplotypes contain any of the candidate's antibody specificities (with mean fluorescence intensity $> 5,000$), or both. In this chapter, we do not consider KPD programs that include compatible pairs.

2.4.2 Simulating a match run and an evolving KPD program

We simulate a KPD match run on a collection of incompatible pairs, $\mathcal{V}$, by simulating each of its four steps, as discussed in Section 2.3.6.

(i) Determine $\mathcal{E}$ by checking the simulated blood types and HLA haplotypes.

(ii) Estimate both u_e and p_e based on generated pair characteristics.

(iii) Apply $\mathcal{C}^*(\mathcal{G})$, $\overline{\mathcal{C}}^*(\mathcal{G})$, or $\overline{\mathcal{S}}^*(\mathcal{G})$ to arrange kidney exchanges.

(iv) Since a potential transplant may not be viable, we simulate such uncertainty via a Bernoulli trial with the probability of success equal to that edge probability; the realization of such a Bernoulli trial will indicate if a pair proceeds to an actual transplant and hence leaves the pool, or remains in the pool and awaits future matches.

To address the feature that a KPD program is evolving over time as discussed in Section 2.3.6, we first generate an initial KPD pool of N incompatible pairs as described in

Section 2.4.1. Further, we assume that the arrival of new incompatible pairs follows a Poisson process with a rate λ. This rate may be governed by a log-linear model with covariates of age, candidate illness, and the relationship between donor and candidate, among others. Also, we assume that the withdrawal of existing pairs from the KPD pool follows another Poisson process with a rate μ; and the log-linear model for μ could include other relevant covariates.

2.5 Simulation results

The proposed simulation system in Section 2.4 enables us to investigate various aspects associated with the management of a KPD program, among which comparing different allocation strategies is certainly of special interest. Simulation results including such comparisons are presented in this section. We implement the simulation system using the C++ programming language, and the related IP problems are solved by Gurobi Optimizer.

2.5.1 Problem complexity

The number of exchange cycles or sets can be enormous in a reasonably large KPD pool of, say, several hundred pairs, even though the size of exchange cycles or sets is restricted to at most three. Such complexity causes solving related IP problems computationally rather expensive. Table 2.1 summarizes the averaged numbers (over 200 rounds of simulations) of exchange cycles and sets of up to three pairs in pools of various sizes, 100, 200, 300, 400, and 500, where incompatible pairs are generated according to Section 2.4.1.

Table 2.1: The averaged numbers of exchange cycles or exchange sets up to three pairs in KPD pools of various sizes; standard deviations are given in the parentheses; the summary is calculated over 200 rounds of simulations.

	exchange cycles (length 2 & 3)	exchange sets (size 2 & 3)
pool size	mean (standard deviation)	mean (standard deviation)
100	388 (237)	383 (229)
200	2,659 (998)	2,630 (977)
300	9,413 (3,164)	9,305 (3,100)
400	21,076 (6,140)	20,829 (5,992)
500	40,290 (9,337)	39,815 (9,120)

2.5.2 Simulation setup

We perform a total of 600 simulations, in which $\{p_e, e \in \mathcal{E}\}$ is generated from a uniform distribution, $U(0.1, 0.5)$; this distribution gives, for a virtual-crossmatch compatible transplant, an average successful rate of 30%, a reasonable number according to our experience at the University of Michigan and the Alliance for Paired Donation. In order to obtain a set of utilities for illustrative purpose, we generated utilities from two uniform distributions, $U(10, 20)$ and $U(10, 30)$; these are used to generate $\{u_e, e \in \mathcal{E}\}$. Each distribution of utilities is used for 200 simulations; edge utilities are fixed at 1 for the remaining 200 simulations. Notice that these simulation parameters are chosen in such ways to illustrate the application of our proposed simulation system; in practice, these utilities and probabilities should be estimated based on real data as we have discussed in Section 2.3.5.

We start each round of simulation by generating an initial pool of $N = 200$ incompatible pairs. Additionally generated pairs then enter the pool according to a Poisson process with $\lambda = 10$ pairs per month over a period of $m = 24$ months. For simplicity, we assume no existing pairs drop out of the pool.

A match run is executed at the end of each month on this evolving pool, starting with the initial 200 pairs. Pairs that arrive during the time of a match run will not participate in the current run, but wait for the next one. For each simulation, we make three copies of the evolving KPD program and execute each of the three match runs, which arrange kidney exchanges according to $\mathcal{C}^*(\mathcal{G})$, $\overline{\mathcal{C}}^*(\mathcal{G})$, and $\overline{\mathcal{S}}^*(\mathcal{G})$), on each copy such that we can directly compare these allocation strategies. At the end of each match run, the KPD pool is updated with some pairs leaving or staying. We record several important measures needed for comparison, such as cumulative claimed utilities, cumulative number of transplants, and blood types of candidates receiving transplants.

The above description provides a simple simulation setup that enables us to examine and compare different allocation strategies. More realistic models will be further explored by incorporating more comprehensive data on pair characteristics and actual transplant operations, when such data become available to us via data-usage agreements.

2.5.3 Results

First, we report on the cumulative number of transplants over a period of 24 months across three allocation strategies, $\mathcal{C}^*(\mathcal{G})$, $\overline{\mathcal{C}}^*(\mathcal{G})$, and $\overline{\mathcal{S}}^*(\mathcal{G})$, and under three different models of utilities, $U(1, 1)$, $U(10, 20)$, and $U(10, 30)$; see Figure 2.3, which unveils a consistent pattern regardless of utility models that $\overline{\mathcal{S}}^*(\mathcal{G})$ results in the greatest number of transplants, whereas $\mathcal{C}^*(\mathcal{G})$ leads to the fewest number of transplants. Take Figure 2.3-C as an example. When the edge utility is fixed at 1, allocation strategy $\overline{\mathcal{S}}^*(\mathcal{G})$ results in a median of 50 (over 200 simulations) transplants after match run 10 (at the end of month 9); in contrast, for exactly the same evolving KPD program, strategy $\mathcal{C}^*(\mathcal{G})$ only leads to a median of 34 transplants over the same period of time. Results from simulation also show that allocation strategy $\overline{\mathcal{C}}^*(\mathcal{G})$ allows for a much higher number of transplants than $\mathcal{C}^*(\mathcal{G})$ does, though it performs worse than $\overline{\mathcal{S}}^*(\mathcal{G})$.

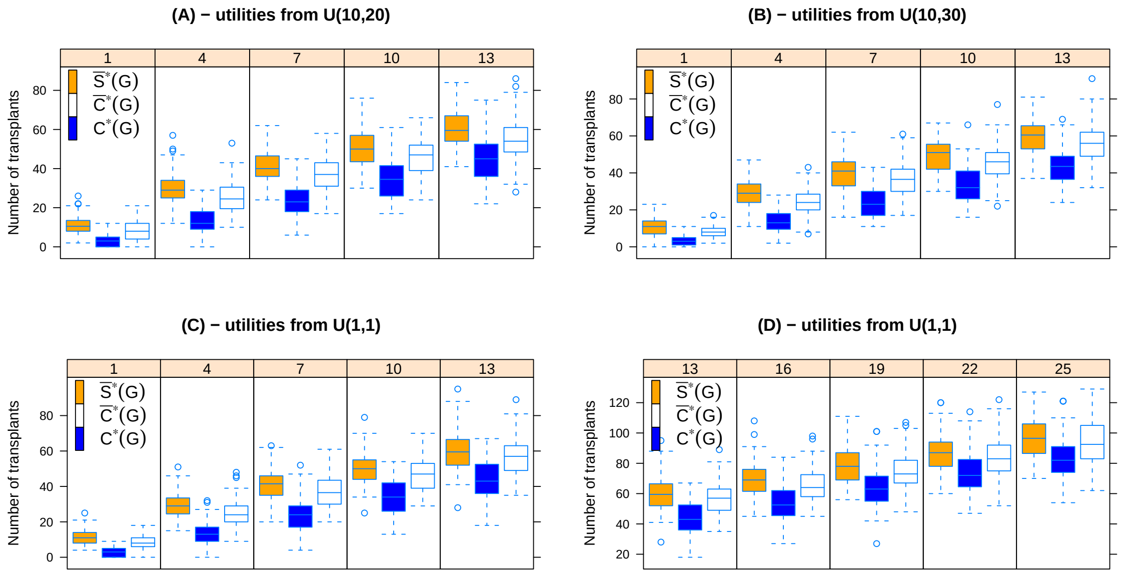

Figure 2.3: Cumulative number of transplants over 12 or 24 months across three allocation strategies, under three utility generating mechanisms (results are based on 200 simulations). Edge utilities are generated from $U(1,1)$, $U(10,20)$, and $U(10,30)$; and edge probabilities are generated from $U(0.1,0.5)$. The numbers at the top of each panel indicate match runs.

Figure 2.4 demonstrates that $\overline{\mathcal{S}}^*(\mathcal{G})$ is advantageous over both $\mathcal{C}^*(\mathcal{G})$ and $\overline{\mathcal{C}}^*(\mathcal{G})$ in that $\overline{\mathcal{S}}^*(\mathcal{G})$ on average achieves the largest cumulative claimed utility. Notice that when edge utilities are fixed at 1, the cumulative claimed utility is the same as the cumulative number of transplants; therefore, we only compare these allocation strategies under two other utility models, i.e. $U(10,20)$ and $U(10,30)$.

We can also examine via the proposed simulation system other aspects of a KPD program. For example, we are interested in exploring how the chance of having a transplant is associated with blood types. In practice, candidates with blood type O are usually in a disadvantageous position due to the limitation that they can only receive kidneys from blood type O donors, who, however, can donate to candidates of any blood type. This phenomenon is clearly observed in Figure 2.5, where about 60% of the incoming candidates are of blood type O, while only about 40% of the performed transplants involve a blood O

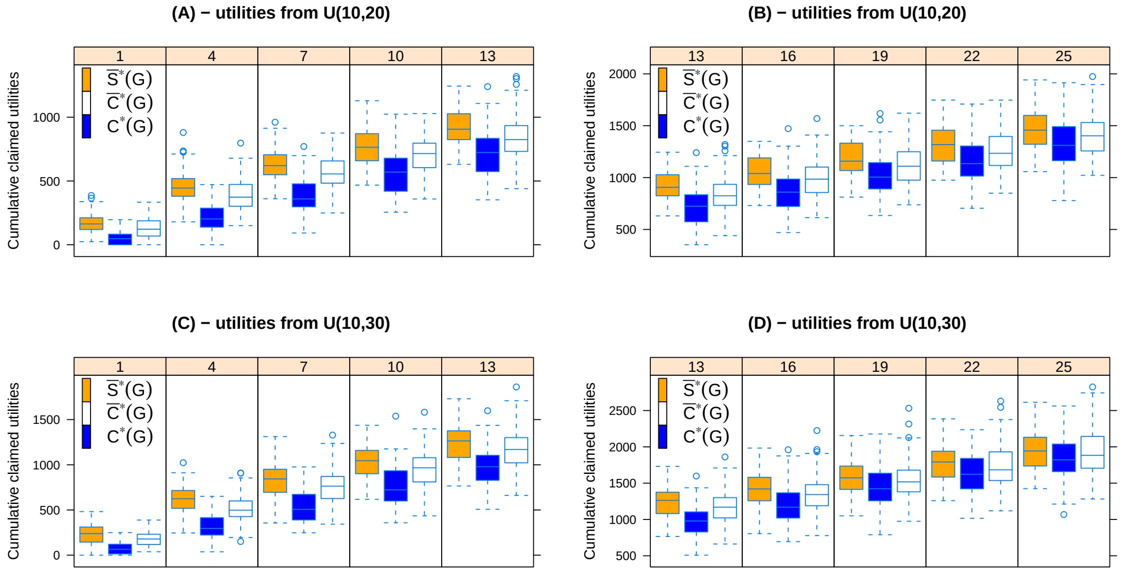

Figure 2.4: Cumulative claimed utilities of transplants over 24 months across three allocation strategies, under two utility generating mechanisms (results are based on 200 simulations). Edge utilities are generated from $U(10, 20)$, and $U(10, 30)$; and edge probabilities are generated from $U(0.1, 0.5)$. The numbers at the top of each panel indicates match runs.

candidate. As a consequence, candidates of the other blood types (A, B, and AB) are more represented among all candidates receiving a transplant. One possible solution to this difficulty is to assign a bonus utility to a potential transplant that involves an O donor giving to an O candidate, or prohibit the option of an O donor being allocated to a non-O candidate.

2.6 Concluding remarks

In this chapter, we have proposed a novel approach to organizing kidney exchanges in an evolving KPD program. Our approach identifies the maximum expected-utility set-based allocation that not only allows for a more general outcome-based evaluation of kidney transplants, but also takes into consideration stochastic features in managing a KPD program. Further, this approach exploits possible back-up exchanges when the originally planned allocation cannot be fully executed. Being able to provide such back-up exchanges is even more valuable considering that in practice utilities that relate to expected outcomes could usually decrease over time; it is generally better for a transplant to be performed sooner than later.

We have also suggested a data-based simulation system that enables us to examine various aspects of an evolving KPD program. This simulation system allows us to emulate

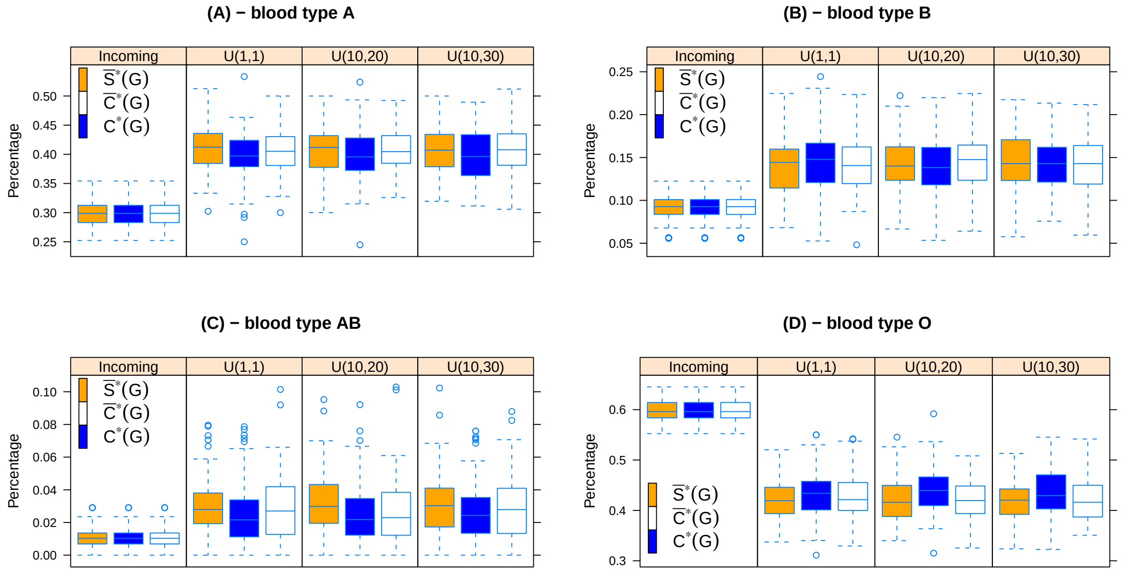

Figure 2.5: Summaries of the percentage of candidates with certain blood type among all candidates joining the KPD pool and that percentage among all candidates who received a transplant (results are based on 200 simulations). Similar pattern are observed under three utility generating mechanisms, i.e. $U(1, 1)$, $U(10, 20)$, and $U(10, 30)$.

genetic and demographic data from existing KPD programs, and to derive statistical models that mimic actual KPD programs. In particular, we have suggested (i) models for donor and candidate characteristics as well as their arrival in and withdrawal from a KPD pool; (ii) models for the estimation of the outcome-based utility of a potential transplant; and (iii) models for the prediction of the probability that a planned transplant would indeed occur. Utilizing such a simulation system, we are able to quantitatively compare different kidney allocation strategies; and results shed light on decision support applicable to actual KPD programs.

We have illustrated the proposed simulation system to compare several kidney allocation strategies. Results from simulation studies suggest advantages of adopting the maximum expected-utility set-based allocation over the other two allocation strategies. Such advantages are attributive to the introduction of an exchange sets, in which uncertainties in a KPD system are properly incorporated and possible contingency allocations are allowed should the planned exchanges fail to be executed. In future work, we plan to base our simulation on more realistic models that will be developed by incorporating additional KPD program source data.

Another possible future investigation is to consider exchanges initiated by an altruistic donor (Rees et al., 2009). An altruistic donor does not arrive with a designated candidate, but instead volunteers to donate a kidney to a pool of waiting candidates. Figure 2.6 gives an illustration of a chain of transplants initiated by an altruistic donor. Since transplants along the chain are not required to be performed simultaneously, a bridge donor (namely the donor whose incompatible candidate received a kidney but has yet to donate) at the end of the chain could later donate to a candidate who will arrive in the future, whose willing but incompatible donor then becomes a new bridge donor. In this respect, the chain is open-ended and increases the chance for a highly sensitized candidate to receive a compatible kidney. The proposed work could be easily extended to include altruistic donors as participants in a KPD program. More specifically, a chain of kidney transplants can be viewed as a special exchange cycle, in which the bridge donor "donates" to a phantom candidate associated with the altruistic donor who first initiated that chain, and hence forms a hypothetical "edge" (denoted by dashed arrowed lines in Figure 2.6). Future research on KPD with altruistic donors is certainly of high importance and great interest; we will report those results in a future publication.

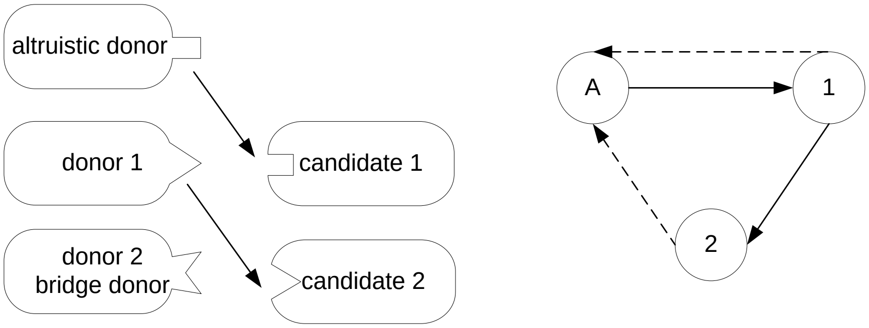

Figure 2.6: A chain of transplants initiated by an altruistic donor. The dashed arrow indicates a hypothetical "edge", which represents a "donation" to a phantom candidate associated with the altruistic donor.

CHAPTER 3

Decision Making in KPD Programs with Additional Altruistic Donors

In this chapter, we focus on the management of KPD programs with altruistic donors. We propose a strategy to sequentially allocate the altruistic donor (or bridge donor) so as to maximize the expected utility over a certain given number of moves.

3.1 Introduction

In recent years, KPD has also been extended to include living non-directed donors (LND-s), or altruistic donors; these are donors who have no designated candidate and decide to donate voluntarily. In this context, an altruistic donor may donate to the candidate of an incompatible pair with the understanding that the donor of that pair will become a *bridge donor*, and further donate to the candidate of a second pair, and so on; such a process continues and thus forms an LND-initiated chain. One advantage to such chains as compared to two-way or higher order exchange cycles is that transplants along the chain do not need to be performed simultaneously (Montgomery et al., 2006; Roth et al., 2006). As a consequence, the donor whose incompatible candidate received another donor's kidney but has yet to donate could later donate to a candidate who will arrive in the future, hence the name "bridge donor". For this reason, this LND-initiated chain is sometimes called a non-simultaneous extended altruistic donor (NEAD) chain (Rees et al., 2009). Kidneys from altruistic donors used to be designated to patients with no living donors and who therefore are placed on a deceased-donor waiting list. A NEAD chain, however, allows for passing the altruism beyond saving just one patient, to an extent that would potentially benefit a greater number of patients. Such an advantage has already been demonstrated via simulation studies by Gentry et al. (2009) and Ashlagi et al. (2011).

In this chapter, we consider a strategy for developing a NEAD chain under uncertainties in a KPD program with one altruistic donor. We also discuss in general some possible extensions of this strategy in order to incorporate multiple altruistic donors. Analogous to the way a computer plays chess, we propose an approach to sequentially allocate the altruistic donor (or bridge donor) so as to maximize the expected utility over a certain given number of moves. The idea is to evaluate different allocation options available for each altruistic donor (or bridge donor) by looking several moves ahead along a derived look-ahead search tree, and then proceed with the allocation evaluated most favorably.

The rest of the chapter is organized as follows: in Section 3.2, we introduce a graph representation for KPD programs with altruistic donors. With this representation, we define the *optimal policy* in the context of managing a KPD program with one altruistic donor. This optimal policy can be obtained in general by following a standard decision tree analysis, which we briefly illustrate in Section 3.3. The computation associated with this decision tree-based approach, however, is very expensive for large KPD programs. To address this issue, we propose, in Section 3.4, a more efficient and practical approach which sequentially extends a NEAD chain according to the calculation performed along a look-ahead search tree. Section 3.5 provides simulation studies to illustrate our proposed strategy and reports some simulation results. In Section 3.6, we conclude with some discussion on possible extensions to incorporate multiple altruistic donors.

3.2 Problem formulation

In this section, we first describe a graph representation for KPD programs that include incompatible pairs as well as altruistic donors. Utilizing this graph representation, we then define the optimal policy in the management of a KPD program with one altruistic donor.

3.2.1 Graph representation

We represent a KPD program as a *directed graph*, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the *vertex set*, $\mathcal{V} \equiv \mathcal{V}(\mathcal{G}) = \{1, 2, \cdots, n\}$, consists of m altruistic donors and $n - m$ incompatible donor-candidate pairs, where $m \leq n$. We denote by, $\mathcal{V}_a \equiv \mathcal{V}_a(\mathcal{G}) = \{1, 2, \cdots, m\}$, the collection of altruistic donors, and $\mathcal{V}_p \equiv \mathcal{V}_p(\mathcal{G}) = \mathcal{V} \setminus \mathcal{V}_a$, the set of incompatible pairs. The *edge set*, $\mathcal{E} \equiv \mathcal{E}(\mathcal{G})$, is a binary relation on $\mathcal{V}$, consisting of ordered pairs of vertices in $\mathcal{V}$. An edge from i to j, denoted as (i, j), implies that the donor in pair i (or the altruistic donor i) is predicted to be compatible with the candidate in pair j. Such a prediction is based on a *virtual crossmatch* test, which involves computer cross-checking for blood type compatibility as well as comparing candidate antibodies against donor HLA antigens. Before a predicted compatible transplant can be further considered for an actual surgical operation, it must be confirmed by a labor-intensive *laboratory crossmatch* test to assure histocompatibility, which involves incubating the serum of a candidate with the white blood cells of a prospective donor. Figure 3.1 illustrates such a graph representation for a two-way exchange, a three-way exchange, and a NEAD chain.

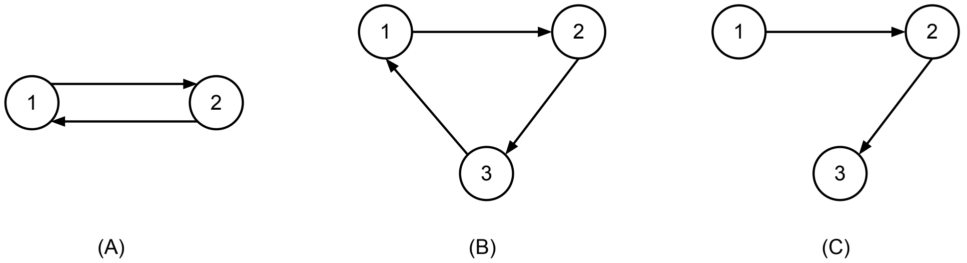

Figure 3.1: (A): A graph representation of a KPD program with a two-way exchange cycle, where $\mathcal{V}_p = \{1, 2\}$ and $\mathcal{E} = \{(1, 2), (2, 1)\}$; (B): A graph representation of a KPD program with a three-way exchange cycle, where $\mathcal{V}_p = \{1, 2, 3\}$ and $\mathcal{E} = \{(1, 2), (2, 3), (3, 1)\}$; (C): A graph representation of a NEAD chain, where $\mathcal{V}_a = \{1\}$, $\mathcal{V}_p = \{2, 3\}$ and $\mathcal{E} = \{(1, 2), (2, 3)\}$; donor 3 at the end of the chain becomes a bridge donor.

The virtual crossmatch test is necessary because in practice the laboratory crossmatch test cannot be undertaken on all possibly compatible donors and candidates due to labor and resource limitations. Further, even if the laboratory crossmatch result is negative (non-reactive), an actual transplant operation may not occur due to other friction including, for example, donor refusal and illness or death of the candidate. To incorporate such stochastic features, we associate with each edge, $e = (i, j)$, a probability (denoted as p_e or p_{ij}) that e, if chosen, could result in an actual transplant operation (Li et al., 2011). Throughout the rest of the chapter, we use the term "*is viable*" to indicate that an edge could lead to an actual transplant.

In addition, we associate with each edge (or potential transplant) a general utility (Li et al., 2011). Such utilities are often rule-based and determined by various attributes such as degree of sensitization of the candidate against the potential donor pool, or time since enrolment in the KPD. These utilities could also be based on predicted medical outcomes such as the estimated graft or patient survival, or the incremental years of recipient life that would accrue with a kidney transplant as opposed to remaining on dialysis. For each potential transplant $e = (i, j)$, we denote such an assigned utility as u_e or u_{ij}. In this chapter, our attention is not on the estimation of edge utilities and probabilities. It is worth noting though that research along this line is important and needed in the practical management of a KPD program; see more discussion on this aspect in Wolfe et al. (2008), Schaubel et al. (2009), and Li et al. (2011).

3.2.2 The optimal policy

The problem of selecting a long NEAD chain and then arranging transplants accordingly is that in practice this long chain can rarely be fully implemented. This is because the chain would break as soon as one transplant cannot proceed as planned. In this chapter, we propose to extend a NEAD chain sequentially in a near optimal way by selecting one potential transplant recipient at each time. In subsequent discussion, we note how this can be used as the basis of more general approaches.

Consider a KPD program with only one altruistic donor, i.e. $\mathcal{V}_a = \{1\}$. This naturally implies $(i, 1) \notin \mathcal{E}$, for all $i \in \mathcal{V}$. For $j \in \mathcal{V}$ such that $j = 1$ or $(1, j) \in \mathcal{E}$, let $\mathcal{G}(j) \equiv (\mathcal{V}_j, \mathcal{E}_j)$ be a *subgraph* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where

$$\begin{aligned}\mathcal{V}_j &= \{v \in \mathcal{V} : v \text{ is } \textit{accessible} \text{ from } j\},\\ \mathcal{E}_j &= \{(v_1, v_2) \in \mathcal{E} : v_1 \in \mathcal{V}_j, v_2 \in \mathcal{V}_j, v_2 \neq j\}.\end{aligned}$$

In this chapter, a vertex j is said to be accessible from a vertex i if $i = j$ or if there exists a set of edges in $\mathcal{E}$, denoted as $\{(i_k, i_{k+1}), k = 0, 1, \cdots, n\}$ such that $i_0 = i$ and $i_{n+1} = j$. In general terms, $\mathcal{G}(j)$ represents the resulting KPD graph if the transplant according to $(1, j) \in \mathcal{E}$ is arranged and j becomes a bridge donor.

Managing a KPD program with one altruistic donor could then be viewed as a sequential decision problem, in which we start with $U = 0$ and $\mathcal{G} = \mathcal{G}(1)$, and then repeat the following steps until $|\mathcal{V}(\mathcal{G})| = 1$:

(i) choose one edge from $A \equiv \{(1, j) : (1, j) \in \mathcal{E}\}$, say $(1, b)$.

(ii) if $(1, b)$ is viable, set

$$\begin{aligned}U &\leftarrow U + u_{1b},\\ \mathcal{G} &\leftarrow \mathcal{G}(b),\\ 1 &\leftarrow b;\end{aligned}$$

if, however, $(1, b)$ is not viable, set

$$\mathcal{G} \leftarrow \mathcal{G}_{-(1,b)}(1), \text{ where } \mathcal{G}_{-(1,b)} = (\mathcal{V}, \mathcal{E} \setminus \{(1, b)\}).$$

The above iterative procedure actually defines a *policy* in the context of managing a KPD program by specifying what action from A to take in step (i) at each loop; some sample policies are,

$$
\begin{aligned}
b &= \operatorname*{argmax}_{j:(1,j)\in A} u_{1j} \\
b &= \operatorname*{argmax}_{j:(1,j)\in A} u_{1j} p_{1j}.
\end{aligned}
$$

For any given policy on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the value of U after the loop terminates can be interpreted as the cumulative claimed utility. This value, which we denote by U_∞, is random; and its expectation could then be naturally used to evaluate the corresponding policy. Among all policies defined in the above way, the *optimal policy* refers to the one that generates the highest value of $E(U_\infty)$. This way of defining the optimal policy provides a formal framework that will prove convenient in later discussions, even though in general one rarely follows this optimal policy through until the iterative procedure ends. This is an important issue, arising due to various practical concerns, that we will revisit in Section 3.4.2.

Figure 3.2-A provides an illustrative example, where $\mathcal{G}$ represents a KPD program with four incompatible pairs (vertices 2, 3, 4 and 5) and one altruistic donor (vertex 1). Starting from $\mathcal{G}$, the action space is $A = \{(1,2),(1,3)\}$ and suppose we proceed by selecting $(1,2)$. If it is viable, this would lead to $\mathcal{G}(2)$, denoted as $\mathcal{G}_9$ in Figure 3.2-A, and the resulting value of U_∞ is u_{12}; if $(1,2)$ is not viable, we end up with $\mathcal{G}_1$, at which the updated action space becomes $A = \{(1,3)\}$. We then continue by selecting $(1,3)$, and if it is not viable, we stop at $\mathcal{G}_2$; if $(1,3)$ is viable, we then proceed to $\mathcal{G}_3$, at which the updated action space becomes $A = \{(3,4),(3,5)\}$; and we continue this process by selecting one allocation from A.

3.3 Standard decision tree analysis

The optimal policy introduced in the previous section can be obtained by conducting a standard decision tree analysis, which we briefly illustrate below using a small example. The computation associated with such an analysis, however, can be rather complicated for large problems. We will return to this computational issue in Section 3.4.

The structure of $\mathcal{G}$ in Figure 3.2-A cannot be used directly for a standard decision tree analysis due to the existence of various fall-back options; for example, if edge $(1,3)$ is selected but not viable, we could fall back to $(1,2)$. The complete analysis is instead provided by a derived decision tree (oriented from left to right) as in Figure 3.2-B, where squares represent *decision nodes* and circles indicate *chance nodes*. Each decision node is followed in this tree by a fixed number of chance nodes associated with all actions available at that decision node. Each chance node is then followed by two decision nodes corresponding to

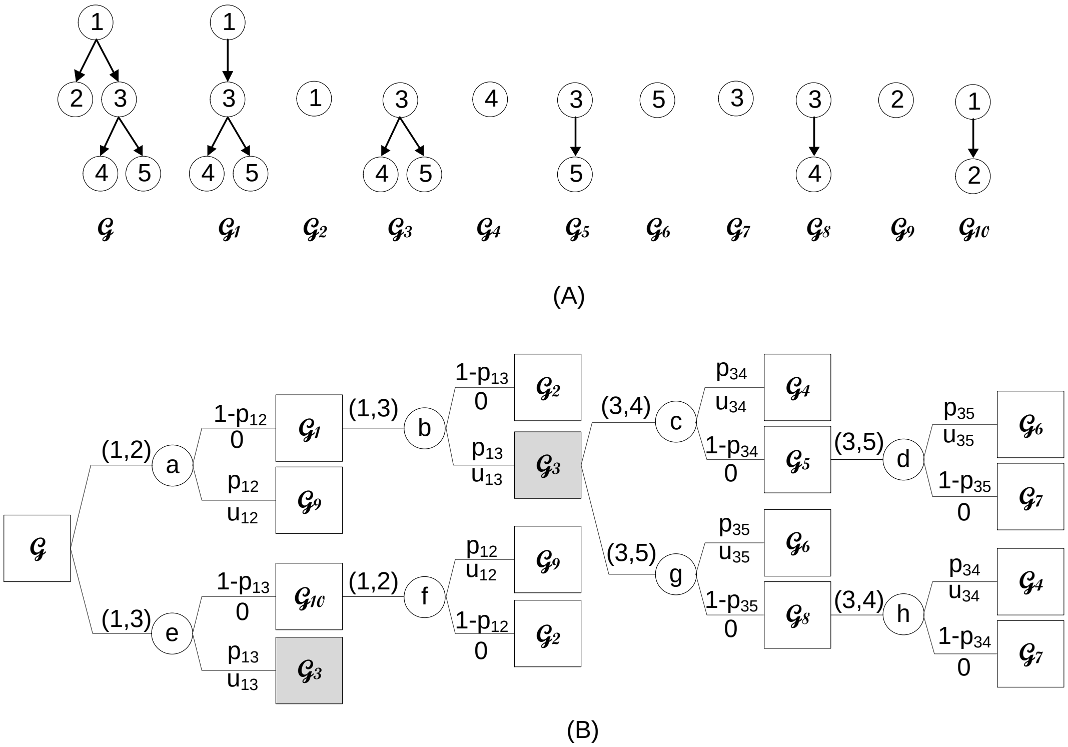

Figure 3.2: (A): A KPD program $\mathcal{G}$ with one altruistic donor and four incompatible pairs as well as various subgraphs of $\mathcal{G}$; (B): A standard decision tree analysis for a KPD program $\mathcal{G}$ as in (A), with squares representing decision nodes and circles indicating chance nodes; the decision node $\mathcal{G}_3$ (which is shaded) appears twice in the tree and hence is only drawn once.

the two possible outcomes of choosing that chance node: one outcome is that the chosen transplant $e \in \mathcal{E}$ is viable, resulting in a utility of u_e, whereas the other is that e is not viable, for which zero utility is generated. These two utilities are associated with the edges from the chance node to the two corresponding decision nodes. For example, in Figure 3.2-B, starting from the decision node $\mathcal{G}$, two actions are available, either arrange a transplant according to edge $(1, 2)$ leading to chance node a or according to edge $(1, 3)$ leading to chance note e. In the case where $(1, 2)$ is chosen, associated with the chance node a are two possible outcomes, $\mathcal{G}_1$ and $\mathcal{G}_9$, which occur with probabilities $1 - p_{12}$ and p_{12} respectively. If $\mathcal{G}_9$ occurs, we claim a utility of u_{12}, and zero utility is generated if $\mathcal{G}_1$ occurs, for which we continue on this analysis from chance node b.

The *Expected value* (EV) associated with a chance node or a decision node is calculated alternately in a backward direction along the tree from the right to the left. Precisely, (i) the EV at a leaf decision node is 0 (this could be set to some non-zero number to represent the potential value associated with the corresponding bridge donor; see more discussion on this in Section 3.6); (ii) the EV at a chance node is computed by taking a weighted average of the sums of the utilities along the edges originating at this chance node and the EVs at the corresponding successor decision nodes; (iii) the EV at a non-leaf decision node is calculated by taking the maximum of the EVs of its children nodes.

For example, in Figure 3.2-B, the EVs at decision nodes $\mathcal{G}_5$ and $\mathcal{G}_8$ are $EV[\mathcal{G}_5] = EV[d] = p_{35}u_{35}$ and $EV[\mathcal{G}_8] = EV[h] = p_{34}u_{34}$ respectively. The EVs at chance nodes c and g are $EV[c] = p_{34}u_{34} + (1 - p_{34})EV[\mathcal{G}_5]$ and $EV[g] = p_{35}u_{35} + (1 - p_{35})EV[\mathcal{G}_8]$ respectively. This indicates that $EV[c] \geq EV[g]$ if and only if $u_{34} \geq u_{35}$, and the action taken at $\mathcal{G}_3$ is therefore $(3, 4)$ or $(3, 5)$ depending on which one has the larger edge utility. The EV at node $\mathcal{G}_3$ is then calculated as

$$\begin{aligned} EV[\mathcal{G}_3] &= \max\{EV[c], EV[g]\} \\ &= \max\{p_{34}u_{34} + (1 - p_{34})p_{35}u_{35}, p_{35}u_{35} + (1 - p_{35})p_{34}u_{34}\}. \end{aligned} \tag{3.1}$$

After computing EVs associated with all decision and chance nodes in this way, the optimal policy is then to adopt, at each decision node, the action associated with the chance node that has the maximum EV. This procedure starts from the root decision node.

3.4 A look-ahead search tree-based strategy

The structure of the derived decision tree in Figure 3.2-B is much more complicated than the structure of $\mathcal{G}$ itself in Figure 3.2-A. As a result, the standard decision tree analysis as introduced in Section 3.3 results in substantial computational difficulties when the KPD graph is large. To address this issue, we propose in this section a more efficient and practical approach that relies on evaluating different allocations for each altruistic donor (or bridge donor) according to a derived look-ahead search tree.

3.4.1 Identifying the optimal policy via a search tree

First, we observe the following fact: consider a KPD program, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}_a = \{1\}, \mathcal{V}_p = \{2, 3, \cdots, n\}$, and $\mathcal{E} = \{(1, i) : i = 2, 3, \cdots, n\}$. Without loss of generality, assume $u_{12} \geq u_{13} \geq \cdots \geq u_{1n}$. For this specific KPD program, the optimal policy to follow at $\mathcal{G}$ is to try transplant $(1, 2)$, and if it fails then try $(1, 3)$, then $(1, 4)$ and so forth.

The associated EV of this policy is

$$EV[\mathcal{G}] = \sum_{k=2}^{n} \left\{ u_{1k} p_{1k} \prod_{i=2}^{k-1} (1 - p_{1i}) \right\}. \tag{3.2}$$

Based on this fact, we could then select the optimal action to take from $\mathcal{G}$ directly and hence avoid explicitly constructing a decision tree and calculating the EV associated with each node of the tree, as required by the standard decision analysis in Section 3.3. For example, applying formula (3.2) at the decision node $\mathcal{G}_3$ in Figure 3.2-B would lead to the optimal action of taking $(3, 4)$ or $(3, 5)$ depending on which one has a larger utility; and the EV at $\mathcal{G}_3$ is therefore computed as

$$EV[\mathcal{G}_3] = \mathbf{1}_{[u_{34} \geq u_{35}]} \{p_{34}u_{34} + (1 - p_{34})p_{35}u_{35}\} + \mathbf{1}_{[u_{34} < u_{35}]} \{p_{35}u_{35} + (1 - p_{35})p_{34}u_{34}\}. \tag{3.3}$$

Note that formula (3.3) is exactly equal to the one calculated via a standard decision analysis as in formula (3.1), which, however, requires calculating EVs at additional nodes $\mathcal{G}_5$ and $\mathcal{G}_8$.

Further, the following result holds. For a KPD program, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}_a = \{1\}$, let $A \equiv \{(1, j) : (1, j) \in \mathcal{E}\}$ and $u^*_{1j} \equiv u_{1j} + EV[\mathcal{G}(j)]$, for all $(1, j) \in A$. Without loss of generality, we assume $A = \{(1, j) : j = 2, 3, \cdots, l\}$ and $u^*_{12} \geq u^*_{13} \geq \cdots \geq u^*_{1l}$. Then the optimal decision to take at $\mathcal{G}$ is to attempt transplant $(1, 2)$; and if it fails, try $(1, 3)$ and then $(1, 4)$, and so on; the associated EV is

$$EV[\mathcal{G}] = \sum_{k=2}^{l} \left\{ p_{1k} u^*_{1k} \prod_{i=2}^{k-1} (1 - p_{1i}) \right\}. \tag{3.4}$$

Based on this result, we could evaluate various choices in A by u^*_{1j} and then proceed with the one having the largest value. We repeatedly apply this procedure from terminal nodes up to sequentially form a NEAD chain, with formula (3.4) evaluating the expected utility in this process.

To identify the optimal action to take at $\mathcal{G}$, we recursively apply formula (3.4), which in fact does not require calculating EVs associated with all decision nodes and chance nodes, but only a fraction of them. These requited ones can then be organized according to their dependence relationship as in (3.4) to form a *search tree*. In this tree, the node on the left hand side of (3.4) is the *parent* while the nodes on the right hand side denote *children*; and edges connecting them represent the corresponding actions. The structure of this tree therefore allows us to compute EVs associated with its nodes recursively in a backward manner from the leaf nodes to the root.

Figure 3.3 provides an example of a search tree and illustrates calculating EVs associated with its nodes; the search tree in this figure only involves 5 nodes, much less than that of the decision tree in Figure 3.2-B. The optimal action to take at $\mathcal{G}$ in this example is edge $(1, 3)$ if $u^*_{12} = u_{12}$ is smaller than $u^*_{13} = u_{13} + EV[\mathcal{G}_3]$ or edge $(1, 2)$ if otherwise; the EV at $\mathcal{G}$ is therefore computed as

$$EV[\mathcal{G}] = \mathbf{1}_{[u^*_{13}\geq u^*_{12}]}\left\{p_{13}u^*_{13} + (1-p_{13})p_{12}u^*_{12}\right\} + \mathbf{1}_{[u^*_{13}<u^*_{12}]}\left\{p_{12}u^*_{12} + (1-p_{12})p_{13}u^*_{13}\right\}.$$

Clearly, the decision analysis according to this search tree is much simpler than the one by a standard decision tree as in Figure 3.2-B.

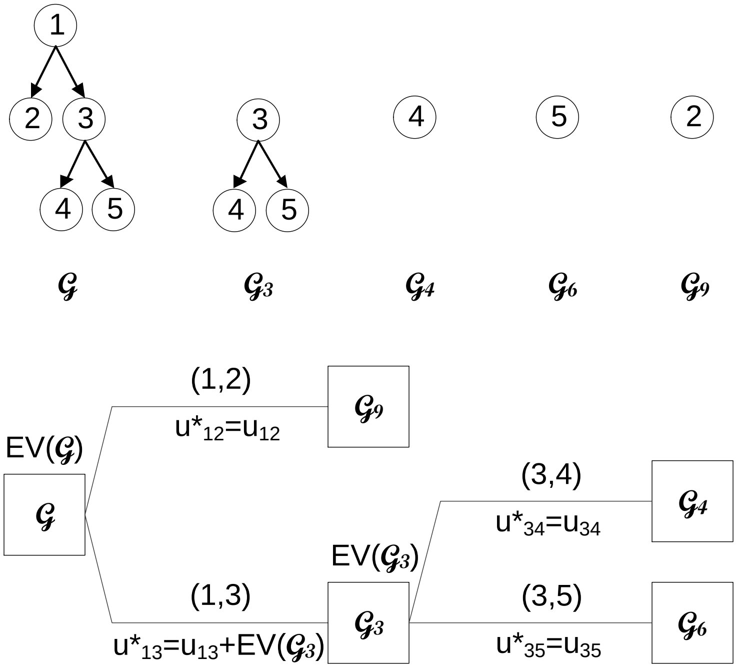

Figure 3.3: A search tree-based analysis for a KPD program $\mathcal{G}$.

In general, the search tree associated with a KPD program can be constructed by an algorithm based on the classic depth-first search (DFS). We developed such an algorithm that also computes the EVs while performing a DFS on the KPD graph. The optimal policy is then determined by the following iterative procedure:

(i) for $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, construct the corresponding search tree by following a DFS-based algorithm, and compute the EV associated with each node of this search tree; this is done recursively from the terminal nodes up to the root node.

(ii) update the current action space $A \equiv \{(1,j) : (1,j) \in \mathcal{E}(\mathcal{G})\}$, and calculate $u^*_{1j} = u_{1j} + EV[\mathcal{G}(j)]$ for $(1,j) \in A$.

(iii) choose $(1,b) \in A$ with $b = \text{argmax}_{j:(1,j)\in A}\, u^*_{1j}$.

(iv) if $(1,b)$ is viable, set

$$\begin{aligned} &\mathcal{G} \leftarrow \mathcal{G}(b), \ \text{i.e. update the KPD graph,} \\ &1 \leftarrow b, \ \text{i.e. set the bridge donor } b \text{ as the new altruistic donor;} \end{aligned}$$

if, however, $(1,b)$ is not viable, set

$$\mathcal{G} \leftarrow \mathcal{G}_{-(1,b)}(1), \ \text{where } \mathcal{G}_{-(1,b)} = (\mathcal{V}, \mathcal{E} \setminus \{(1,b)\})\,.$$

(v) go back to (ii) until $|\mathcal{V}(\mathcal{G})| = 1$.

3.4.2 A depth-k search tree

Although the search tree-based approach allows for a much more efficient analysis than the analysis produced by a standard decision tree, constructing such a search tree, is still NP-hard for a general KPD graph, even without considering the effort entailed in computing EVs associated with nodes along that tree. This unfortunate fact poses a substantial difficulty in identifying the optimal policy when the KPD program is large.

Further, a more important issue is that the optimal policy (assuming it is readily computable) will very likely turn out to be not thoroughly implementable in practice. This is mainly because in practice the process of initiating and extending a NEAD chain would require a relatively long period of time, during which the KPD pool would constantly update itself in that new pairs might arrive and existing ones might withdraw; patients in the pool might also be transplanted via exchanges among incompatible pairs, which could be arranged in parallel with the NEAD chain mechanism. Thus, looking a long way down the tree in assessing strategies from the root node is generally not that useful.

To alleviate such problems, we propose to proceed by first deriving a subtree from the original search tree, which we call a *depth-k search tree*. Such a subtree can be readily obtained by the same DFS-based algorithm as introduced in Section 3.4.1, by simply restricting the depth of the search from the root note to k. We then follow the recursive relationship as in formula (3.4) to calculate the EVs associated with corresponding nodes in the subtree. We start this calculation from the leaf nodes at depth k and work up through the tree to the root node. The EV of a leaf node is set to zero or some reasonable measurement of the value of the corresponding bridge donor (see Section 3.6 for more discussion). Notice that at this stage the iterative procedure presented in Section 3.4.1 could be similarly applied, but with one modification that — if the chosen action $(1, b)$ is viable, we regenerate a depth-k search tree rooted at $\mathcal{G}(b)$ and compute EVs associated with the nodes of this new tree.

In exchange for the optimality that exists only in a rather idealized scenario, the policy obtained according to a depth-k search tree provides a more practical evaluation of potential bridge donors and a greatly reduced computational complexity. Further, our simulation results (see Section 3.5) suggest that the allocation strategy derived from a search tree performs reasonably well for a moderate depth of, say, 3 or 4. Notice that the depth-1 search tree constructed according to formula (3.2) provides the same analysis as the one via a standard decision tree of depth $2n + 1$.

3.5 Simulation studies

So far in this chapter, we have explored a look-ahead search tree-based approach to manage KPD programs with one altruistic donor. This approach sequentially extends a NEAD chain by selecting one potential bridge donor at one time, taking into consideration the operational uncertainties and the long-term consequences associated with various possible selections. In this section, we provide simulation results of applying such an allocation strategy to manage a simulated KPD program.

3.5.1 Simulating incompatible pairs and altruistic donors

We simulate incompatible pairs and altruistic donors as in Li et al. (2011). For an incompatible pair, we simulate its candidate and donor separately from their own population distributions. Candidates are sampled at random (with replacement) from a database of incompatible pairs, which is derived from the University of Michigan KPD program. This database currently consists of 115 transplant candidates, each having at least one willing but incompatible donor. We aim to incorporate additional databases from other KPD programs for the purpose of having a wider candidate variation. Donors, on the other hand, are simulated by sampling their blood types and HLA haplotypes respectively. Blood type is drawn from its U.S. population distribution: O, 44%; A, 42%; B, 10%; and AB, 4%

(Stanford Blood Center, 2010). Then, donor's HLA haplotypes are sampled according to a population frequency table derived from a public database on potential bone marrow donors (Maiers et al., 2007).

We consider a simulated donor-candidate pair as an incompatible pair and hence include it in the KPD pool, if either their blood types mismatch or the donor's HLA haplotypes overlap with some of the candidate's antibody specificities. Finally, an altruistic donor is generated in the same way as we have described above for generating a donor in an incompatible pair.

3.5.2 Simulation setup

In the KPD graph representation as in Section 3.2.1, each potential transplant (which is predicted to be compatible by a virtual crossmatch test) is assigned a probability to reflect the inherent uncertainty in the system and a general utility to quantify the rule-based or outcome-based evaluation of that potential transplant. As we have mentioned, estimation of these probabilities and utilities is an important aspect in the practical management of a KPD program. This also forms an independent line of research in parallel with the work of developing KPD allocation strategies. For illustrative purpose, our approach here is to obtain these utilities and probabilities according to certain simplified probability distributions, and then use them to study the method proposed in Section 3.4.2.

We perform a total of $3,000$ simulations. In all simulations, edge probabilities are generated from a uniform distribution, $U(0.1, 0.5)$, which suggests an average successful rate of 30% for a predicted compatible (by virtual crossmatch test) transplant. This rate is in line with our experience at the University of Michigan KPD program and the Alliance for Paired Donation. For edge utilities, we fix them at 1 for $1,000$ simulations, draw them from uniform $U(10, 20)$ and $U(10, 30)$ respectively for the remaining $2,000$ simulations (with $1,000$ each). For each simulation, we execute an allocation strategy based on depth-k search tree for k equal to 1, 2, 3, 4, and 5 respectively. We then record important performance measures such as cumulative claimed utilities and cumulative number of transplants. Note that when k is equal to 1, the allocation strategy simply corresponds to selecting, among all possible choices available for the altruistic donor, the one that has the largest edge utility.

3.5.3 Simulation results

First, we report on the cumulative number of transplants achieved in simulated KPD programs with one altruistic donor and 100 incompatible donor-candidate pairs. We compare the average number of transplants across different values of k and under the three utility generating distributions. Table 3.1 provides summary comparison, in which we observe a consistent pattern where the number of transplants performed increases with k. This is true regardless of which distribution is used to generate edge utilities.

	$u_e = 1$	$u_e \sim U(10, 20)$		$u_e \sim U(10, 30)$	
depth-k	mean N	mean N	mean U_∞	mean N	mean U_∞
$k = 1$	3.16	3.18	55.99	3.18	80.22
$k = 2$	8.17	6.65	112.21	6.19	149.58
$k = 3$	8.70	7.87	128.93	7.63	176.78
$k = 4$	8.74	8.26	133.54	7.99	181.92
$k = 5$	8.89	8.41	134.45	8.29	185.43

Table 3.1: Summary of the averaged number of transplants performed (denoted by N) and the averaged cumulative utilities claimed (denoted by U_∞), by implementing a depth-k search tree-based allocation strategy on a simulated KPD program (with one altruistic donor and 100 incompatible pairs). Edge utilities are generated from $U(1, 1)$, $U(10, 20)$, and $U(10, 30)$; and edge probabilities are generated from $U(0.1, 0.5)$. The summary is calculated over $3,000$ rounds of simulations, with $1,000$ simulations for each utility generating distribution.

Another observation is that the extra benefit in the number of transplants through increasing k is diminishing as k gets large. For example, when edge utilities are generated from $U(10, 20)$, increasing the value of k from 1 to 4 would almost triple the total number of transplants (on average from 3.18 to 8.26); however, further increasing k (from 4 to 5) appears to have very limited effects.

In terms of comparing the cumulative claimed utility, Table 3.1 also demonstrates similar patterns to those observed above for comparing the number of transplants. These results suggest that $k = 3$ or 4 would provide a satisfactory solution in practice. Further investigation, however, with data from more KPD programs could be useful. Notice that when edge utilities are fixed at 1, the cumulative claimed utility is the same as the cumulative number of transplants.

Finally, we take a look at the correlation matrix among five variables; each variable represents the number of transplants performed when k is equal to each one of the five values. We anticipate that the correlation between variable 4 (the number of transplants achieved when $k = 4$) and variable 5 would be higher than the correlation between variable 1 and variable 5. Table 3.2 exactly unveils such a pattern in three correlation matrices (with each one corresponding to one utility generating distribution). Similar observation is noted as well in correlation matrices for the cumulative claimed utility.

1	.34	.34	.33	.31
−	1	.67	.67	.66
−	−	1	.79	.73
−	−	−	1	.81
−	−	−	−	1

1	.27	.29	.26	.29
−	1	.50	.51	.48
−	−	1	.66	.58
−	−	−	1	.72
−	−	−	−	1

1	.29	.27	.29	.29
−	1	.46	.48	.48
−	−	1	.58	.57
−	−	−	1	.72
−	−	−	−	1

Table 3.2: Three correlation matrices for the total number of transplants performed in a depth-k search tree-based allocation strategy across different values of k. The entry at the ith row and the jth column represents the correlation between the total number of transplants when $k = i$ and that when $k = j$ when managing the same simulated KPD program (with one altruistic donor and 100 incompatible pairs). Matrix on the left: $u_e = 1$; matrix in the middle: $u_e \sim U(10, 20)$; matrix on the right: $u_e \sim U(10, 30)$.

3.6 Concluding remarks

In this chapter, we have studied the problem of managing a KPD program with one altruistic donor. One important yet challenging part of this problem is to recognize various friction (as discussed in Section 3.2.1) inherent in the system and to guide the decision-making process accordingly by taking into account these uncertainties. Realizing the fact that a long pre-specified NEAD chain in practice can almost never be implemented as planned, we propose to initiate and extend such a chain in a sequential way by selecting potential transplant recipients one at a time. Each selection is made keeping in mind the associated long-term consequences so as to maximize the expected gain over a certain given number of moves. In order to do this efficiently and practically, we construct a depth-k search tree for a KPD graph using a DFS-based algorithm. We then evaluate various choices available for each altruistic donor (or bridge donor) according to the calculation performed along that search tree, and recommend the choice with the greatest evaluation.

In the process of extending a NEAD chain, the bridge donor at the end of the current chain might be incompatible with a majority of the candidate population, which would significantly prolong the waiting time for that bridge donor to be matched with a present or future candidate. Furthermore, this hard-to-match bridge donor may also be more likely to withdraw from the KPD pool and so terminate the NEAD chain. This is because in general the longer the waiting time to match a bridge donor, the higher the probability that this bridge donor may reconsider the original promise of making a donation. To partially avoid this unfortunate circumstance, some KPD programs have not allowed a blood type AB donor to become a bridge donor unless an immediate match is available. Actually, this issue can be paretically addressed by our proposed sequential allocation strategy, because when evaluating various choices available for an altruistic donor, our strategy, by incorporating the long-term consequences associated with each choice, would in general very unfavorably assess the choice that could lead to a hard-to-match bridge donor. One way to further address this issue, as we have briefly mentioned in Section 3.3 and Section 3.4.2, is to assign each possible bridge donor a reasonable *base utility*. This base utility represents the potential "contribution" from a bridge donor; naturally, a hard-to-match bridge donor would be assigned a small base utility and an easy-to-match one would be given a large value.

Although this chapter has focused on managing a KPD program with one altruistic donor, the proposed approach can be generalized to incorporate multiple altruistic donors. One way to achieve this is for each altruistic donor to construct a depth-k search tree and use this tree to evaluate various allocations options available for that altruistic donor according to the method in Section 3.4.2. Among all allocations possible for these altruistic donors, we select a disjoint collection such that the overall expected utility can be maximized. More specifically, let $\mathcal{V}_a \equiv \{1, 2, \cdots, m\}$ be m altruistic donors in a general KPD program. We denote by $\mathcal{A} \equiv \cup_{i=1}^{m}\{(i,j) \in \mathcal{E}\}$ the possible allocations available for these m altruistic donors. Each potential transplant, $(i,j) \in \mathcal{A}$, can be evaluated by its expected utility, which is calculated as $u_{ij}^* = u_{ij} + EV[\mathcal{G}_{[i]}(j)]$, where $\mathcal{G}_{[i]} \equiv \mathcal{G}(i)$. We then select from $\mathcal{A}$ a disjoint collection of edges (or transplants), in the sense that no two edges can share a common vertex, so as to maximize the sum of expected utilities. For those selected transplants, viable ones would result in actual operations and generate new bridge donors, and altruistic donors and incompatible pairs involved in non-viable transplants are recycled back to the KPD pool.

Notice that the above way of allocating multiple altruistic donors can be arranged in parallel with the way of selecting exchange sets (or cycles) among incompatible pairs. Let $\mathcal{S}_k$ the collection of all exchange sets of size up to k among $(n-m)$ incompatible pairs. For each $S \in \mathcal{S}_k$, EU_S represents its expected utility and Y_S is a decision variable equal to 1 if S is selected and 0 if not; for each $(i,j) \in \mathcal{A}$, Z_{ij} is another decision variable whose value is 1 if (i,j) is chosen for a transplant and 0 otherwise, and this potential transplant (i,j) is evaluated by u_{ij}^*. We then manage such a KPD program by solving the following IP problem:

$$\max_{\{Y_S\},\{Z_{ij}\}} \left\{ \sum_{S \in \mathcal{S}_k} Y_S EU_S + \sum_{(i,j) \in \mathcal{A}} Z_{ij} u_{ij}^* \right\}, \tag{3.5}$$

$$\text{subject to} \sum_{S \in \mathcal{S}_k(l)} Y_S + \sum_{(i,j) \in \mathcal{A}(l)} Z_{ij} \le 1, \forall l \in \mathcal{V}, \tag{3.6}$$

where, in (3.6), $\mathcal{S}_k(l)$ represents the exchange sets in $\mathcal{S}_k$ that contain l and $\mathcal{A}(l)$ similarly denotes a subset of transplants in $\mathcal{A}$ that involve l.

The solution to this IP problem specifies in a systematic way which transplants to choose for those m altruistic donors as well as how to arrange exchanges among the remaining $(n - m)$ incompatible pairs. With regard to altruistic donors, we proceed by selecting a set of transplants according to $\{(i, j) \in \mathcal{A} : Z_{ij} = 1\}$. On the other hand, to organize exchanges among incompatible pairs as determined by $\{S \in \mathcal{S}_k : Y_S = 1\}$, we arrange transplants according to the cycle within a chosen exchange set that has the largest utility, and fall back to the next largest one (when possible) if the initial targeted cycle is not viable. Therefore, this strategy of simultaneously allocating altruistic donors and selecting exchange sets is particularly useful, at least in the practical sense, for the management of large KPD programs with a moderate number of altruistic donors.

The mechanism of a NEAD chain allows the altruism from a single altruistic donor to benefit a potentially large number of patients, but it does so exclusively for patients recruiting a willing but incompatible living donor. This mechanism excludes patients without a designated living donor and who are therefore placed on a deceased-donor waiting list. Among those patients who would benefit from this NEAD chain, approximately 73% of them are white; whereas those who would not benefit from this mechanism form a 52% non-white population (Segev et al., 2008). On the other hand, not all altruistic (or bridge) donors are well suited for initiating and extending chains among a pool of incompatible pairs. For example, consider a KPD pool in which an altruistic donor may not be in a good position (because of either incompatibility or poor utility) to be matched up with any candidate. In this case, rather than placing this altruistic donor in a waiting "mode" for a potentially long time, redirecting him/her to a deceased-donor waiting list, where a compatible patient with potentially good transplant outcomes might be identified rather easily, appears a more suitable alternative.

To decide whether an altruistic donor is better off initiating a NEAD chain or donating directly to someone on a deceased-donor waiting list, we suggest to evaluate an altruistic donor by the utility expected to achieve if this donor is chosen to initiate a chain. To be precise, we may first perform the calculation according to formula (3.4) as in Section 3.4.1 to evaluate the expected utility for each altruistic donor, i.e. $\{EV[\mathcal{G}(i)] : i \in \mathcal{V}_a\}$. The result from this evaluation could then be used to assess the suitability of assigning each altruistic donor to a deceased-donor waiting list; a relatively high value of $EV[\mathcal{G}(i)]$ would recommend reserving altruistic donor i for extending a NEAD chain while a comparatively low value of $EV[\mathcal{G}(i)]$ would indicate a transplant to someone waiting for a deceased-donor kidney. It is worth pointing out that different ways of assigning edge utilities and probabilities could be adopted in calculating $\{EV[\mathcal{G}(i)] : i \in \mathcal{V}_a\}$. This would provide extra benefit in allowing more control over what kidneys in general are distributed to a deceased-donor waiting list. For example, if each edge is assigned an equal utility while the edge probability remains representing the likelihood of that edge being viable, then altruistic (or bridge) donors who are less compatible with candidates in the current KPD pool would be more likely to be directed to a deceased-donor waiting list.

CHAPTER 4

On Various Aspects of Managing KPD Programs

In this chapter, we examine in greater detail two important aspects in the management of a KPD program — robust organ allocations and operational uncertainties. We then propose a general mathematical framework based on Markov decision processes (MDPs) to systematically formulate the problem of managing KPD programs in the presence of altruistic donors. Finally, we conclude with some discussion and future directions of this dissertation work.

4.1 Introduction

This chapter synthesizes and builds upon the work from previous chapters. One major aim is to extend approaches introduced in earlier chapters to more general and practical situations. We first extend the idea of using exchange sets for planning organ allocations; the outcome of this extension is a more robust allocation strategy derived from the concept of strongly connected components (SCCs). We then study more carefully the operational uncertainties inherent in the management of KPD programs. To better describe these stochastic features, we propose to differentiate them as the uncertainty on the edges and the uncertainty on the vertices. As a consequence, the independence assumption (among edges) adopted in previous chapters can be relaxed, and accordingly the method of computing expected utilities can be further generalized.

Another contribution of this chapter is in the development of a general mathematical framework, which can be used to rigorously formulate the problem of managing KPD programs with altruistic donors. This framework is derived from theories of Markov decision processes (MDPs). MDP-based approaches have been widely and successfully used in many industrial and operational applications, but are underutilized in the field of medical decision making (Alagoz et al., 2010). In this chapter, we present this MDP framework for KPD programs with one altruistic donor, in which the goal is to specify a desired *action* for each *state* of the program such that the expected utility would be maximized. We then generalize this framework to incorporate multiple altruistic donors. Finally, we conclude with some discussion and potential future directions.

4.2 Strongly connected components

In Chapter 2, we have developed three allocation strategies for organizing organ exchanges in a KPD program; two of them are cycle-based approaches — choosing disjoint exchange cycles according to different criterion, and the third one, in contrast, is set-based, aiming to select a collection of mutually exclusive exchange sets. The benefit of considering exchange sets as opposed to cycles, as we have discussed earlier in this dissertation, lies in that doing so would allow us to exploit potential fall-back options when the originally planned allocation cannot be fully executed due to various uncertainties in a KPD program.

Actually, being able to provide such fall-back options can go beyond the idea of using exchange sets to plan for organ allocations. An exchange set requires that there is a cycle involving all elements of that set; this restriction is somewhat arbitrary and can be relaxed. Consider, for example, Figure 4.1, which consists of three incompatible pairs and two two-way cycles. According to our previous definitions, this structure is neither an exchange set nor an exchange cycle; therefore, it would not be selected as part of a cycle-based or set-based allocation. On the other hand, considering this structure while planning for organ allocations does provide advantages: if all edges are viable, we could simply arrange transplant operations according to one of the two cycles that has a larger utility; if certain edges are not viable, we could still fall back to a back-up cycle provided that both edges of that cycle are viable.

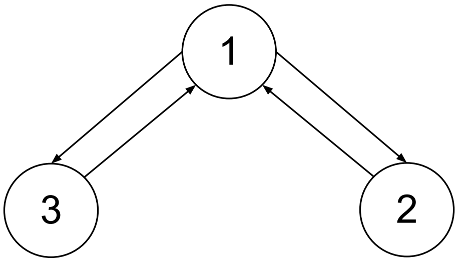

Figure 4.1: Three incompatible pairs forming two two-way cycles; this structure is an SCC of size three, denoted as $S = (\{1,2,3\}, \{(1,2),(2,1),(1,3),(3,1)\})$.

For a KPD program $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the above example leads us to define a *strongly connected component* (SCC) as an *induced subgraph* (see the definition in Chapter 2) of $\mathcal{G}$ such that within this subgraph every vertex is *accessible* from all other vertices. In this dissertation, a vertex j is said to be accessible from a vertex i if $i = j$ or if there exists a set of edges in $\mathcal{E}$, denoted as $\{(i_k, i_{k+1}), k = 0, 1, \cdots, n\}$ such that $i_0 = i$ and $i_{n+1} = j$. The *size* of an SCC is simply the number of its vertices.

It is straightforward to notice that every exchange set is indeed an SCC, but not vice versa. In fact, the only SCC of size three that is not an exchange set is the one in Figure 4.1. The expected utility associated with an SCC could be calculated in a way similar to that for evaluating exchange sets; for example, the SCC in Figure 4.1 has an expected utility of

$$\begin{aligned} p_{12}p_{21}p_{13}p_{31} \max\{u_{12} + u_{21}, u_{13} + u_{31}\} + \\ p_{12}p_{21}(1 - p_{13}p_{31})(u_{12} + u_{21}) + p_{13}p_{31}(1 - p_{12}p_{21})(u_{13} + u_{31}). \end{aligned}$$

Clearly, this way of measuring an SCC takes into consideration various operational uncertainties in a KPD program as well as possible contingent plans. We then define an SCC-based allocation as a disjoint collection of SCCs of size up to k; and the corresponding optimal allocation (i.e. the SCC-based allocation with the largest expected utility) can be found by adopting an IP formulation that is similar to those used previously. In practice, we only consider SCCs of size, say three or four, due to similar logistical concerns as discussed earlier in this dissertation.

In addition to planning organ exchanges by SCCs, the structure of an SCC could be further explored to reduce the required computation in managing a KPD program. We have observed in Chapter 2 that the computation associated with organizing kidney exchanges is rather expensive when managing large KPD programs. As Table 2.1 in Chapter 2 has suggested, the number of exchange sets increases exponentially with the size of a KPD graph. Therefore, it would take a substantial amount of computing time to even enumerate and store this huge number of exchange sets in a large program, not to mention that formulating and solving the corresponding IP problem would require even more computing power according to our experience from simulation studies.

This exponential growth rate, on the other hand, implies that managing two KPD programs of size n each would actually involve fewer computational resources than managing one KPD program of size $2n$. In other words, if a large KPD program can be divided into several smaller "separated" pieces, we could first work on each piece individually and concurrently and then combine together their results. This strategy would offer a significant amount of computational benefit; and the main idea is to divide a large KPD graph into several disjoint *strongly connected regions*. Below, we briefly describe this "divide-and-conquer" strategy.

We call two vertices u and v of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *connected* if u is accessible from v, and v is accessible from u. It is straightforward to check that this relation between vertices is *reflexive*, *symmetric*, and *transitive*; and hence it is an *equivalence* relation on $\mathcal{V}$. Further, this equivalence relation partitions $\mathcal{V}$ into disjoint sets, the subgraph *induced* by which are called *strongly connected regions* of the graph $\mathcal{G}$. In Figure 4.2, there are four strongly connected regions.

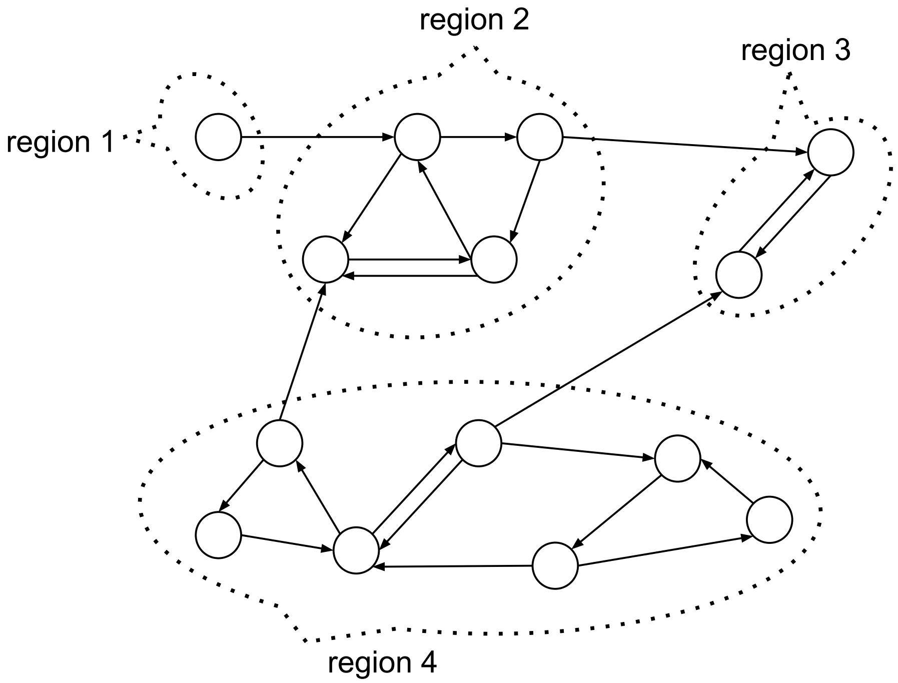

Figure 4.2: A KPD graph with four strongly connected regions.

In the management of KPD programs, the following two approaches actually produce exactly the same set-based (or cycle-based or SCC-based) allocation: (i) enumerate all exchange sets (or cycles or SCCs) of $\mathcal{G}$ with certain size restriction, and then formulate and solve the corresponding IP problem; (ii) partition $\mathcal{G}$ into disjoint strongly connected regions, and apply the former approach to each one of the regions. However, approach (ii) would greatly reduce the required computation, as we have previously discussed, if in practice, a large KPD program can indeed be divided into several smaller regions.

The number of strongly connected regions and the size of each region vary from one KPD program to another. We should also notice that no gain in reduced computation can be achieved if a large KPD graph itself is a strongly connected region, or the gain is rather limited if there is a single region involving most of the pairs in the graph. In these cases, an approximate approach may be employed, which is to first remove some edges from the KPD graph according to certain criteria; for example, an edge will be removed if its associated utility is less than certain value or its associate probability is smaller than some number, or both. These threshold values should be determined in advance with expert guidance from clinicians. We then partition the resulting graph into disjoint strongly connected regions, and, for each region, adopt a set-based (or SCC-based) allocation with those removed edges reinstalled. It is worth noting that the trade-off between such an approximate (and ad hoc) strategy and reduced computation is not clear at this stage. Further examination with data from other KPD programs would certainly be useful to address these aspects.

4.3 Uncertainties in a KPD program

In Chapter 2, we have proposed allocation strategies to take into consideration various uncertainties inherent in the management of a KPD program. One such uncertainty arises when performing a confirmatory laboratory test on predicted compatible pairs, because the result from this laboratory test might disprove what an earlier-performed computer test has suggested. Uncertainty can also arise due to the fact that a candidate might develop new antibodies against the allocated donor after the laboratory test but before the actual transplant operation, or that a donor-candidate pair may not accept the proposed assignment. For differentiation purpose, we call this type of uncertainty the *uncertainty on the edges*. The other type of uncertainty could arise, for example, when a candidate or donor becomes ill before a scheduled transplant can be carried forward or when a candidate dies or a donor is (temporarily) not available due to scheduling conflict. These events would affect any organ assignment involving that donor-candidate pair. We, therefore, call this type of uncertainty the *uncertainty on the vertices*.

To explicitly recognize these stochastic features, we have associated with each edge (or potential transplant) in a KPD graph, $e \in \mathcal{E}$, a probability p_e; this probability corresponds to the chance that the edge e, if chosen, would actually result in a transplant operation (i.e. the chance of edge e being viable). Then the probability that an exchange cycle C is viable can be computed as $\prod_{e \in \mathcal{E}(C)} p_e$, assuming that edges in a cycle have an independence relationship. Formally speaking, this assumption states that $\{X_e : e \in \mathcal{E}(C)\}$ forms an independent collection of random variables, where X_e is a Bernoulli variable taking value 1 if e is viable, and 0 otherwise. In this section, we first argue that this independence assumption is reasonable if the uncertainty on the vertices is negligible.

4.3.1 Ignoring the uncertainty on the vertices

Consider, for example, a three-way exchange cycle, denoted as $C_1 = \langle i, j, k \rangle$, in which all six donors and candidates are guaranteed to be available for transplant operations in the next scheduled match run. Put in other words, assume that no pair sickness, death, or scheduling conflict is involved in the process of arranging transplants according to this three-way cycle. In this example, where the uncertainty on the vertices is eliminated, whether $X_{e_{ij}}$ is equal to 1 or 0 will only be affected by characteristics of donor i and candidate j. These characteristics determine whether the laboratory crossmatch test would disprove the compatibility predicted by a computer crossmatch test or whether candidate j would develop new antibodies against donor i after the laboratory test but before the actual transplant operation. Similarly, the value of $X_{e_{jk}}$ is only affected by characteristics of donor j and candidate k; and the value of $X_{e_{ki}}$ is determined by characteristics of donor k and candidate i. Therefore, in this setting, it is reasonable to assume that $\{X_{e_{ij}}, X_{e_{jk}}, X_{e_{ki}}\}$ forms an independent collection of Bernoulli variables.

In practice, the uncertainty on the edges mostly depends on the quality of a computer crossmatch test. Thanks to recent advances in technologies of antibody identification, the virtual computer test can serve as a good tool to predict the actual HLA compatibility between donor and candidate. According to a recent study by Tambur et al. (2009), the false-negative (i.e. the computer test predicts a non-reactive or negative result whereas the laboratory test result is reactive or positive) rate could reach a level that is lower than 10%, and the false-positive rate could be as low as 15%. It is worth mentioning that these rates vary among transplant centers depending on what antibody identification method is used in the computer test.

However, for KPD programs with a moderate level of uncertainty on the vertices, the independence assumption is no longer valid. As a result, we need a more general way to calculate the probability that an exchange cycle or an exchange set is viable. In what follows, we discuss these aspects in detail.

4.3.2 Considering uncertainties on both the edges and the vertices

We first illustrate that the independence assumption would no longer hold when the uncertainty on the vertices cannot be ignored. Consider again a three-way exchange cycle, $C_1 = \langle i, j, k \rangle$; and assume that the false-negative rate of a computer crossmatch test is 20%. We also assume that with probability equal to 50% respectively, patient i and patient j would become really sick and so unavailable for a transplant in the next scheduled match run. Further, assume that with probability equal to 50%, pair k would be able to find a kidney or a compatible match elsewhere outside this particular KPD program and so could also be unavailable. In this setup, where all numbers are chosen for illustrative purpose, we can calculate the probability of edges e_{ij} being viable as $\Pr(X_{e_{ij}} = 1) = 0.8 \times 0.5 \times 0.5 = 0.2$. Clearly, this calculation relies on a reasonable assumption that whether pair i will be ready for a transplant, whether pair j will be ready for a transplant, and the outcome of a laboratory crossmatch test on e_{ij} are independent with each other. Similarly, the probabilities of edges e_{jk} and e_{ki} being viable can be calculated as $\Pr(X_{e_{jk}} = 1) = \Pr(X_{e_{ki}} = 1) = 0.8 \times 0.5 \times 0.5 = 0.2$.

By assuming independence among $X_{e_{ij}}$, $X_{e_{jk}}$, and $X_{e_{ki}}$, we would calculate the probability of cycle C_1 being viable as $0.2^3 = 0.8\%$. Notice, however, that these Bernoulli variables are indeed dependent on one another, since knowing $X_{e_{ij}} = 1$ actually increases $\Pr(X_{e_{jk}} = 1)$ from 0.2 to 0.4, i.e. $\Pr(X_{e_{jk}} = 1 \mid X_{e_{ij}} = 1) = 0.5 \times 0.8 = 0.4$; and similarly $\Pr(X_{e_{ki}} = 1 \mid X_{e_{ij}} = 1) = 0.5 \times 0.8 = 0.4$. Therefore, we should compute the probability that cycle C_1 is viable as

$$\begin{aligned}\Pr(X_{e_{ij}} = 1, X_{e_{jk}} = 1, X_{e_{ki}} = 1) &= \Pr(X_{e_{ij}} = 1) \times \Pr(X_{e_{jk}} = 1 \mid X_{e_{ij}} = 1)\\ &\quad \times \Pr(X_{e_{ki}} = 1 \mid X_{e_{ij}} = 1, X_{e_{jk}} = 1)\\ &= (0.5 \times 0.5 \times 0.8) \times (0.5 \times 0.8) \times 0.8 = 6.4\%,\end{aligned}$$

which is much larger than 0.8%, the number calculated previously under the independence assumption.

It is actually more natural to compute the above probability as

$$\begin{aligned}\Pr(X_{e_{ij}} = 1, X_{e_{jk}} = 1, X_{e_{ki}} = 1) &= \Pr(\{X_{e_{ij}} = 1, X_{e_{jk}} = 1, X_{e_{ki}} = 1\} \cap A)\\ &= \Pr(A) \times \Pr(X_{e_{ij}} = 1, X_{e_{jk}} = 1, X_{e_{ki}} = 1 \mid A),\end{aligned}$$

where A refers to the event "pairs i, j, and k are guaranteed ready for transplant operations in the next scheduled match run"; notice that $\{X_{e_{ij}} = 1, X_{e_{jk}} = 1, X_{e_{ki}} = 1\}$ is a subset of A. It is clear from this interpretation that $\Pr(A)$ measures the uncertainty on the vertices, and $\Pr(X_{e_{ij}} = 1, X_{e_{jk}} = 1, X_{e_{ki}} = 1 \mid A)$ quantifies the uncertainty on the edges. In the above example, it is straightforward to see that $\Pr(A) = 0.5^3$ and $\Pr(X_{e_{ij}} = 1, X_{e_{jk}} = 1, X_{e_{ki}} = 1 \mid A) = 0.8^3$.

4.3.3 Computing expected utilities

In general, for pair i in a KPD program, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let q_i denote the probability that pair i will be ready for a transplant operation in the next scheduled match run, i.e. $\{q_i \mid i \in \mathcal{V}\}$

quantifies the uncertainty on the vertices. For a potential transplant $e = (i, j) \in \mathcal{E}$, we redefine p_e as the conditional probability of having a viable transplant given that pair i and pair j are guaranteed available when this transplant is indeed operated. Therefore, $\{p_e \mid e \in \mathcal{E}\}$ measures the uncertainty on the edges.

An exchange cycle, C, is viable with probability equal to

$$P(C) := \prod_{i \in \mathcal{V}(C)} q_i \times \prod_{e \in \mathcal{E}(C)} p_e,$$

according to this general independence assumption as discussed in Section 4.3.2. Then the expected utility of cycle C can be calculated as $EU(C) = P(C)U(C)$, where $U(C) = \sum_{e \in \mathcal{E}(C)} u_e$ represents the cycle utility. This way of calculating the expected utility of an exchange cycle extends the method we have introduced in Chapter 2; and similarly the maximum expected-utility cycle-based allocation introduced there could be easily generalized by adopting this new way of calculating $EU(C)$.

Next, we see how this method can be extended to compute expected utility of an exchange set. Let $S := (\mathcal{V}(S), \mathcal{E}(S))$ be an exchange set of size $|\mathcal{V}(S)|$. For a subset of $\mathcal{V}(S)$, $\tilde{\mathcal{V}} \subset \mathcal{V}(S)$, define

$$P(\tilde{\mathcal{V}}) := \prod_{i \in \tilde{\mathcal{V}}} q_i \times \prod_{i \in \mathcal{V}(S) \setminus \tilde{\mathcal{V}}} (1 - q_i),$$

which is the probability that pairs in $\tilde{\mathcal{V}}$ will be ready for transplant operations in the next match run while those in $\mathcal{V}(S) \setminus \tilde{\mathcal{V}}$ will not be. Then consider the subgraph of S induced by $\tilde{\mathcal{V}}$, which we denote as $\tilde{S} := (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$. This subgraph basically represents what the original exchange set, S, would turn out to be if the uncertainty on the vertices can be determined beforehand. The expected utility of this subgraph can be calculated by adopting a similar formulation as introduced in Chapter 2. To be precise, for each $\tilde{\mathcal{E}}' \subset \tilde{\mathcal{E}}$ in $\tilde{S} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$, define

$$P(\tilde{\mathcal{E}}') := \prod_{e \in \tilde{\mathcal{E}}'} p_e \times \prod_{e \in \tilde{\mathcal{E}} \setminus \tilde{\mathcal{E}}'} (1 - p_e).$$

For $\tilde{S}' := (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}')$, let $U(\tilde{\mathcal{E}}') := \sum_{C \in \mathcal{C}^*(\tilde{S}')} U(C)$, where $\mathcal{C}^*(\tilde{S}')$ indicates the maximum utility cycle-based allocation on $\tilde{S}'$, and $U(C)$ represents the utility of cycle C. Then, the expected utility of $\tilde{S}$ can be computed as

$$\overline{U}(\tilde{V}) = \sum_{\tilde{\mathcal{E}}' \subset \tilde{\mathcal{E}}} U(\tilde{\mathcal{E}}')P(\tilde{\mathcal{E}}').$$

Finally, it follows that the expected utility of the original exchange set, S, is

$$EU(S) = \sum_{\tilde{V} \subset V} P(\tilde{V})\overline{U}(\tilde{V}). \tag{4.1}$$

We have generalized the method of computing expected utilities for exchange cycles and sets; and this can be done in a very similar way to SCCs as well. For simplicity, we provide below an illustrative example to demonstrate such an extension. Consider an SCC of size 3 as in Figure 4.1, denoted by $S_1 = (\{1,2,3\}, \{e_{12}, e_{21}, e_{13}, e_{31}\})$. In S_1, uncertainties on the vertices are characterized by $q_1 = q_2 = q_3 = 0.5$, and uncertainties on the edges are quantified by $p_{e_{12}} = p_{e_{21}} = p_{e_{13}} = p_{e_{31}} = 0.8$. Further, assume that $u_{12} = u_{21} = 10$ and $u_{13} = u_{31} = 15$. According to formula (4.1), Table 4.1 illustrates the way of computing expected utility associated with S_1.

$\tilde{V}$	$P(\tilde{V})$	$\overline{U}(\tilde{V})$
$\{1,2,3\}$	0.5^3	$30 \times 0.8^2 + 20 \times (1 - 0.8^2) \times 0.8^2$
$\{1,2\}$	0.5^3	20×0.8^2
$\{1,3\}$	0.5^3	30×0.8^2
other subsets of $\{1,2,3\}$	0.5^3	0

Table 4.1: Illustration of computing $EU(S_1)$ for $S_1 = (\{1,2,3\}, \{e_{12}, e_{21}, e_{13}, e_{31}\})$, where $q_1 = q_2 = q_3 = 0.5$, $p_{e_{12}} = p_{e_{21}} = p_{e_{13}} = p_{e_{31}} = 0.8$, and $u_{12} = u_{21} = 10$ and $u_{13} = u_{31} = 15$.

4.4 MDP framework for KPD programs with one altruistic donor

KPD programs with altruistic donors can be more generally analyzed using a Markov decision process (MDP), which provides a rigorous mathematical framework to formulate the problem. In this section, we present this MDP framework for KPD programs with one altruistic donor. The full framework is involved, and we first start with an illustrative example.

4.4.1 An illustrative example

In Figure 4.3, the initial state is taken as the altruistic donor (denoted as vertex 1) and three possible actions, namely “1 to 2”, “1 to 3”, and “1 to 4”. The task at this initial state is to select one action from these three; assume that the action “1 to 2” is chosen. If this chosen action leads to actual completed transplant, the resulting state is then a chain (from 1 to 2) and three new associated actions, namely “2 to 5”, “2 to 6”, and “2 to 7”; if, however, the action “1 to 2” fails to be implemented, the next state would then be defined as an altruistic donor 1 and the other two actions, “1 to 3” and “1 to 4”. Thus, each state in the MDP is specified by a set of completed transplants and a set of currently allowed actions. With the state space so specified, we seek a policy (a mapping from the state space into the action space) that specifies an action to be taken for each possible state. Our goal is to find the optimal policy that gives the maximum expected utility.

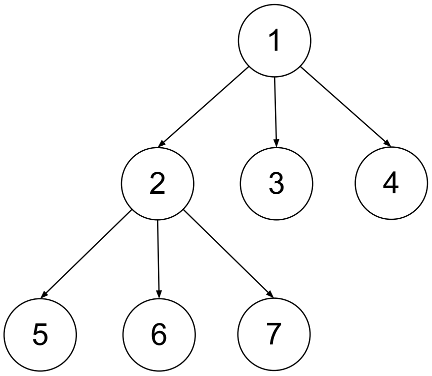

Figure 4.3: An illustrative example with one altruistic donor and six incompatible pairs.

4.4.2 Notation

We adopt the notation introduced in Chapter 3, where a KPD program is represented as a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In this graph, $\mathcal{V} \equiv \mathcal{V}(\mathcal{G}) = \{1, 2, \cdots, n\}$ consists of m altruistic donors, denoted as $\mathcal{V}_a \equiv \mathcal{V}_a(\mathcal{G}) = \{1, 2, \cdots, m\}$, and $n - m$ incompatible pairs, denoted as $\mathcal{V}_p \equiv \mathcal{V}_p(\mathcal{G}) = \mathcal{V} \setminus \mathcal{V}_a$; and $\mathcal{E} \equiv \mathcal{E}(\mathcal{G})$ is a binary relation on $\mathcal{V}$, indicating predicted compatibilities among donors and candidates.

We consider a KPD program with one altruistic donor, i.e. $\mathcal{V}_a = \{1\}$. This naturally implies that $(i, 1) \notin \mathcal{E}$, for all $i \in \mathcal{V}$. For this KPD program, we define an altruistic donor-initiated *path* (of length k, where $k \geq 1$), as a sequence of distinct vertices in $\mathcal{V}$, denoted by

$$\mathbf{h} \equiv \langle h_1, h_2, \cdots, h_k \rangle,$$

such that $h_1 = 1$ and $(h_{j-1}, h_j) \in \mathcal{E}$, for $j = 2, 3, \cdots, k$. Further, let

$$\mathcal{V}(\mathbf{h}) \equiv \{h_1, h_2, \cdots, h_k\}$$

and

$$\mathcal{E}(\mathbf{h}) \equiv \{(h_{j-1}, h_j), j = 2, 3, \cdots, k\}.$$

Note that $\mathcal{E}(\mathbf{h}) = \emptyset$, if $|\mathcal{V}(\mathbf{h})| = 1$.

Let H be the collection of all these altruistic donor-initiated paths in $\mathcal{G}$, and for $\mathbf{h} = \langle h_1, h_2, \cdots, h_k \rangle \in H$, define

$$\mathcal{E}_{\mathbf{h}} \equiv \{(h_k, v) \in \mathcal{E} : v \in \mathcal{V} \setminus \mathcal{V}(\mathbf{h})\}.$$

To better explain these notations, we provide below an illustrative example. Consider a small KPD graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}_a = \{1\}$, $\mathcal{V}_p = \{2, 3\}$, and $\mathcal{E} = \{e_{12}, e_{13}, e_{23}, e_{32}\}$, where, for notational convenience, e_{ij} represents (i, j). In this KPD graph, we notice that

$$H = \{\langle 1 \rangle, \langle 1, 2 \rangle, \langle 1, 2, 3 \rangle, \langle 1, 3 \rangle, \langle 1, 3, 2 \rangle\},$$

and for $\mathbf{h} = \langle 1 \rangle$, we have $\mathcal{V}(\mathbf{h}) = \{1\}$, $\mathcal{E}(\mathbf{h}) = \emptyset$, and $\mathcal{E}_{\mathbf{h}} = \{e_{12}, e_{13}\}$; for $\mathbf{h} = \langle 1, 2 \rangle$, then $\mathcal{V}(\mathbf{h}) = \{1, 2\}$, $\mathcal{E}(\mathbf{h}) = \{e_{12}\}$, and $\mathcal{E}_{\mathbf{h}} = \{e_{23}\}$.

4.4.3 MDP formulation

In this subsection, we propose a MDP framework for KPD programs with one altruistic donor. The *state space* is defined by

$$S \equiv \{(\mathbf{h}, A) : \mathbf{h} \in H, A \subset \mathcal{E}_{\mathbf{h}}\}.$$

Further, we observe that S could be partitioned as

$$S = \cup_{\mathbf{h} \in H} \{(\mathbf{h}, A) : A \subset \mathcal{E}_{\mathbf{h}}\} = \cup_{\mathbf{h} \in H} S_{\mathbf{h}},$$

where $S_{\mathbf{h}} \equiv \{(\mathbf{h}, A) : A \subset \mathcal{E}_{\mathbf{h}}\}$.

For the example mentioned in the last subsection, we have

$$S_{\langle 1\rangle} = \{(\langle 1\rangle, \{e_{12}, e_{13}\}), (\langle 1\rangle, \{e_{12}\}), (\langle 1\rangle, \{e_{13}\}), (\langle 1\rangle, \emptyset)\}.$$

We call $(\mathbf{h}, A) \in S$ an *end state* if $A = \emptyset$; and let

$$S^{\emptyset} \equiv \{(\mathbf{h}, \emptyset) : \mathbf{h} \in H\}$$

be the set of all end states.

For a state $s = (\mathbf{h}, A) \in S \setminus S^{\emptyset}$, where $\mathbf{h} = \langle h_1, h_2, \cdots, h_k\rangle$, an *action* $e_{kk'} \in A$ on s would lead to the next state (i) $s' = (\mathbf{h}', \mathcal{E}_{\mathbf{h}'})$, where $\mathbf{h}' = \langle h_1, \cdots, h_k, h_{k'}\rangle$, if $e_{kk'}$ is viable; or (ii) $s' = (\mathbf{h}, A \setminus \{e_{kk'}\})$, if $e_{kk'}$ is not viable. In this sense, the set A consists of all allowed actions on s. For an end state $s = (\mathbf{h}, \emptyset)$, we define a special "action", denoted as $\emptyset$, which leads to the same state s as the next state. The *action space* is therefore denoted as $\tilde{\mathcal{E}} \equiv \cup_{\mathbf{h}\in H}\mathcal{E}_{\mathbf{h}}$, which is a subset of (not necessarily equal to) $\mathcal{E} \cup \{\emptyset\}$.

A *policy* on managing a KPD program (with one altruistic donor) can then be defined as a mapping,

$$\pi : S \to \tilde{\mathcal{E}},$$

such that

$$\begin{aligned} &\pi(s) \in A \text{ for } s = (\mathbf{h}, A) \in S \setminus S^{\emptyset}, \\ &\pi(s) = \emptyset \text{ for } s \in S^{\emptyset}. \end{aligned}$$

In general terms, $\pi((\mathbf{h}, A))$ specifies what action from A to take for the state $(\mathbf{h}, A)$ in $S \setminus S^{\emptyset}$; for $s \in S^{\emptyset}$, $\pi(s) = \emptyset$ represents the special action that would lead to the same s as the next state. We denote the collection of all policies by Π.

Consider a stochastic process, or a discrete Markov chain in particular, on the state space S, $\{X_n : n \geq 0\}$, with initial state

$$X_0 = s_0 \equiv (\langle 1\rangle, \mathcal{E}_{\langle 1\rangle}).$$

Further, a given policy, π, specifies such a process by specifying its transition probabilities as follows: (i): for a state $s = (\mathbf{h}, A) \in S \setminus S^{\emptyset}$, where $\mathbf{h} = \langle h_1, h_2, \cdots, h_k\rangle$, if $\pi(s) = e_{kk'}$, then for $n \geq 0$,

$$\Pr(X_{n+1} = s' | X_n = s) = p_{kk'}, \text{ for } s' = (\mathbf{h}', \mathcal{E}_{\mathbf{h}'}), \text{ where } \mathbf{h}' = \langle h_1, \cdots, h_k, h_{k'} \rangle;$$
$$\Pr(X_{n+1} = s' | X_n = s) = 1 - p_{kk'}, \text{ for } s' = (\mathbf{h}, A \setminus \{e_{kk'}\}).$$

(ii): for a state $s \in S^{\emptyset}, \Pr(X_{n+1} = s | X_n = s) = 1$, for all $\pi \in \Pi$. We denote this process as $\{X_n^\pi : n \geq 0\}$ when it is necessary to emphasize its dependence on π.

In the previous example, where $X_0 = s_0 = (\langle 1 \rangle, \{e_{12}, e_{13}\})$, if $\pi(s_0) = e_{12}$, then $\Pr(X_1 = s | X_0 = s_0) = p_{12}$, where $s = (\langle 1, 2 \rangle, \{e_{23}\})$, and $\Pr(X_1 = s | X_0 = s_0) = 1 - p_{12}$, where $s = (\langle 1 \rangle, \{e_{13}\})$; if $\pi'(s_0) = e_{13}$, then $\Pr(X_1 = s | X_0 = s_0) = p_{13}$, where $s = (\langle 1, 3 \rangle, \{e_{32}\})$, and $\Pr(X_1 = s | X_0 = s_0) = 1 - p_{13}$ when $s = (\langle 1 \rangle, \{e_{12}\})$.

Let $R(s)$ be the *reward* received at a state $s = (\mathbf{h}, \emptyset) \in S^{\emptyset}$,

$$R(s) \equiv \sum_{e \in \mathcal{E}(\mathbf{h})} u_e.$$

Further, let N^π be the first time the process (specified by $\pi \in \Pi$) is in $S^{\emptyset}$, i.e.

$$N^\pi = \min\{n : X_n^\pi \in S^{\emptyset}\}.$$

The *expected reward* is then defined as

$$ER^\pi = E[R(X_{N^\pi}^\pi)].$$

Finally, the *optimal policy*, π^*, is the one among all policies that gives the largest expected reward, i.e.

$$\pi^* = \operatorname*{argmax}_{\pi \in \Pi} ER^\pi.$$

Solving for the optimal policy is simple for small KPD (or MDP) problems, but is computationally difficult for large or even moderate ones, which makes it difficult to develop practical algorithms based on this MDP framework. This is because the size of the state space grows exponentially with the number of vertices in a KPD graph; see Section 4.6 for more discussion on the utility of this MDP formulation.

4.5 MDP framework for KPD programs with multiple altruistic donors

The MDP framework for a single altruistic donor can be extended to incorporate multiple altruistic donors. This extension closely resembles the approach introduced in the previous section.

We consider a general KPD program $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with multiple altruistic donors. Let $\mathcal{V}_a \equiv \mathcal{V}_a(\mathcal{G}) = \{1, 2, \cdots, m\}$ be the collection of m altruistic donors in $\mathcal{V}$; and $\mathcal{V}_p \equiv \mathcal{V}_p(\mathcal{G}) = \mathcal{V} \setminus \mathcal{V}_a$ represents the set of incompatible pairs. For example, Figure 4.4 lists a KPD program with $\mathcal{V}_a = \{1, 2\}$, $\mathcal{V}_p = \{3, 4\}$ and $\mathcal{E} = \{e_{13}, e_{14}, e_{24}\}$.

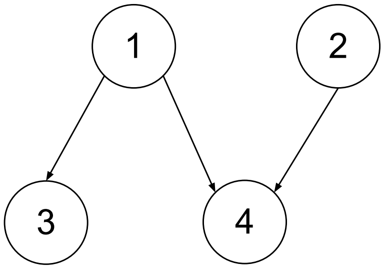

Figure 4.4: A KPD program with two altruistic donors and two incompatible pairs.

As in Section 4.4, let H_i be the collection of all paths initiated by altruistic donor i, where $i = 1, 2, \cdots, m$. Further, define

$$H \equiv \{(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m) \in H_1 \times H_2 \times \cdots \times H_m : \mathcal{V}(\mathbf{h}_i) \cap \mathcal{V}(\mathbf{h}_j) = \emptyset, \forall i \neq j\}.$$

In Figure 4.4, we see that

$$H = \{(\langle 1 \rangle, \langle 2 \rangle), (\langle 1 \rangle, \langle 2, 4 \rangle), (\langle 1, 3 \rangle, \langle 2 \rangle), (\langle 1, 3 \rangle, \langle 2, 4 \rangle), (\langle 1, 4 \rangle, \langle 2 \rangle)\}.$$

For $\mathbf{h} \in H$, let

$$\mathcal{E}_{\mathbf{h}} = \{(v, v') \in \cup_{i=1}^{m} \mathcal{E}_{\mathbf{h}_i} : v' \notin \cup_{i=1}^{m} \mathcal{V}(\mathbf{h}_i)\}.$$

In Figure 4.4, if $\mathbf{h} = (\langle 1 \rangle, \langle 2 \rangle)$, then $\mathcal{E}_{\mathbf{h}} = \{e_{13}, e_{14}, e_{24}\}$; if $\mathbf{h} = (\langle 1, 4 \rangle, \langle 2 \rangle)$, then $\mathcal{E}_{\mathbf{h}} = \emptyset$.

The problem of managing KPD problems with multiple altruistic donors could then be formulated as follows:

The *state space* is defined as

$$S \equiv \{(\mathbf{h}, A) : \mathbf{h} \in H, A \subset \mathcal{E}_{\mathbf{h}}\}.$$

Further, S could be partitioned as

$$S = \cup_{\mathbf{h}\in H}\{(\mathbf{h}, A) : A \subset \mathcal{E}_{\mathbf{h}}\} = \cup_{\mathbf{h}\in H} S_{\mathbf{h}},$$

where $S_{\mathbf{h}} \equiv \{(\mathbf{h}, A) : A \subset \mathcal{E}_{\mathbf{h}}\}$. $(\mathbf{h}, A)$ is an *end state* if $A = \emptyset$; and let $S^{\emptyset} \equiv \{(\mathbf{h}, \emptyset) : \mathbf{h} \in H\}$ be the set of all end states.

For a state $s = (\mathbf{h}, A) \in S \backslash S^{\emptyset}$, where $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m)$, with $\mathbf{h}_1 = \langle h_{11}, \cdots, h_{1k_1}\rangle$, $\mathbf{h}_2 = \langle h_{21}, \cdots, h_{2k_2}\rangle$, $\cdots$, and $\mathbf{h}_m = \langle h_{m1}, \cdots, h_{mk_m}\rangle$, an *action*, $e_{k_j k'_j} \in A$, on s leads to the next state (i) $s' = (\mathbf{h}', \mathcal{E}_{\mathbf{h}'} \cap A)$, where $\mathbf{h}' = (\mathbf{h}_1, \cdots, \mathbf{h}'_j, \cdots, \mathbf{h}_m)$ with $\mathbf{h}'_j = \langle h_{j1}, \cdots, h_{jk_j}, h_{jk'_j}\rangle$, if $e_{k_j k'_j}$ is viable; or (ii) $s' = (\mathbf{h}, A \setminus \{e_{k_j k'_j}\})$, if $e_{k_j k'_j}$ is not viable. In this sense, the set A consists of all allowed actions. For a state $s = (\mathbf{h}, \emptyset)$, we define a special “action”, denoted as $\emptyset$, which leads to the same s as the next state. The *action space* is therefore denoted as $\tilde{\mathcal{E}} \equiv \cup_{\mathbf{h}\in H}\mathcal{E}_{\mathbf{h}}$, which is a subset of (not necessarily equal to) $\mathcal{E} \cup \{\emptyset\}$.

A *policy* on managing a KPD program (with multiple altruistic donors) can then be defined as a mapping,

$$\pi : S \to \tilde{\mathcal{E}},$$

such that

$$\begin{aligned} &\pi(s) \in A \text{ for } s = (\mathbf{h}, A) \in S \setminus S^{\emptyset}, \\ &\pi(s) = \emptyset \text{ for } s \in S^{\emptyset}. \end{aligned}$$

In general terms, $\pi((\mathbf{h}, A))$ specifies what action from A to take for a state $(\mathbf{h}, A) \in S \setminus S^{\emptyset}$. For $s \in S^{\emptyset}$, $\pi(s) = \emptyset$ is the special action mentioned earlier that leads to the same s as the next state. The collection of all policies is denoted by Π.

Consider a stochastic process, or a discrete Markov chain in particular, on the state space S, $\{X_n : n \geq 0\}$, with initial state

$$X_0 = s_0 \equiv (\mathbf{h}, \mathcal{E}_{\mathbf{h}}),$$

where $\mathbf{h} = (\langle 1\rangle, \langle 2\rangle, \cdots, \langle m\rangle)$. Further, a given policy, π, specifies such a process by specifying its transition probabilities as follows: (i): for a state $s = (\mathbf{h}, A) \in S \setminus S^{\emptyset}$, where

$\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m)$, with $\mathbf{h}_1 = \langle h_{11}, \cdots, h_{1k_1}\rangle$, $\mathbf{h}_2 = \langle h_{21}, \cdots, h_{2k_2}\rangle$, $\cdots$, and $\mathbf{h}_m = \langle h_{m1}, \cdots, h_{mk_m}\rangle$, if $\pi(s) = e_{k_j k'_j}$, then for $n \geq 0$, $\Pr(X_{n+1} = s'|X_n = s) = p_{k_j k'_j}$, for $s' = (\mathbf{h}', \mathcal{E}_{\mathbf{h}'} \cap A)$, where $\mathbf{h}' = (\mathbf{h}_1, \cdots, \mathbf{h}'_j, \cdots, \mathbf{h}_m)$ with $\mathbf{h}'_j = \langle h_{j1}, h_{j2}, \cdots, h_{jk_j}, h_{jk'_j}\rangle$; $\Pr(X_{n+1} = s'|X_n = s) = 1 - p_{k_j k'_j}$, for $s' = (\mathbf{h}, A \setminus \{e_{k_j k'_j}\})$. (ii): for a state $s \in S^{\emptyset}, \Pr(X_{n+1} = s|X_n = s) = 1$, for all$\pi \in \Pi$. We represent this process as $\{X_n^\pi : n \geq 0\}$ when it is necessary to emphasize its dependence on π.

Let $R(s)$ be the *reward* received at a state $s = (\mathbf{h}, \emptyset) \in S^{\emptyset}$,

$$R(s) = \sum_{i=1}^{m} \sum_{e \in \mathcal{E}(\mathbf{h}_i)} u_e.$$

Further, let N^π be the first time the process (specified by $\pi \in \Pi$) is in $S^{\emptyset}$, i.e.

$$N^\pi = \min\{n : X_n^\pi \in S^{\emptyset}\}.$$

The *expected reward* is then defined as

$$ER^\pi = E[R(X_{N^\pi}^\pi)].$$

Finally, the optimal policy π^* is the one among all possible policies that gives the maximal expected reward, i.e.

$$\pi^* = \operatorname*{argmax}_{\pi \in \Pi} ER^\pi.$$

We have defined above a particular class of policies that share one common feature: each policy assigns one transplant at one time for only one altruistic donor though multiple altruistic donors are actually available in a KPD program. A natural generalization to consider is a collection of policies that would allow more than one transplant to be planned simultaneously; each transplant involves one altruistic donor. Below, we rigorously define these generalized policies and examine their associated properties. We then demonstrate the relationship between this new class of policies and the old class.

Consider another collection of policies on managing a KPD program, which is defined via

$$\gamma : S \to 2^{\tilde{\mathcal{E}}},$$

such that

$$\begin{aligned}&\gamma(s) \in 2^A \setminus \emptyset \text{ for } s = (\mathbf{h}, A) \in S \setminus S^{\emptyset},\\ &\gamma(s) = \emptyset \text{ for } s \in S^{\emptyset}.\end{aligned}$$

Notice that we require edges in $\gamma(s)$ are "disjoint" for $s \in S$.

In general terms, $\gamma((\mathbf{h}, A))$ specifies what disjoint edges from A to select for $(\mathbf{h}, A) \in S \setminus S^{\emptyset}$. For $s \in S^{\emptyset}$, $\gamma(s) = \emptyset$ is the special action that leads to the same s as the next state. The collection of all such policies is denoted by Γ.

Consider a stochastic process, or a discrete Markov chain in particular, on the state space S, $\{Y_n : n \geq 0\}$, with

$$Y_0 = s_0 \equiv (\mathbf{h}, \mathcal{E}_{\mathbf{h}}),$$

where $\mathbf{h} = (\langle 1\rangle, \langle 2\rangle, \cdots, \langle m\rangle)$.

Further, such a process is specified by $\gamma \in \Gamma$ as follows: (i): for a state $s = (\mathbf{h}, A) \in S \setminus S^{\emptyset}$, where $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m)$, with $\mathbf{h}_1 = \langle h_{11}, h_{12}, \cdots, h_{1k_1}\rangle$, $\mathbf{h}_2 = \langle h_{21}, h_{22}, \cdots, h_{2k_2}\rangle$, $\cdots$, and $\mathbf{h}_m = \langle h_{m1}, h_{m2}, \cdots, h_{mk_m}\rangle$, w.l.o.g., let $\gamma(s) = \{e_{k_j k'_j} : j = 1, 2, \cdots, J\} \subset A$. Consider

$$\begin{aligned}H' \equiv \{(&\mathbf{h}'_1, \cdots, \mathbf{h}'_J, \mathbf{h}_{J+1}, \cdots, \mathbf{h}_m) :\\ &\mathbf{h}'_j = \mathbf{h}_j \text{ or } \langle h_{j1}, h_{j2}, \cdots, h_{jk_j}, h_{jk'_j}\rangle, j = 1, 2, \cdots, J\},\end{aligned}$$

and for $\mathbf{h}' \in H'$, let

$$\begin{aligned}&\delta^j_{\mathbf{h}'} \equiv \mathbf{1}_{[\mathbf{h}'_j \neq \mathbf{h}_j]}, j = 1, 2, \cdots, J,\\ &A_{\mathbf{h}'} \equiv \mathcal{E}_{\mathbf{h}'} \cap A \setminus \{e_{k_j k'_j} : \delta^j_{\mathbf{h}'} = 0\}.\end{aligned}$$

It follows that the set of all possible next states is

$$S' \equiv \{(\mathbf{h}', A_{\mathbf{h}'}) : \mathbf{h}' \in H'\},$$

and

$$\Pr(Y_{n+1} = (\mathbf{h}', A_{\mathbf{h}'}) | Y_n = s) = \prod_{i=1}^{J} (p_{k_j k'_j})^{\delta^j_{\mathbf{h}'}} (1 - p_{k_j k'_j})^{1-\delta^j_{\mathbf{h}'}},$$

$$\text{for } (\mathbf{h}', A_{\mathbf{h}'}) \in S'.$$

(ii): for a state $s \in S^{\emptyset}, \Pr(Y_{n+1} = s | Y_n = s) = 1, \forall \gamma \in \Gamma$.

We represent this process as $\{Y_n^{\gamma} : n \geq 0\}$ when it is necessary to emphasize its dependence on γ.

Then the following result holds: for any $\gamma \in \Gamma$, there always exists at least one $\pi \in \Pi$, such that

$$\begin{aligned} & Y_0^{\gamma} = X_0^{\pi} = s_0 \text{ and} \\ & Y_1^{\gamma} = X_k^{\pi} \text{ a.s. for some } k \leq m \text{ (actually point-wise) .} \end{aligned}$$

Basically, this result informs us that in theory it is enough to only consider policies in Π. To be precise, we have: for any $\gamma \in \Gamma$, there always exists at least one $\pi \in \Pi$, such that

$$Y_{N^{\gamma}}^{\gamma} = X_{N^{\pi}}^{\pi} \text{ a.s.(actually point-wise) .}$$

Solving for the optimal policy in this case is even more complicated than that in the single altruistic donor case. This issue poses a substantial challenge for obtaining the optimal policy within a reasonable amount of computing time for KPD programs even of moderate size, similar to the issue mentioned at the end of Section 4.4. However, near-optimal policies may be obtained to approximately solve this large MDP problem. We will briefly discuss this aspect in Section 4.6.

4.6 Discussion and future directions

In this Chapter, we have proposed an SCC-based approach to arrange organ exchanges among incompatible pairs. This approach builds upon previously introduced set-based and cycle-based allocation strategies — it allows for both a general utility-based evaluation of potential transplants and an explicit incorporation of various operational uncertainties. This SCC-based approach also extends previous approaches — it takes into consideration further additional fall-back options while planning for organ exchanges and hence provides a more robust allocation strategy. This robustness is a desired property in the management of a KPD program because in practice utilities related to medical outcomes usually decrease over time; it is in general better for a transplant to be performed sooner than later.

It is anticipated that the benefit of considering SCCs of size up to three in the planning of organ exchanges would be relatively limited as compared with that of considering exchange sets of size up to three. This is because, as we have mentioned in Section 4.2, there is only one additional SCC structure that is not an exchange set when restricting attention to exchanges among three incompatible pairs. However, if we count SCCs of size four that are not exchange sets, many more structures would arise; for example, Figure 4.5 shows two of them. This observation would imply that a greater amount of gain could be expected when considering SCCs of size up to four in kidney exchanges. To further explore this SCC-based allocation strategy, simulation studies similar to the ones done in Chapter 2 could be conducted. Notice that it makes practical sense in these simulation studies to somehow take into account that a transplant is better to be performed sooner than later.

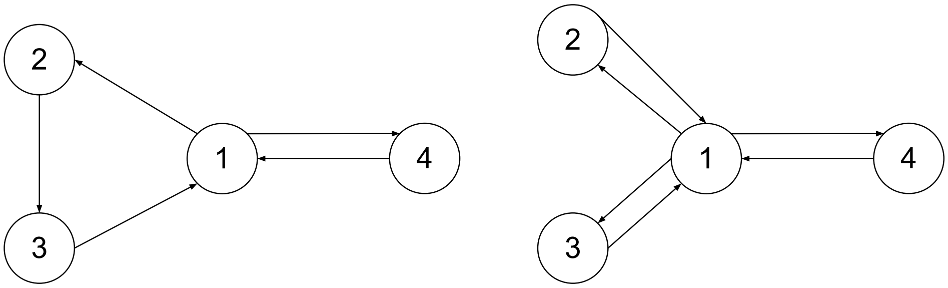

Figure 4.5: Two of the many SCCs of size four that are not exchange sets

In this chapter, we have also proposed to differentiate the operational uncertainties in a KPD program as the ones on the edges and the ones on the vertices. In addition, we have relaxed the independence assumption adopted in previous chapters, and generalized the way of computing expected utilities. In practice, in order to accurately estimate $\{q_i \mid i \in \mathcal{V}(\mathcal{G})\}$, $\{p_e \mid e \in \mathcal{E}(\mathcal{G})\}$ and $\{u_e \mid e \in \mathcal{E}(\mathcal{G})\}$, appropriate statistical models with relevant clinical data from donors and candidates are needed. The development of these models, however, is not the focus of this dissertation. On the other hand, it is worth noting that incorporating these uncertainties on the vertices and edges, even in a rudimentary way, should be encouraged in the management of a KPD program; we would expect a substantial amount of gain by properly assigning even a single average number to $\{q_i \mid i \in \mathcal{V}(\mathcal{G})\}$ and $\{p_e \mid e \in \mathcal{E}(\mathcal{G})\}$ respectively.

We have also built a general MDP framework to rigorously formulate the problem of managing KPD programs with altruistic donors. This MDP formulation was first developed for KPD programs with a single altruistic donor and then generalized to allow multiple altruistic donors. As we have mentioned at the end of Section 4.4 and 4.5, it requires an enormous amount of computational resources to acquire the optimal policy for large KPD programs. This computational challenge poses a serious impediment to the development of practical algorithms based on the MDP framework. In this sense, to manage large KPD programs, we would still recommend the allocation method derived from a depth-k search tree (See Chapter 3). The purpose of this MDP formulation, however, is to consider, from a different perspective, the problem of optimally forming an altruistic donor-initiated chain, and create the link between this optimization problem and the MDP framework. Notice that near-optimal policies can be obtained in the broader context of solving general MDP problems with very large number of states (Kearns et al., 2002; Chang et al., 2005). This line of research is a rich territory, and may shed light on some future development of our work. This direction will not be further explored in this dissertation.

In accessing the management of KPD programs, one criterion is the total number of transplants achieved. This number itself, however, does not reveal the whole picture. For example, suppose that executing match runs once per two months would result in, on average, 20 transplants for each match run, while executing match runs more frequently, at the rate of once per month, could generate about 10 transplants for each run. In this case, both strategies would produce about 60 transplants during the first half year and hence performs equally well according to this criterion. On the other hand, the second strategy of monthly executing match runs shortens the patient's waiting time as compared with the first strategy. Different from utilities associated with potential transplants, which could be outcome based in some centers while rule based in other centers, time to transplant has a natural interpretation and itself is certainly of great interest to both patients and transplant centers.

The investigation of time to transplant therefore provides, from another perspective, the necessity of exploring various other aspects associated with the management of a KPD program, such as how the waiting time is affected by allocation strategies, the percentage of current panel reactive antibody (PRA) values, donor and candidate blood types, and so on. It is also important to understand how the frequency of executing match runs would affect outcomes such as time to transplant. Further, in KPD programs with altruistic donors, their arrival rate certainly would affect time to transplant for those incompatible pairs in the pool; this could be examined in detail via simulation studies by varying the altruistic donor arrival rate and then comparing various outcomes.

Another important issue to consider lies in that some pairs tend to wait in a KPD pool for a relatively long time due to their rare blood types and specific HLA antibodies. One way to address this issue is to increase their corresponding edge utilities proportionally with their waiting time; for instance, the edge utility may get a 20% increase after each match run. Effectiveness of this strategy could then be accessed via simulation studies.

BIBLIOGRAPHY

Abraham, D. J., Blum, A., and Sandholm, T. (2007). Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. *Ec'07: Proceedings of the Eighth Annual Conference on Electronic Commerce*, pages 295–304.

Alagoz, O., Hsu, H., Schaefer, A., and Roberts, M. (2010). Markov decision processes: A tool for sequential decision making under uncertainty. *Medical Decision Making*, 30(4):474–483.

Ashlagi, I., Gilchrist, D. S., Roth, A. E., and Rees, M. A. (2011). Nonsimultaneous chains and dominos in kidney-paired donation - revisited. *American Journal of Transplantation*, 11(5):984–994.

Chang, H. S., Fu, M. C., Marcus, S. I., Chang, H. S., Fu, M. C., and Marcus, S. I. (2005). An adaptive sampling algorithm for solving markov decision processes. *Operations Research*, 53:126–139.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467.

Evans, R. W., Manninen, D. L., Garrison, L. P., Hart, L. G., Blagg, C. R., Gutman, R. A., Hull, A. R., and Lowrie, E. G. (1985). The quality of life of patients with end-stage renal disease. *New England Journal of Medicine*, 312(9):553–559.

Gentry, S. E., Montgomery, R. A., Swihart, B. J., and Segev, D. L. (2009). The roles of dominos and nonsimultaneous chains in kidney paired donation. *American Journal of Transplantation*, 9(6):1330–1336.

Hariharan, S., Johnson, C. P., Bresnahan, B. A., Taranto, S. E., McIntosh, M. J., and Stablein, D. (2000). Improved graft survival after renal transplantation in the united states, 1988 to 1996. *New England Journal of Medicine*, 342(9):605–612.

Kearns, M., Mansour, Y., and Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning*, 49:193–208.

Keizer, K. M., de Klerk, M., Haase-Kromwijk, B. J. J. M., and Weimar, W. (2005). The dutch algorithm for allocation in living donor kidney exchange. *Transplantation Proceedings*, 37(2):589–591.

Laupacis, A., Keown, P., Pus, N., Krueger, H., Ferguson, B., Wong, C., and Muirhead, N. (1996). A study of the quality of life and cost-utility of renal transplantation. *Kidney International*, 50(1):235–242.

Li, Y. J., Kalbfleisch, J. D., Song, P. X., Zhou, Y., Leichtman, A. B., and Rees, M. A. (2011). Optimization and simulation of an evolving kidney paired donation (kpd) program. *The University of Michigan Department of Biostatistics Working Paper Series*, 90.

Maiers, M., Gragert, L., and Klitz, W. (2007). High-resolution hla alleles and haplotypes in the united states population. *Human Immunology*, 68(9):779–788.

Montgomery, R. A., Gentry, S. E., Marks, W. H., Warren, D. S., Hiller, J., Houp, J., Zachary, A. A., Melancon, J. K., Maley, W. R., Rabb, H., Simpkins, C., and Segev, D. L. (2006). Domino paired kidney donation: a strategy to make best use of live non-directed donation. *The Lancet*, 368(9533):419–421.

Park, K., Moon, J. I., Kim, S. I., and Kim, Y. S. (1999). Exchange donor program in kidney transplantation. *Transplantation*, 67(2):336–338.

Rapaport, F. T. (1986). The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proceedings*, 18:5–9.

Rees, M. A., Kopke, J. E., Pelletier, R. P., Segev, D. L., Rutter, M. E., Fabrega, A. J., Rogers, J., Pankewycz, O. G., Hiller, J., Roth, A. E., Sandholm, T., Ünver, M. U., and Montgomery, R. A. (2009). A nonsimultaneous, extended, altruistic-donor chain. *New England Journal of Medicine*, 360(11):1096–1101.

Ross, L. F., Rubin, D. T., Siegler, M., Josephson, M. A., Thistlethwaite, J. R., and Woodle, E. S. (1997). Ethics of a paired-kidney-exchange program. *New England Journal of Medicine*, 336(24):1752–1755.

Roth, A. E., Sönmez, T., and Ünver, M. U. (2004). Kidney exchange. *Quarterly Journal of Economics*, 119(2):457–488.

Roth, A. E., Sönmez, T., and Ünver, M. U. (2005). A kidney exchange clearinghouse in new england. *American Economic Review*, 95(2):376–380.

Roth, A. E., Sönmez, T., and Ünver, M. U. (2007). Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97(3):828–851.

Roth, A. E., Sönmez, T., Ünver, M. U., Delmonico, F. L., and Saidman, S. L. (2006). Utilizing list exchange and nondirected donation through 'chain' paired kidney donations. *American Journal of Transplantation*, 6(11):2694–2705.

Russell, J. D., Beecroft, M. L., Ludwin, D., and Churchill, D. N. (1992). The quality of life in renal transplantation-a prospective study. *Transplantation*, 54(4):656–660.

Schaubel, D. E., Wolfe, R. A., and Port, F. K. (2006). A sequential stratification method for estimating the effect of a time-dependent experimental treatment in observational studies. *Biometrics*, 62(3):910–917.

Schaubel, D. E., Wolfe, R. A., Sima, C. S., and Merion, R. M. (2009). Estimating the effect of a time-dependent treatment by levels of an internal time-dependent covariate: Application to the contrast between liver wait-list and posttransplant mortality. *Journal of the American Statistical Association*, 104(485):49–59.

Segev, D. L., Gentry, S. E., Warren, D. S., Reel, B., and Montgomery, R. A. (2005). Kidney paired donation and optimizing the use of live donor organs. *Journal of the American Medical Association*, 293(15):1883–1890.

Segev, D. L., Kucirka, L. M., Gentry, S. E., and Montgomery, R. A. (2008). Utilization and outcomes of kidney paired donation in the united states. *Transplantation*, 86(4):502–510.

Shapley, L. and Scarf, H. (1974). On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37.

Tambur, A., Ramon, D., Kaufman, D., Friedewald, J., Luo, X., Ho, B., Skaro, A., Caicedo, J., Ladner, D., Baker, T., Fryer, J., Gallon, L., Miller, J., Abecassis, M., and Leventhal, J. (2009). Perception versus reality?: Virtual crossmatch–how to overcome some of the technical and logistic limitations. *American Journal of Transplantation*, 9(8):1886–1893.

Terasaki, P. I., Cecka, J. M., Gjertson, D. W., and Takemoto, S. (1995). High survival rates of kidney-transplants from spousal and living unrelated donors. *New England Journal of Medicine*, 333(6):333–336.

Wolfe, R. A., Ashby, V. B., Milford, E. L., Ojo, A. O., Ettenger, R. E., Agodoa, L. Y. C., Held, P. J., and Port, F. K. (1999). Comparison of mortality in all patients on dialysis, patients on dialysis awaiting transplantation, and recipients of a first cadaveric transplant. *New England Journal of Medicine*, 341(23):1725–1730.

Wolfe, R. A., McCullough, K. P., Schaubel, D. E., Kalbfleisch, J. D., Murray, S., Stegall, M. D., and Leichtman, A. B. (2008). Calculating life years from transplant (lyft): Methods for kidney and kidney-pancreas candidates. *American Journal of Transplantation*, 8(4p2):997–1011.