Improving Policy-Constrained Kidney Exchange via Pre-Screening

Anonymous Author(s)
Affiliation
Address
email

Abstract

In barter exchanges, participants swap goods with one another without exchanging money; these exchanges are often facilitated by a central clearinghouse, with the goal of maximizing the aggregate quality (or number) of swaps. Barter exchanges are subject to many forms of uncertainty–in participant preferences, the feasibility and quality of various swaps, and so on. Our work is motivated by kidney exchange, a real-world barter market in which patients in need of a kidney transplant swap their willing living donors, in order to find a better match. Modern exchanges include 2- and 3-way swaps, making the kidney exchange *clearing problem* NP-hard. Planned transplants often *fail* for a variety of reasons–if the donor organ is rejected by the recipient's medical team, or if the donor and recipient are found to be medically incompatible. Due to 2- and 3-way swaps, failed transplants can "cascade" through an exchange; one US-based exchange estimated that about 85% of planned transplants failed in 2019. Many optimization-based approaches have been designed to avoid these failures; however most exchanges cannot implement these methods, due to legal and policy constraints. Instead, we consider a setting where exchanges can *query* the preferences of certain donors and recipients–asking whether they would accept a particular transplant. We characterize this as a two-stage decision problem, in which the exchange program (a) queries a small number of transplants before committing to a matching, and (b) constructs a matching according to fixed policy. We show that selecting these edges is a challenging combinatorial problem, which is non-monotonic and non-submodular, in addition to being NP-hard. We propose both a greedy heuristic and a Monte Carlo tree search, which outperforms previous approaches, using experiments on both synthetic data and real kidney exchange data from the United Network for Organ Sharing.

1 Introduction

We consider a multi-stage decision problem in which a decision-maker uses a fixed *policy* to solve a hard (stochastic) problem. Before using the policy, the decision-maker can first *measure* some of the uncertain problem parameters–in a sense, guiding the policy toward a better solution. Our primary motivation is kidney exchange, a process where patients in need of a kidney transplant swap their (willing) living donors, in order to find a better match. Many government-run kidney exchanges match patients and donors using a *matching algorithm* that follows strict policy guidelines [8]; this matching algorithm is often written into law or policy, and is not easily modified. Modern kidney exchanges use both cyclical swaps and chain-like structures (initiated by an unpaired altruistic donor) [23], and identifying the max-size or max-weight set of transplants is both NP- and APX-hard [1, 7].

In kidney exchange–as in many resource allocation settings–information used by the decision-maker is subject to various forms of uncertainty. Here we are primarily concerned with uncertainty in the *feasibility* of potential transplants: if a donor is matched with a potential recipient, will the transplant actually occur? Planned transplants may *fail* for a variety of reasons: for example, medical

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

testing may reveal that the donor and recipient are incompatible (a *positive crossmatch*); the recipient or their medical team may reject a donor organ in order to wait for a better match; or the donor may decide to donate elsewhere before the exchange is planned. Failed transplants are especially troublesome in kidney exchange, due to the cycle and chain structures used: for example, suppose that a cyclical swap is planned between three patient/donor pairs; if any one of the planned transplants fails, then none of the other transplants in that cycle can occur. Unfortunately, it is quite common for planned transplants to fail. For example, the United Network for Organ Sharing (UNOS[1]) estimates that in FY2019, about 85% of their planned kidney transplants failed [18].

Various matching algorithms have been proposed that aim to mitigate transplant failures (for example, using stochastic optimization [15, 3], robust optimization [21], or conditional value at risk [6]). However, implementing these strategies would require modifying fielded matching algorithms–which in many cases would require changing law or policy. One way to avoid failures without modifying the matching algorithm is to *pre-screen* potential transplants [18, 9, 10], by communicating with the recipients' medical team and possibly using additional medical tests. Pre-screening transplants is costly, as it requires scarce time and resources. Furthermore, there are often many thousand potential transplants in any given exchange; selecting which transplants to screen is not easy.

In this paper we investigate methods for selecting a limited number of transplants to pre-screen, in order to "guide" the matching algorithm to a better outcome. We formalize this as a multistage stochastic optimization problem, and we consider both an *offline* setting (where screenings are selected all at once), and an *online* setting (where screenings are selected sequentially).

Related Work. While kidney exchange is known to be a hard packing problem, several algorithms exist that are scalable in practice, and are used by fielded exchanges [14, 3, 19]. Prior work has addressed potential transplant failures; our model is inspired by Dickerson et al. [15]. Pre-screening potential transplants has also been addressed in prior work ([10, 22], and § 5.1 of [12]), and our model is similar to stochastic matching and stochastic k-set packing [5]. However there are substantial differences between these models and ours: (a) many prior approaches assume that a large number of transplants may be pre-screened [10, 22]–on the order of one for each patient in the exchange; we assume far fewer screenings are possible; (b) prior work often assumes a *query-commit* setting–where successfully pre-screened transplants *must* be matched. Instead we assume that non-screened transplants may also be matched–which more-accurately represents the way that modern exchanges operate; (c) most prior work assumes that transplants that pass pre-screening are guaranteed to result in a transplant. In reality, transplants often fail after pre-screening, a fact reflected in our model.

One of our approaches is based on *Monte Carlo Tree Search* (MCTS), which allows efficient exploration of intractably large decision trees. While MCTS is primarily associated with Markov decision processes and game-playing [11], it has been used successfully for combinatorial optimization [16]. We use a version of MCTS, Upper Confidence Bounds for Trees (UCT), which balances exploration and exploitation by treating each tree node as a multi-armed bandit problem [4, 17].

Our Contributions

1. (§ 2) We formalize the *policy-constrained edge query problem*: where a decision-maker (such as a kidney exchange program) selects a set of potential edges (potential transplants) to pre-screen, prior to constructing a final packing (a set of transplants) using a fixed algorithm. This model generalizes existing models in the literature, as edge failure probabilities depend on whether or not the edge is pre-screened. Further, we allows for context-specific constraints, such as those imposed by public policy or the particular hospital or exchange.

2. (§ 3) We prove that when the decision-maker uses a max-weight packing policy (the most common choice among fielded exchanges), the edge query problem is both non-monotonic and non-submodular in the set of queried edges. Despite these worst-case findings we show that this problem is nearly monotonic for real and synthetic data, and simple algorithms perform quite well. On the other hand, when the decision-maker uses a *failure-aware* (stochastic) packing policy, the edge query problem becomes monotonic under mild assumptions.

3. (§ 4) We conduct numerical experiments on both simulated and real exchange data from the United Network for Organ Sharing (UNOS). We demonstrate that our methods substantially outperform prior approaches and a randomized baseline.

[1]UNOS is the organization tasked with overseeing organ transplantation in the US: `https://unos.org/`.

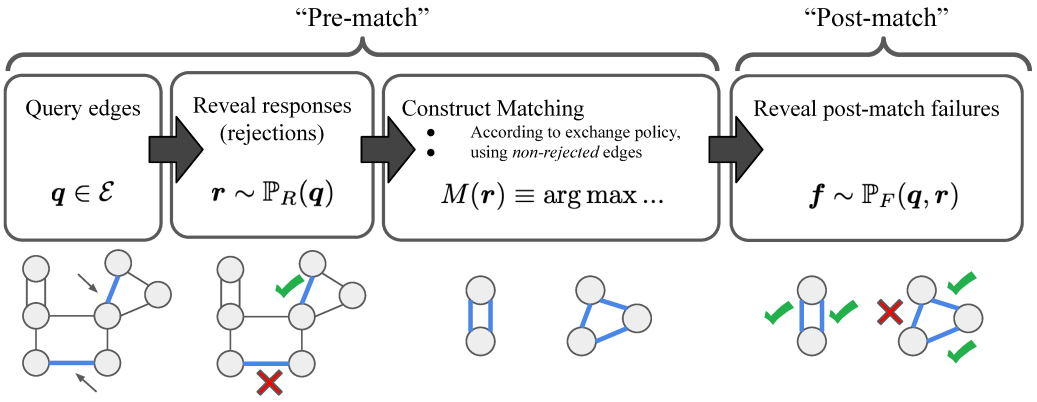


Figure 1: Single-stage edge selection: First, edges are selected to be queried, and responses revealed. Then, a final matching is constructed according to the exchange's matching policy. Finally, the post-match edge failures are revealed.

2 The Policy-Constrained Edge Query Problem

Kidney exchanges are represented by a graph $G = (E, V)$ where vertices V represent (incompatible) patient-donor pairs, and non-directed donors (NDDs) who are willing to donate without receiving a kidney in return. Directed edges $e \in E$ between vertices represent potential transplants from the donor of one vertex to the patient of another. Edge weights represent the "utility" of an edge, and are typically set by exchange policy. Solutions to a kidney exchange problem (henceforth, *matchings*) consist of both directed *cycles* on G containing only patient-donor pairs, and directed *chains* beginning with an NDD and passing through one or more pairs; see Appendix A for an example exchange graph. Each vertex may participate in only one edge in a matching–as each vertex can donate and receive at most one kidney.

Vectors are denoted in bold, and are indexed by either cycles or edges: $\boldsymbol{y}_e$ indicates the element of $\boldsymbol{y}$ corresponding to edge e, and $\boldsymbol{x}_c$ is the element of $\boldsymbol{x}$ corresponding to cycle c. Our notation uses a *cycle-chain* representation for matchings[2]: let $\mathcal{C}$ represent cycles and chains in G, where each cycle and chain corresponds to a list of edges; as is standard in modern exchanges, we assume that cycles and chains are limited in length. Matchings are expressed as a binary vector $\boldsymbol{x} \in \{0,1\}^{|\mathcal{C}|}$, where $\boldsymbol{x}_c = 1$ if cycle/chain c is in the matching, and 0 otherwise. Let $\boldsymbol{w}_c$ be the weight of cycle/chain c (the sum of c's edge weights). Let $\mathcal{M}$ denote the set of *feasible matching*–that is, the set of vertex-disjoint cycles and chains on G. The total weight of a matching is simply the summed weights of all its constituent cycles and chains: $\sum_{c \in \mathcal{C}} \boldsymbol{x}_c \boldsymbol{w}_c$. We denote *sets* of edges using binary vectors, where $\boldsymbol{q} \in \{0,1\}^{|E|}$ represents the set of all edges with $\boldsymbol{q}_e = 1$.

In the remainder of this paper we refer to pre-screening a transplant as *querying an edge*, in order to be consistent with the literature.

Selecting Edge Queries. Our setting consists of two phases (see Figure 1): during *pre-match*, the decision-maker selects edges to query, and each queried edge is either accepted or rejected; then the decision-maker constructs a matching using a fixed policy. During *post-match*, each match edge either fails (no transplant) or succeeds (the transplant proceeds). We consider two version of the pre-match phase: in the *single-stage* version, the decision-maker selects all queries before observing edge responses (accept/reject); in the *multi-stage* version, one edge is selected at a time and responses are observed immediately.

Unlike most prior work, edges in our model may fail during both the pre- and post-match phase. For example, suppose the decision-maker queries an edge from a 60-year-old non-directed donor, to a 35-year-old recipient; if the recipient or their medical team rejects the elderly donor and decides to wait for a younger donor, this is a pre-match rejection. Instead suppose the edge is not queried, and it is included in the final matching; if medical screening reveals that the patient and donor are incompatible, this is a post-match failure. We refer to pre-match failures as *rejections* and post-match failures as *failures*; however we make no assumption about their cause. We represent potential failures

[2]Our experiments use the position-indexed formulation, which is more compact and equivalent [14].

and rejections using binary random variables: $\boldsymbol{r} \in \{0,1\}^{|E|}$ denotes pre-match rejections, where $\boldsymbol{r}_e = 1$ if e is queried and rejected, and 0 otherwise ($\boldsymbol{r}_e = 0$ for all non-queried edges). Similarly $\boldsymbol{f} \in \{0,1\}^{|E|}$ denotes post-match failures, where $\boldsymbol{f}_e = 1$ if edge e fails post-match, and 0 otherwise. We assume that the distribution of rejections $\boldsymbol{r} \sim \mathbb{P}_R(\boldsymbol{q})$ is known, and depends on $\boldsymbol{q}$; we assume the distribution of failures $\boldsymbol{f} \sim \mathbb{P}_F(\boldsymbol{q}, \boldsymbol{r})$ is known, and depends on both $\boldsymbol{q}$ and $\boldsymbol{r}$.

Rejections and failures impact the matching through the *weight* of each cycle and chain. If any cycle edge fails, then *no* transplants in the cycle can proceed; if a chain edge fails, than all edges *following* it cannot proceed.[3] Suppose we observe failures $\boldsymbol{f}$; the *final matching weight* of c is

$$F(c, \boldsymbol{y}) \equiv \begin{cases} \sum_{e \in c} w_e & \text{if } \sum_{e \in c} \boldsymbol{y}_e = 0 \\ 0 & \text{if } c \text{ is a cycle and } \sum_{e \in c} \boldsymbol{y}_e > 0 \\ \sum_{e \in c'} w_e & \text{if } c \text{ is a chain, where } c' \text{ includes all edges up to the first failed edge.} \end{cases}$$

Thus the *post-match expected weight* of matching $\boldsymbol{x}$, due to both rejections $\boldsymbol{r}$ and failures $\boldsymbol{f}$, is

$$W(\boldsymbol{x}; \boldsymbol{q}, \boldsymbol{r}) \equiv \mathop{\mathbb{E}}_{\boldsymbol{f} \sim \mathbb{P}_F(\boldsymbol{q}, \boldsymbol{r})} \left[\sum_{c \in \mathcal{C}} \boldsymbol{x}_c \, F(c, \boldsymbol{r} + \boldsymbol{f}) \right].$$

Matching Policy In this paper we assume that the final matching is constructed using a fixed matching policy, which uses only *non-rejected* edges; we denote this policy by $M(\boldsymbol{r})$. We focus primarily on the *max-weight* policy $M^{\texttt{MAX}}(\cdot)$, which is used by most fielded exchanges, and the *failure-aware* policy $M^{\texttt{FA}}(\cdot)$, which maximizes the expected post-match weight [15]:

$$M^{\texttt{MAX}}(\boldsymbol{r}) \in \mathop{\arg\max}_{\boldsymbol{x} \in \mathcal{M}} \sum_{c \in \mathcal{C}} \boldsymbol{x}_c \, F(c, \boldsymbol{r}) \,, \qquad M^{\texttt{FA}}(\boldsymbol{r}) \in \mathop{\arg\max}_{\boldsymbol{x} \in \mathcal{M}(\boldsymbol{r})} \mathop{\mathbb{E}}_{\boldsymbol{f} \sim \mathbb{P}_F(\boldsymbol{q}, \boldsymbol{r})} \left[\sum_{c \in \mathcal{C}} \boldsymbol{x}_c \, F(c, \boldsymbol{r} + \boldsymbol{f}) \right].$$

Next we formalize the *edge selection problem*–the main focus of this paper. We denote by $\mathcal{E}$ the set of "legal" edge subsets, subject to exchange-specific constraints; we assume that $\mathcal{E}$ is a matroid with ground set E. For example, the decision-maker may limit the number of queries issued to any one medical team (vertex in G) or transplant center (group of vertices). We aim to select an edge set $\boldsymbol{q} \in \mathcal{E}$ which maximizes the *expected weight* of the final matching. These edges are selected using only the distribution of future rejections and failures; we take a *stochastic optimization* approach, maximizing the expected outcome over this uncertainty.

Single-Stage Setting. The single-stage policy-constrained edge selection problem (henceforth, the *edge selection problem*) is expressed as

$$\max_{\boldsymbol{q} \in \mathcal{E}} V^S(\boldsymbol{q}) \,, \qquad \text{with} \quad V^S(\boldsymbol{q}) \equiv \mathop{\mathbb{E}}_{\boldsymbol{r} \sim \mathbb{P}_R(\boldsymbol{q})} \left[\, W\left(M(\boldsymbol{r}); \boldsymbol{q}, \boldsymbol{r}\right) \, \right], \tag{1}$$

where, $M(\boldsymbol{r})$ denotes the matching policy after observing rejections $\boldsymbol{r}$, and $W(\boldsymbol{x}; \boldsymbol{q}, \boldsymbol{r})$ denotes the post-match expected weight of matching $\boldsymbol{x}$. Exact evaluation of $V^S(\boldsymbol{q})$ is often intractable, as the support of $\mathbb{P}_R(\boldsymbol{q})$ grows exponentially in $|\boldsymbol{q}|$. In experiments we approximate $V^S(\boldsymbol{q})$ using sampling, and these approximations converge for a moderate number of samples (see Appendix B).

Multistage Setting. In the multi-stage setting, edge rejections are observed immediately after each edge is queried. The multi-stage problem is expressed as

$$\max_{\boldsymbol{q}^1 \in \mathcal{E}_1} \mathop{\mathbb{E}}_{\boldsymbol{r}^1 \sim \mathbb{P}_R(\boldsymbol{q}^1)} \left[\max_{\boldsymbol{q}^2 \in \mathcal{E}_1} \mathop{\mathbb{E}}_{\boldsymbol{r}^2 \sim \mathbb{P}_R(\boldsymbol{q}^2)} \left[\ldots \max_{\boldsymbol{q}^K \in \mathcal{E}_1} \mathop{\mathbb{E}}_{\boldsymbol{r}^K \sim \mathbb{P}_R(\boldsymbol{q}^K)} \left[\, W\left(M(\boldsymbol{r}); \boldsymbol{q}, \boldsymbol{r}\right) \, \right] \right] \ldots \right], \tag{2}$$

where $\boldsymbol{q} \equiv \sum_{i=1}^K \boldsymbol{q}^i$ denotes all queried edges, $\boldsymbol{r} \equiv \sum_{i=1}^K \boldsymbol{r}^i$ denotes all rejections, and $\mathcal{E}_1 \subseteq \mathcal{E}$ be denotes the legal edge subsets containing only one edge. First, we observe that Problems 1 and 2 require evaluating a matching policy $M(\boldsymbol{r})$. In the case of kidney exchange, evaluating both the max-weight policy $M^{\texttt{MAX}}(\cdot)$ and the failure-aware policy $M^{\texttt{FA}}(\cdot)$ require solving NP-hard problems; thus Problems 1 and 2 are at least NP-hard as well.

However, regardless of matching policy, the question whether *edge selection* is is hard. We observe that while these problems are difficult in principle, experiments (§ 4) show that they are easy in practice. Proofs of the following propositions can be found in Appendix D.

[3] This assumes that chains can be *partially* executed: for example, suppose that the 4^{th} edge in a 10-edge chain fails; the first three edges can still be matched, and the post-failure chain weight sums only these three edges. Not all fielded exchanges use this policy: some exchanges cancel the entire chain if one of its edges fails.

Proposition 2.1. *With matching policy $M^{\texttt{FA}}(\cdot)$, the objective of Problem 1 is non-monotonic in the number of queried edges, even with independent edge distributions.*

In other words, querying additional edges can sometimes lead to a *worse* outcome. This is somewhat counter-intuitive; one might think that providing additional information to the matching policy would strictly improve the outcome. This is a worst-case result–and in fact our experiments demonstrate that querying edges almost always leads to a better final matching weight.

Proposition 2.2. *With matching policy $M^{\texttt{MAX}}(\cdot)$, the objective of Problem 1 is non-submodular in the set of queried edges.*

In other words, certain edges are *complementary* to each other–and querying complementary edges simultaneously can yield a greater improvement than querying them separately. Taken together, these propositions indicate that single-stage edge selection with matching policy $M^{\texttt{MAX}}(\cdot)$ is a challenging combinatorial optimization problem. On the other hand, using the failure-aware matching policy $M^{\texttt{FA}}(\cdot)$ allows us to avoid some of these issues.

Proposition 2.3. *With matching policy $M^{\texttt{FA}}(\cdot)$, and if all edges are independent, the objective of Problem 1 is monotonic in the set of queried edges.*

While Propositions 2.1 and 2.2 state that single-stage edge selection is challenging in the worst case, our computational results suggest that these problems are often easier on realistic exchanges.

3 Solving the Policy-Constrained Edge Query Problem

First we propose an exhaustive tree search which returns an optimal solution to Problem 1 given enough time. Building on this, we propose a Monte Carlo Tree Search algorithm and a simple greedy algorithm. Our multi-stage approaches are very similar to these, and can be found in Appendix E.

Our optimal exhaustive search uses a *search tree* where each tree node corresponds to an edge subset in $\boldsymbol{q} \in \mathcal{E}$. The children of node $\boldsymbol{q}$ correspond to any $\boldsymbol{q}' \in \mathcal{E}$ which are equivalent to the parent $\boldsymbol{q}$, but include one additional edge: $C(\boldsymbol{q}) \equiv \{(\boldsymbol{q} + \boldsymbol{q}') \; \forall \boldsymbol{q}' \in \mathcal{E} : \; |\boldsymbol{q}'| = 1 \;\mid\; (\boldsymbol{q} + \boldsymbol{q}') \in \mathcal{E}\}$. We say that edge sets (or tree nodes) containing L edges are on the L^{th} *level* of the tree. We refer to nodes with no children as *leaf nodes*. Unlike other tree search settings, the optimal solution to Problem 1 may be at *any* node of the tree, not only leaf nodes; this is a consequence of non-monotonicity (see Proposition 2.1). The tree defined by root node $\boldsymbol{q} = \boldsymbol{0}$ and child function $C(\boldsymbol{q})$ contains all legal edge subsets in $\mathcal{E}$, when $\mathcal{E}$ is a matroid. Thus, *any* exhaustive tree search algorithm (such as depth-first search) will identify an optimal solution, given enough time and memory.

Of course exhaustive search is only tractable if $\mathcal{E}$ is small. Consider the class of *budgeted* edge sets $\mathcal{E}(\Gamma)$ used in our experiments: $\mathcal{E}(\Gamma) \equiv \{\boldsymbol{q} \in \{0,1\}^{|E|} \;\mid\; |\boldsymbol{q}| \leq \Gamma\}$ (edge sets containing at most Γ edges). The number of edge sets in $\mathcal{E}(\Gamma)$ grows roughly exponentially in Γ and $|E|$, and is impossible to enumerate even for small graphs. Suppose a graph has 50 edges and we have an edge budget of five: there are over two million edge sets in $\mathcal{E}(5)$. Even small exchange graphs can have thousands of edges, and thus $\mathcal{E}(\Gamma)$ cannot be enumerated. Therefore, we propose search-based approach.

Monte Carlo Tree Search for Edge Selection (`MCTS`): We propose a tree-search algorithm for single-stage edge selection, `MCTS`, based on Monte Carlo Tree Search (MCTS), with the Upper Confidence for Trees (UCT) algorithm [17]. UCT aims to learn the value of tree nodes using repeated sampling and simulation. When the set of tree nodes is too large to enumerate UCT can use a huge amount of memory–by storing values for each visited node. To limit both memory use and runtime, we incrementally search the tree from a temporary root node. Beginning from the root (the the empty edge set), we use UCB sampling on the next L levels of nodes–where L is a small fixed integer. After a fixed time limit, sampling stops and we set the *new* root node to the current root's best child according to its UCB estimate–using the method of [17]. This process repeats until we reach the final level of the search tree. Algorithm 1 gives a pseudocode description of `MCTS`, which uses Algorithm 2 as a submethod. While often successful, MCTS requires extensive training and parameter tuning. As a simpler alternative, we propose a greedy algorithm.

Single-Stage Greedy Algorithm: `Greedy`. Like `MCTS`, our greedy algorithm (`Greedy`) begins with the empty edge set as the root node, and iteratively searches deeper levels of the tree. However unlike `MCTS`, `Greedy` simply selects the child node with the greatest objective value in Problem 1–that is, *greedily* improving the objective value; see Appendix E for a pseudocode description.

ALGORITHM 1: `MCTS`: Tree Search for Single-Stage Edge Selection

(input) K: maximum size of any legal edge set
(input) T: time limit per level
(input) L: number of look-ahead levels

$\boldsymbol{q}^R \leftarrow \mathbf{0}$ root node (no edges)
$\boldsymbol{q}^* \leftarrow \mathbf{0}$ the best visited node
$V^* \leftarrow$ objective value of $\boldsymbol{q}^*$
for $N = 1, \ldots, K$ **do**
  $M \leftarrow \min\{N + L, K\}$
  $Q \leftarrow$ all nodes in levels N to M
  $U[\boldsymbol{q}] \leftarrow 0 \; \forall \boldsymbol{q} \in Q$ UCB value estimate
  $V[\boldsymbol{q}] \leftarrow 0 \; \forall \boldsymbol{q} \in Q$ objective value
  $N[\boldsymbol{q}] \leftarrow 0 \; \forall \boldsymbol{q} \in Q$ number of visits
  while *less than time* T *has passed* **do**
    `Sample`($\boldsymbol{q}^R$, M)
  $\boldsymbol{q}^R \leftarrow \arg\max_{\boldsymbol{q} \in C(\boldsymbol{q}^R)} U[\boldsymbol{q}]$
  Delete $U[\cdot]$, $V[\cdot]$, and $N[\cdot]$
return $\boldsymbol{q}^*$

ALGORITHM 2: `Sample`: Sampling function used by `MCTS`

(input) $\boldsymbol{q}$, M

$N[\boldsymbol{q}] \leftarrow N[\boldsymbol{q}] + 1$
$V[\boldsymbol{q}] \leftarrow$ objective of edge set $\boldsymbol{q}$ in Problem 1
if $V[\boldsymbol{q}] > V^*$ **then**
  $\boldsymbol{q}^* \leftarrow \boldsymbol{q}$, $V^* \leftarrow V[\boldsymbol{q}]$
if $\boldsymbol{q}$ *has no children* **then**
  return $V[\boldsymbol{q}]$
if $\boldsymbol{q}$ *has children* **then**
  if $|\boldsymbol{q}| < M$ **then**
    $\boldsymbol{q}' \leftarrow \arg\max_{\boldsymbol{q} \in C(\boldsymbol{q}^R)} U[\boldsymbol{q}] + \text{UCB}[\boldsymbol{q}]$
    $U[\boldsymbol{q}] \leftarrow U[\boldsymbol{q}] +$ `Sample`($\boldsymbol{q}'$, M)
  else
    $\boldsymbol{q}' \leftarrow$ a random descendent of $\boldsymbol{q}$ at any level
    $V' \leftarrow$ objective value of $\boldsymbol{q}'$ in Problem 1
    if $V' > V^*$ **then**
      $\boldsymbol{q}^* \leftarrow \boldsymbol{q}'$, $V^* \leftarrow V'$
    $U[\boldsymbol{q}] \leftarrow U[\boldsymbol{q}] + V'$

4 Computational Experiments

We conduct a series of computational experiments using both synthetic data, and real kidney exchange data from UNOS; all code for these experiments is available online.[4] In these experiments, "legal" edge sets are the budgeted edge sets defined as $\mathcal{E}(\Gamma) \equiv \{\boldsymbol{q} \in \{0,1\}^{|E|} \mid |\boldsymbol{q}| \leq \Gamma\}$. In Sections 4.1 and 4.2 we present results in the single- and multi-stage edge selection settings, respectively. We use two types of data for these experiments:

Real Data. We use exchange graphs from the United Network for Organ Sharing (UNOS), representing UNOS match runs between 2010 and 2016. Some of these exchange graphs only have the trivial matching (no cycles or chains), or they have only one non-trivial matching. We ignore these graphs because the matching policy is a "constant" function (to return the one feasible matching) and edge queries cannot change the outcome. Removing these, we are left with 240 UNOS exchange graphs.

Synthetic Data. We generate random kidney exchange graphs based on directed Erdős-Rényi graphs defined using parameters N and p: let V be a fixed set of N vertices; for each pair of vertices (V_1, V_2) there is an edge from V_1 to V_2 with probability p, and an edge from V_2 to V_1 with probability p (independent of the edge from V_1 to V_2). Any vertices with no incoming edges are considered NDDs.

In these experiments edge rejections and failures are independently distributed for each edge e; let P_R be the rejection probability, P_Q is the post-match success probability if e is queried/accepted, and P_N is the success probability if e is not queried. To simulate edge rejections and failures we use two synthetic edge distributions: *Simple* and *KPD*. In the *Simple* distribution, $P_R = 0.5$, $P_Q = 1$, and $P_N = 0.5$ for all edges. The *KPD* distribution is inspired by the fielded exchange setting from which we draw our real underlying compatibility graphs. According to UNOS, about 34% of all edges are rejected by a donor or recipient pre-match [18]; we draw P_R uniformly from $U(0.25, 0.43)$ for each edge. Edges ending in highly-sensitized patients (who are often less healthy and more likely to be incompatible) are considered high-risk; for these edges we draw P_Q from $U(0.2, 0.5)$ and P_N from $U(0.0, 0.2)$. For other edges we draw P_Q from $U(0.9, 1.0)$ and P_N from $U(0.8, 0.9)$.

4.1 Single-Stage Edge Selection Experiments

In this section we compare against the baseline of a max-weight matching *without* edge queries (using policy $M^{\texttt{MAX}}(\cdot)$). Let V_X be the objective[5] of Problem 1 achieved by method X, we calculate $\Delta^{\texttt{MAX}}$ (the relative difference from baseline) as $\Delta^{\texttt{MAX}} \equiv (V_X - V^S(\mathbf{0}))/V^S(\mathbf{0})$. A value of $\Delta^{\texttt{MAX}} = 0$ means that method X did not improve over the baseline, a value of $\Delta^{\texttt{MAX}} = 1$ means that X achieved an objective 100% greater than the baseline, and so on.

[4] (link removed during review)

[5] All objective values are estimated using up to 1000 sampled rejection scenarios (see Appendix B), as it is intractable to evaluate the exact objective of large edge sets.

Table 1: Left: Optimality gap for Greedy, over 100 random graphs with $p = 0.01$ and various N, with edge budget $\Gamma = 3$; bottom row shows the maximum value of %OPT over all graphs. Right: Single-stage results on UNOS graphs using the variable IIAB edge budget (top rows), and the failure-aware method (bottom row). Columns P_X indicates the X^{th} percentile of $\Delta^{\texttt{MAX}}$ over all UNOS graphs.

	Num. Graphs (out of 100)		
%OPT	$N = 50$	$N = 75$	$N = 100$
$[0, 0.1]$	93	93	90
$(0, 1]$	5	4	9
$(1, 2]$	1	3	1
$(2, 100]$	1	0	0
Max %OPT	2.8	1.5	1.0

	Simple edge dist.			*KPD* edge dist.		
Method	P_{10}	P_{50}	P_{90}	P_{10}	P_{50}	P_{90}
MCTS	0.40	0.67	1.11	0.05	0.46	3.56
Greedy	0.42	0.72	1.13	0.01	0.49	3.56
Random	0.00	0.10	0.45	−0.11	0.00	0.63
IIAB	0.21	0.45	0.89	-0.24	0.14	2.66
Fail-Aware	0.00	0.09	0.24	-0.25[†]	0.00[†]	2.17[†]

First we investigate the *difficulty* of edge selection. Using random graphs, we compare Greedy to the *optimal* solution to Problem 1, found by exhaustive search (OPT). We generate three sets of 100 random graphs with $N = 50$, 75, and 100 vertices, and each with $p = 0.01$. For all graphs we run both OPT and Greedy with edge budget 3; we calculate the *optimality gap* of Greedy as $\texttt{\%OPT} \equiv 100 \times (V_{\texttt{OPT}} - V_{\texttt{Greedy}})/V_{\texttt{OPT}}$, where V_X denotes the objective achieved by method X. ($V_{\texttt{OPT}} > 0$ in all graphs used in these experiments.) If $\texttt{\%OPT} = 0$ then Greedy returns an optimal solution, and $\texttt{\%OPT} > 0$ means that Greedy is not optimal. Table 1 (left) shows the number of random graphs binned by %OPT, as well as the maximum %OPT over all graphs. For each N, Greedy returns an optimal solution for at least 90 of the 100 graphs; the *maximum* %OPT over all graphs is 2.8.

We also test Greedy on real UNOS graphs, using maximum budget 100. Figure 2a shows the median $\Delta^{\texttt{MAX}}$ over all UNOS graphs, with shading between the 10^{th} and 90^{th} percentiles. Larger edge budgets almost never decrease the objective achieved by Greedy, and Greedy *never* produces a worse outcome than the baseline. Thus–in our setting–single-stage edge selection is effectively monotonic in our setting, and Greedy is an effective method.

Small Edge Budgets on Real UNOS Graphs We compare all methods on UNOS graphs, using smaller, more-realistic edge budgets from 1 to 10. For MCTS we use a 1-hour time limit per edge (Γ hours total). Figures 2b and 2d compare $\Delta^{\texttt{MAX}}$ for MCTS, Greedy, and random edge selection, for the *Simple* and *KPD* edge distributions, respectively. We draw two conclusions from these results: (1) MCTS and Greedy produce almost identical results, further suggesting that Greedy is nearly optimal in our setting; (2) in our setting, edge selection is *effectively* monotonic, as $\Delta^{\texttt{MAX}}$ almost never decreases. However Figure 2d gives an example of non-monotonicity for both Greedy and Random: in some cases, querying edges can lead to a *worse* outcome than querying no edges.

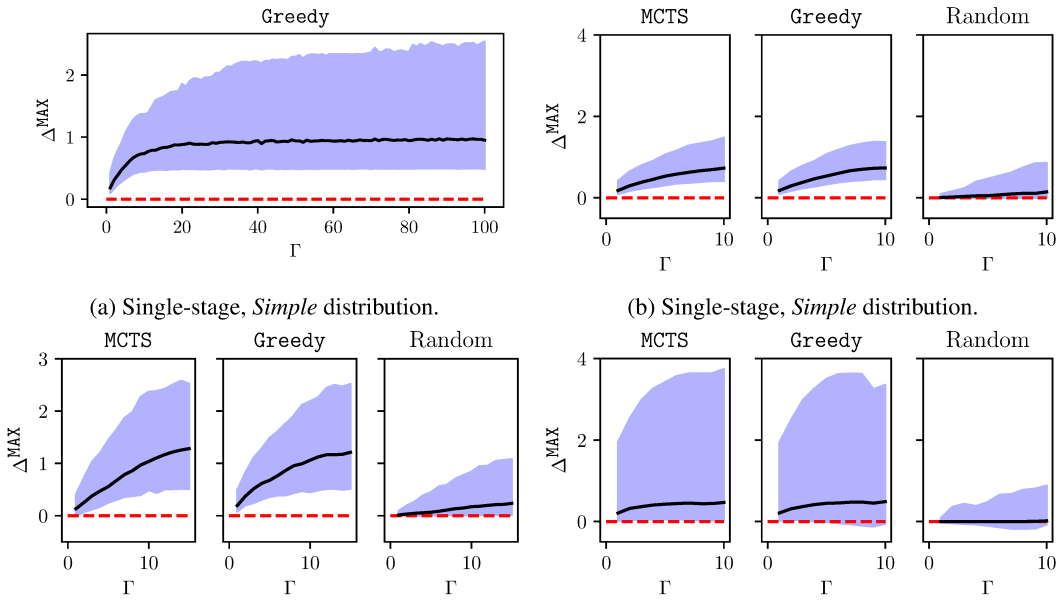
We also compare against two state-of-the-art approaches: the edge selection approach of [10] (IIAB), which uses a *variable* edge budget that depends on the graph structure; and and the failure-aware matching policy of [15] (Fail-Aware), which does not query edges. We for the *KPD* distribution we use an approximation of Fail-Aware, which assumes a uniform edge failure probability. To our knowledge no scalable algorithm exists for the general Fail-Aware method. Table 1 (right) shows a comparison of all edge-selection methods–each using the variable edge budget of IIAB; the bottom row shows results for Fail-Aware.

4.2 Multi-Stage Edge Selection Experiments on UNOS Graphs

We run initial multi-stage edge selection experiments on a subset of 150 randomly chosen UNOS graphs. For each graph we test our multi-stage variants of MCTS and Greedy, and compare with a baseline of random edge selection; as before, MCTS uses a 1-hour training time per level. It is substantially harder to evaluate the multi-stage objective, as each edge edge-selection method changes depending on rejections observed in prior stages. Similarly, the MCTS search tree is orders of magnitude larger in the multi-stage setting: each node in tree corresponds to both an edge set *and* a rejection scenario (see Appendix E).

In these initial experiments we evaluate each method on 10 edge rejections *realizations* (only a small subset). We estimate $\Delta^{\texttt{MAX}}$ for each method and each graph by averaging the final matching weight over all realizations. Figure 2c shows the results of these experiments.

[†]We use an approximation of Fail-Aware for the *KPD* dist.; *true* Fail-Aware should always have $\Delta^{\texttt{MAX}} > 0$.



(a) Single-stage, *Simple* distribution.

(b) Single-stage, *Simple* distribution.

(c) Multi-stage, *Simple* distribution.

(d) Single-stage, *KPD* distribution.

Figure 2: Results for UNOS graphs. Right: edge budget up to 10 for the *Simple* distribution (top) and the *KPD* distribution (bottom). Top-left: `Greedy` with edge budget up to 100, for the simple distribution. Bottom-left: multi-stage methods using the *Simple* distribution. In all plots, a solid line indicates median $\Delta^{\texttt{MAX}}$ over all UNOS graphs, and shading is between the 10^{th} and 90^{th} percentiles; a dotted line indicates the baseline.

These initial multi-stage results are quite similar to our single-stage results. However it is notable that the objective value in the multi-stage setting is somewhat higher than in the single-stage setting–even using the simple method `Greedy`. Further, this suggests that more can be gained by developing a more sophisticated multi-stage edge selection policy. We leave this for future work.

5 Conclusions and Future Research Directions

Many planned kidney exchange transplants *fail* for a variety of reasons; these failures greatly reduce the number of transplants that an exchange can facilitate, and increase the waiting time for many patients in need of a kidney. Avoiding transplant failures is a challenge, as exchanges are often constrained by policy and law in how they match patients and donors. We consider a setting where exchanges can *pre-screen* certain transplants, while still matching patients and donors using a fixed policy. We formalize a multi-stage optimization problem based on realistic assumptions about how transplants fail, and how exchanges match patients and donors; we emphasize that these important assumptions are not included in prior work. While this problem is challenging in theory, we show that it is much easier in practice–with computational experiments using both synthetic data and real data from the United Network for Organ Sharing. In experiments, we find that pre-screening even a small number of potential transplants (around 10) significantly increases the overall quality of the final match–by more than 100% of the original match weight.

Our initial study of the pre-screening problem suggests several areas for future work. First we assume that the distribution of transplant failures is known, when in reality only rough approximations of these distributions are available. Second, we assume that exchange participants (donors, recipients, hospitals) are not strategic. In reality, strategic behavior plays a substantial role in real exchanges [2]; we expect that participants might behave strategically when responding to pre-screening requests. Third, our model does not account for equitable treatment of different patients [20]. For example, it may be the case that pre-screening a transplant decreases the likelihood of the transplant being matched. That might disproportionately impact highly-sensitized patients, which are both sicker and more difficult to match than other patients.

Broader Impact

Patients with end-stage renal disease have only two options: receive a transplant, or undergo dialysis once every few days, for the rest of their lives. In many countries (including the US), these patients register for a deceased donor waiting list–and it can be months or years before they receive a transplant. Many of these patients have a friend or relative willing to donate a kidney, however many patients are incompatible with their corresponding donor. Kidney exchange allows patients to "swap" their incompatible donor, in order to find a *higher-quality* match, *more quickly* than a waiting list. Transplants allow patients a higher quality of life, and cost far less, than lifelong dialysis. About 10% of kidney transplants in the US are facilitated by an exchange.

Our aim in this paper is to understand the impact of transplant pre-screening on a mathematical formulation of kidney exchange. None of our experiments impact actual kidney exchange participants, however the methods we propose for pre-screening transplants could readily be applied by a real exchange, including nascent organ exchanges for other organs such as livers. Thus, we briefly discuss two potential impacts of our methods on a fielded exchange program:

- From a mathematical perspective, we show that pre-screening transplants can–in the worst case–negatively impact the final matching weight (or size) of an exchange (see Proposition 2.1). In other words, pre-screening can result in *fewer* patients receiving kidney transplants, and/or that the resulting transplants are of *lower quality*.
- In kidney exchange, some patients are harder to match than others. These hard-to-match (*highly sensitized*) patients are often sicker, and far less likely to find a compatible kidney donor, than other patients. As a consequence, highly sensitized patients are less likely to be matched by a kidney exchange algorithm [13, 20]; in fact, many exchanges (including the exchange run by UNOS) use policies to prioritize highly sensitized patients so they are not marginalized by a matching algorithm. There is a risk that, by algorithmically selecting transplants for pre-screening, we would further marginalize highly sensitized patients. This risk warrants further study, which we leave for future work.

References

[1] D. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 295–304, 2007.

[2] N. Agarwal, I. Ashlagi, E. Azevedo, C. R. Featherstone, and Ö. Karaduman. Market failure in kidney exchange. *American Economic Review*, 109(11):4026–70, 2019.

[3] R. Anderson, I. Ashlagi, D. Gamarnik, and A. E. Roth. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences*, 112 (3):663–668, 2015.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[5] N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.

[6] H. Bidkhori, J. P. Dickerson, K. Ren, and D. C. McElfresh. Kidney exchange with inhomogeneous edge existence uncertainty. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), forthcoming*, 2020.

[7] P. Biró, D. F. Manlove, and R. Rizzi. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications*, 1(04):499–517, 2009.

[8] P. Biró, J. van de Klundert, D. Manlove, W. Pettersson, T. Andersson, L. Burnapp, P. Chromy, P. Delgado, P. Dworczak, B. Haase, et al. Modelling and optimisation in european kidney exchange programmes. *European Journal of Operational Research*, 2019.

[9] A. Blum, A. Gupta, A. D. Procaccia, and A. Sharma. Harnessing the power of two crossmatches. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 123–140, 2013.

[10] A. Blum, J. P. Dickerson, N. Haghtalab, A. D. Procaccia, T. Sandholm, and A. Sharma. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. *Operations Research*, 2020. Earlier version appeared in the ACM Conference on Economics and Computation (EC), 2015.

[11] B. Bouzy. Associating shallow and selective global tree search with Monte Carlo for $9\times$ 9 Go. In *International Conference on Computers and Games*, pages 67–80. Springer, 2004.

[12] J. P. Dickerson. A unified approach to dynamic matching and barter exchange. Technical report, Doctoral Dissertation, Carnegie Mellon University, Pittsburgh, PA, 2016.

[13] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Price of fairness in kidney exchange. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1013–1020, 2014.

[14] J. P. Dickerson, D. Manlove, B. Plaut, T. Sandholm, and J. Trimble. Position-indexed formulations for kidney exchange. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016.

[15] J. P. Dickerson, A. D. Procaccia, and T. Sandholm. Failure-aware kidney exchange. *Management Science*, 65(4):1768–1791, 2019. Earlier version appeared in the ACM Conference on Economics and Computation (EC), 2013.

[16] B. Kartal, E. Nunes, J. Godoy, and M. Gini. Monte Carlo tree search with branch and bound for multi-robot task allocation. In *The IJCAI-16 workshop on autonomous mobile service robots*, volume 33, 2016.

[17] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[18] R. Leishman. Challenges in match offer acceptance in the optn kidney paired donation pilot program. In *INFORMS Annual Meeting*, 2019. Presentation in Session TB94 - Kidney Allocation & Exchange.

[19] D. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM Journal of Experimental Algorithmics*, 19(1), 2015.

[20] D. C. McElfresh and J. P. Dickerson. Balancing lexicographic fairness and a utilitarian objective with application to kidney exchange. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[21] D. C. McElfresh, H. Bidkhori, and J. P. Dickerson. Scalable robust kidney exchange. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1077–1084, 2019.

[22] M. Molinaro and R. Ravi. Kidney exchanges and the query-commit problem. Manuscript, 2013.

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

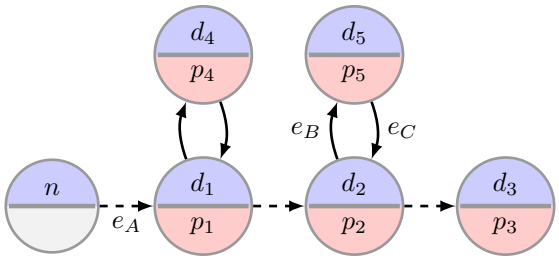


Figure 3: Sample exchange graph with a 3-chain (dashed edges) and two 2-cycles (solid edges). The NDD is denoted by n, and each patient (and associated donor) is denoted by p_i (d_i). If edge e_1 is not queried, or queried and *accepted*, then the chain may be included in the final matching. However if edge e_A is queried and *rejected*, then only the 2-cycles may be included in the final matching.

Edge budgets	$N = 10$	$N = 30$	$N = 50$	$N = 100$	$N = 1000$
1-10	0.0994	0.0568	0.0443	0.0316	0.00978
11-20	0.118	0.0689	0.0533	0.0373	0.0119
21-30	0.131	0.0758	0.0590	0.0418	0.0131
31-40	0.136	0.0787	0.0609	0.0436	0.0135
41-50	0.140	0.0812	0.0627	0.0442	0.014
51-60	0.146	0.0846	0.0653	0.0461	0.0147
61-70	0.152	0.0883	0.0680	0.0481	0.0153
71-80	0.162	0.093	0.0720	0.0514	0.0162
81-90	0.172	0.0995	0.0763	0.0544	0.0172
91-100	0.182	0.104	0.0815	0.0576	0.0180

Table 2: Median normalized standard deviation of the bootstrap mean, over 200 bootstrap samples for each sample size N, binned by edge budget.

A Kidney Exchange

B Estimating The Objective of Problem 1

DCM: @Michael can you review this for accuracy? I added and changed some words, which are hopefully correct

The objective of the single-stage edge selection problem requires evaluating all rejection scenarios $\boldsymbol{r} \sim \mathbb{P}_R(\boldsymbol{q})$, and the support of this distribution grows exponentially in the number of edges $|\boldsymbol{q}|$. In computational experiments, to estimate the objective of Problem 1, we sample up to 1000 scenarios from $\mathbb{P}_R(\boldsymbol{q})$. More explicitly: we *exactly* evaluate the objective of edge sets with fewer than 10 edges; for larger edge sets, we sample the objective using 1000 draws from $\mathbb{P}_R(\boldsymbol{q})$.

Using bootstrapping experiments we demonstrate that our sampling approach is sufficient to accurately estimate the true objective, even for large edge sets. For 152 UNOS graphs, we computed edge sets by running `Greedy` with edge budgets ranging from 1 to 100. For each edge set, we then sample a subset of $N \in \{10, 30, 50, 100, 1000\}$ rejection scenarios, with replacement, from the set of all sampled edge outcomes. For each edge set and choice of N we repeat 200 times and calculate the sample mean for each replication. We then compute the standard deviations of these bootstrap sample means to estimate the variance due to sampling. For each N, we calculate the mean sample standard deviation, normalized by the sample mean. Table 2 shows the median normalized standard deviation for all experiments under each N, with edge budgets aggregated into 10 bins. We find that with $N = 1000$ samples, the standard deviation was on average only about 2% of the overall mean value, even for large edge budgets.

Michael: do we need to cite bootstrapping? this is fairly ad-hoc.

Γ	MCTS			Greedy			Random			IIAB		
	P_{10}	P_{50}	P_{90}	P_{10}	P_{50}	P_{90}	P_{10}	P_{50}	P_{90}	P_{10}	P_{50}	P_{90}
1	**0.10**	**0.18**	**0.42**	**0.10**	**0.18**	**0.42**	0.00	0.01	0.09			
2	**0.17**	**0.30**	**0.65**	**0.17**	0.29	0.64	0.00	0.02	0.16			
3	0.22	0.38	**0.80**	**0.24**	**0.38**	0.79	0.00	0.03	0.23			
4	0.27	0.45	**0.91**	**0.28**	**0.47**	**0.91**	0.00	0.05	0.39			
5	0.31	**0.53**	1.06	**0.33**	**0.53**	**1.07**	0.00	0.05	0.47			
6	0.35	0.58	1.16	**0.38**	**0.60**	**1.18**	0.00	0.07	0.54			
7	0.37	0.63	**1.28**	**0.41**	**0.66**	1.27	0.00	0.09	0.62			
8	0.39	0.67	**1.34**	**0.43**	**0.70**	1.33	0.00	0.11	0.74			
9	0.41	0.69	**1.39**	**0.46**	**0.72**	1.37	0.00	0.11	0.85			
10	0.41	**0.73**	**1.48**	0.46	**0.73**	1.37	0.01	0.15	0.86			
IIAB	0.40	0.67	1.11	**0.42**	**0.72**	**1.13**	0.00	0.10	0.45	0.21	0.45	0.89

Table 3: Single-stage results for the *simple* edge distribution on UNOS graphs: $\Delta^{\texttt{MAX}}$ for each method, and for each edge budget Γ. The 10^{th}, 50^{th} and 90^{th} percentiles of $\Delta^{\texttt{MAX}}$ (P_{10}, P_{50}, and P_{90}) are reported across all UNOS graphs. Maximum value for each Γ is reported in bold.

Γ	MCTS			Greedy			Random			IIAB		
	P_{10}	P_{50}	P_{90}	P_{10}	P_{50}	P_{90}	P_{10}	P_{50}	P_{90}	P_{10}	P_{50}	P_{90}
1	**0.02**	**0.21**	**1.94**	**0.02**	**0.21**	**1.94**	-0.03	0.00	0.08			
2	**0.04**	**0.32**	**2.53**	**0.04**	0.31	2.49	-0.04	0.00	0.36			
3	**0.05**	**0.37**	**2.99**	0.04	**0.37**	**2.99**	-0.06	0.00	0.43			
4	**0.05**	0.41	**3.27**	0.01	**0.42**	**3.27**	-0.07	0.00	0.38			
5	**0.05**	0.43	3.42	-0.01	**0.45**	**3.50**	-0.10	0.00	0.48			
6	**0.05**	0.44	3.57	-0.04	**0.45**	**3.62**	-0.14	0.00	0.66			
7	**0.05**	0.45	**3.63**	-0.07	**0.47**	**3.63**	-0.17	0.00	0.67			
8	**0.03**	0.44	**3.64**	-0.10	**0.48**	3.63	-0.18	0.00	0.78			
9	**0.02**	0.44	**3.64**	-0.12	**0.45**	3.25	-0.17	0.00	0.81			
10	**0.04**	0.47	**3.74**	-0.06	**0.49**	3.36	-0.07	0.02	0.88			
IIAB	**0.05**	0.46	**3.56**	0.01	**0.49**	**3.56**	-0.11	0.00	0.63	-0.24	0.14	2.66

Table 4: Single-stage results for the *KPD* edge distribution on UNOS graphs: $\Delta^{\texttt{MAX}}$ for each method, and for each edge budget Γ. The 10^{th}, 50^{th} and 90^{th} percentiles of $\Delta^{\texttt{MAX}}$ (P_{10}, P_{50}, and P_{90}) are reported across all UNOS graphs. Maximum value for each Γ and each percentile is reported in bold.

C Additional Computational Experiment Results

D Proofs for Section 2

In the proofs of Proposition 2.1 and Proposition 2.2 we consider a setting where all edges' pre-match rejections and post-match failures are i.i.d., where $P_R = 0.5$ is the pre-match rejection probability, $P_Q = 1.0$ is the post-match success probability if the edge is queried-and-accepted, and $P_N = 0.5$ is the success probability if e is not queried. That is, queried edges have rejection probability 0.5, accepted edges have zero failure probability, and non-queried edges have failure probability 0.5.

D.1 Proof of Proposition 2.1

(Proof by counterexample.) We provide an example where querying a single edge results in a *lower* objective value in Problem 1 (i.e., final expected matching weight) than querying no edges–when using the max-weight matching policy $M^{\texttt{MAX}}(\cdot)$.

Consider the exchange graph in Figure 4; edge (E, B) has weight 1.5, while all other edges have weight 1. First we consider the objective due to querying no edges, $V^S(\mathbf{0})$. In this case, no edges can be rejected pre-match, the max-weight matching includes cycle (C, D, F) (expected weight $3 \times (1/2)^3 = 3/8$) and cycle (A, B) (expected weight $2 \times (1/2)^2 = 1/2$), with total expected matching weight $7/8$. That is, $V^S(\mathbf{0}) = 7/8$.

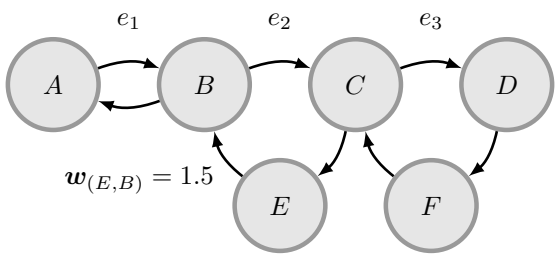


Figure 4: Exchange graph for Propositions 2.1 and 2.2. All edges have weight 1 except for edge (E, B), which has weight 1.5.

Next consider the objective due to querying only edge $e_3 = (C, D)$, and let $\boldsymbol{q}'$ denote edge set $\{e_3\}$. With probability $1/2$, e_3 is rejected and cycle (B, C, E) is the max-weight matching – with expected weight $3.5/8$. With probability $1/2$, e_3 is accepted and the max-weight matching includes cycles (A, B) (with expected weight $1/2$) and (C, D, F) (with expected weight $3/4$); this matching has total expected weight $5/4$. Thus, $V^S(\boldsymbol{q}) = 27/32 < 7/8 = V^S(\mathbf{0})$, which concludes the proof.

D.2 Proof of Proposition 2.2

(Proof by counterexample.) We provide an example where the objective value in Problem 1 (i.e., final expected matching weight) is non-submodular–when using the max-weight matching policy $M^{\texttt{MAX}}(\cdot)$. We use the same rejection and failure distribution as in the proof of Proposition 2.1.

Consider the exchange graph in Figure 4; edge (E, B) has weight 1.5, while all other edges have weight 1. With some abuse of notation, we will denote by $V^S(\{e_a, \ldots, e_N\})$ the objective of Problem 1 due to edge set $\{e_a, \ldots, e_N\}$. Our counterexample for submodularity is that, for this graph,

$$V^S(X \cup \{e_1, e_2\}) + V^S(X) > V^S(X \cup \{e_1\}) + V^S(X \cup \{e_2\}),$$

with set $X \equiv \{e_3\}$. That is, the objective increase due to of querying *both* edges e_1 and e_3 is greater than the combined increase due to querying both edges separately. Next we explicitly calculate each of the above terms.

$V^S(X) = V^S(\{e_3\})$**.** There are two cases to consider:

- e_3 is accepted, with probability $1/2$. The max-weight matching is cycles (A, B) and (C, D, F), with expected weight $(1/2 + 3/4)$,
- e_3 is rejected, with probability $1/2$. The max-weight matching is cycle (B, C, E), with expected weight $3.5/8$.

Thus, $V^S(X) = (1/2)(1/2 + 3/4) + (1/2)(3.5/8) = 27/32$.

$V^S(X \cup \{e_1\}) = V^S(\{e_1, e_3\})$**.** There are four cases to consider:

- e_1 and e_3 are accepted, with probability $1/4$. The max-weight matching is cycles (A, B) and (C, D, F), with expected weight $(1 + 3/8)$,
- e_1 is rejected and e_3 is accepted, with probability $1/4$. The max-weight matching is cycle (B, C, E), with expected weight $3.5/8$.
- e_1 is accepted and e_3 is rejected, with probability $1/4$. The max-weight matching is cycle (B, C, E), with expected weight $3.5/8$.
- e_1 and e_3 are rejected, with probability $1/4$. The max-weight matching is cycle (B, C, E), with expected weight $3.5/8$.

Thus the objective is $V^S(X \cup \{e_3\}) = (1/4)(1 + 3/8) + (3/4)(3.5/8) = 43/64$.

$V^S(X \cup \{e_2\}) = V^S(\{e_2, e_3\})$**.** There are three cases to consider

- e_3 is accepted, with probability $1/2$. The max-weight matching is cycles (A, B) and (C, D, F), with expected weight $(1/2 + 3/4)$,
- e_3 is rejected and e_3 is accepted, with probability $1/4$. The max-weight matching is cycle (B, C, E), with expected weight $3.5/4$,
- e_3 and e_2 are rejected, with probability $1/4$. The max-weight matching is cycle (A, B), with expected weight $1/2$.

Thus the objective is $V^S(X \cup \{e_2\}) = (1/2)(1/2 + 3/4) + (1/4)(3.5/4) + (1/4)(1/2) = 31/32$.

$V^S(X \cup \{e_1, e_2\}) = V^S(\{e_1, e_2, e_3\})$**.** There are four cases to consider:

- e_1 and e_3 are accepted, with probability $1/4$. The max-weight matching is cycles (A, B) and (C, D, F), with expected weight $(1 + 3/4)$,
- e_1 is accepted and e_2 is rejected, with probability $1/4$ (the response from e_3 is irrelevant). The max-weight matching is (A, B) and (C, D, F), with expected weight $1 + 3/8$.
- e_1 is rejected and e_2 is accepted (the response from e_3 is irrelevant), with probability $1/4$. The max-weight matching is cycle (B, C, E), with expected weight $3.5/4$.
- e_1 and e_2 are rejected (the response from e_3 is irrelevant), with probability $1/4$. The max-weight matching is cycle (C, D, F), with expected weight $3/8$.

Thus the objective is $V^S(X \cup \{e_1, e_2\}) = (1/4)(1 + 3/4) + (1/4)(1 + 3/8) + (1/4)(3.5/4) + (1/4)(3/8) = 35/32$.

Finally, we have:

$$\begin{aligned} V^S(X \cup \{e_1, e_2\}) + V^S(X) &= 35/32 + 27/32 \\ &= 1.9375 \end{aligned}$$

and

$$\begin{aligned} V^S(X \cup \{e_1\}) + V^S(X \cup \{e_2\}) &= 43/64 + 31/32 \\ &= 1.640625 \end{aligned}$$

Therefore, $V^S(X \cup \{e_1, e_2\}) + V^S(X) > V^S(X \cup \{e_1\}) + V^S(X \cup \{e_2\})$, which concludes the proof.

D.3 Proof of Proposition 2.3

For the proof of Proposition 2.3 we make one assumption about the distribution of edge rejections and failures: querying *additional* edges cannot increase the overall probability of rejection or failure for any edge.

Assumption D.1. *Let $\boldsymbol{q}, \boldsymbol{r} \in \{0,1\}^{|E|}$ denote initial edge queries and responses. Let $\boldsymbol{q}'$ be* additional *edges, such that $\boldsymbol{q} + \boldsymbol{q}' \in \{0,1\}^{|E|}$ denotes an augmented edge set; let $\boldsymbol{r}' \in \{0,1\}^{|E|}$ denote the responses to edges $\boldsymbol{q}'$ only. We assume that for any such $\boldsymbol{q}$, $\boldsymbol{r}$, and $\boldsymbol{q}'$,*

$$\mathbb{E}\left[\boldsymbol{r} + \boldsymbol{f} \mid \boldsymbol{q}, \boldsymbol{r}\right] \geq \mathbb{E}\left[\boldsymbol{r} + \boldsymbol{r}' + \boldsymbol{f} \mid \boldsymbol{q} + \boldsymbol{q}', \boldsymbol{r}\right] \ .$$

Intuitively, Assumption D.1 excludes distributions where queries arbitrarily increase edge failure or rejection. For example, Assumption D.1 disallows the following distribution: suppose all edges are independent; all queried edges are accepted ($P(\boldsymbol{r}_e = 1 \mid \boldsymbol{q}) = 0$ for all $\boldsymbol{q}$), all accepted edges have failure probability 0.5 ($P(\boldsymbol{f}_e = 1 \mid \boldsymbol{q}_e = 1, \boldsymbol{r}_e = 0) = 0.5$), and all non-queried edges have failure probability 0.1 ($P(\boldsymbol{f}_e = 1 \mid \boldsymbol{q}_e = \boldsymbol{r}_e = 0) = 0.1$). In this case, if an edge is not queried, then it has overall rejection or failure probability 0.1 (i.e., $\mathbb{E}[\boldsymbol{r}_e + \boldsymbol{f}_e \mid \boldsymbol{q}, \boldsymbol{r}] = 0.1$ with $\boldsymbol{q}_e = 0$); if this edge is queried, then it has rejection or failure probability 0.5 (i.e., $\mathbb{E}[\boldsymbol{r}_e + \boldsymbol{r}'_e + \boldsymbol{f}_e \mid \boldsymbol{q} + \boldsymbol{q}', \boldsymbol{r}] = 0.5$ with $\boldsymbol{q}'_e = 1$).

First we prove a handful of useful results.

Definition D.2 (Edge Independence)**.** *Two edges $e, e' \in E$ are* independent *if (a) their rejection distributions are conditionally independent, given whether or not they were queried:*

$$\boldsymbol{r}_e \perp\!\!\!\perp \boldsymbol{r}_{e'} \mid \boldsymbol{q}_e \quad \textit{and} \quad \boldsymbol{r}_e \perp\!\!\!\perp \boldsymbol{r}_{e'} \mid \boldsymbol{q}_{e'}$$

and (b) their failure distributions are conditionally independent, given whether or not they were queried and rejected:

$$\boldsymbol{f}_e \perp\!\!\!\perp \boldsymbol{f}_{e'} \mid \boldsymbol{q}_e, \boldsymbol{r}_e \quad \textit{and} \quad \boldsymbol{f}_e \perp\!\!\!\perp \boldsymbol{f}_{e'} \mid \boldsymbol{q}_{e'}, \boldsymbol{r}_{e'} \ .$$

Lemma D.3. *If all edges are independent, then additional edge queries cannot decrease expected post-match cycle and chain weights. Formally,*

$$\mathbb{E}\left[F(c, \boldsymbol{r} + \boldsymbol{f}) \mid \boldsymbol{q}, \boldsymbol{r}\right] \leq \mathbb{E}\left[F(c, \boldsymbol{r} + \boldsymbol{r}' + \boldsymbol{f}) \mid \boldsymbol{q} + \boldsymbol{q}', \boldsymbol{r}\right]$$

for any $\boldsymbol{q}, \boldsymbol{q}' \in \{0,1\}^{|E|}$ such that $\boldsymbol{q} + \boldsymbol{q}' \in \{0,1\}^{|E|}$, for any $\boldsymbol{r} \in \{0,1\}^{|E|}$, and for all $c \in \mathcal{C}$.

Proof. We address cycles and chains separately.

Cycles. Conditional on fixed $\boldsymbol{q}$ and $\boldsymbol{r}$, the expected weight of cycle $c = (e_1, \ldots, e_L)$ is expressed as

$$\begin{aligned}\mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{f}) \mid \boldsymbol{q}, \boldsymbol{r}\right] &= \left(\sum_{e\in c} w_e\right) \mathbb{E}\left[\prod_{e\in c}(1-\boldsymbol{r}_e-\boldsymbol{f}_e) \mid \boldsymbol{q}, \boldsymbol{r}\right] \\ &= \left(\sum_{e\in c} w_e\right) \prod_{e\in c}\left(1-\mathbb{E}\left[\boldsymbol{r}_e+\boldsymbol{f}_e \mid \boldsymbol{q}, \boldsymbol{r}\right]\right)\end{aligned}$$

where the second step is due to the fact that all $\boldsymbol{f}_e$ are independent. Similarly, for fixed $\boldsymbol{q}'$,

$$\mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f}) \mid \boldsymbol{q}+\boldsymbol{q}', \boldsymbol{r}\right] = \left(\sum_{e\in c} w_e\right) \prod_{e\in c}\left(1-\mathbb{E}\left[\boldsymbol{r}_e+\boldsymbol{r}'_e+\boldsymbol{f}_e \mid \boldsymbol{q}+\boldsymbol{q}', \boldsymbol{r}\right]\right) .$$

Due to Assumption D.1, the following inequality holds for all edges $e \in E$

$$\mathbb{E}\left[\boldsymbol{r}_e+\boldsymbol{f}_e \mid \boldsymbol{q}, \boldsymbol{r}\right] \geq \mathbb{E}\left[\boldsymbol{r}_e+\boldsymbol{r}'_e+\boldsymbol{f}_e \mid \boldsymbol{q}+\boldsymbol{q}', \boldsymbol{r}\right] ,$$

and it follows that

$$\mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{f}) \mid \boldsymbol{q}, \boldsymbol{r}\right] \leq \mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f}) \mid \boldsymbol{q}+\boldsymbol{q}', \boldsymbol{r}\right].$$

Chains. Similarly, the expected weight of chain $c = (e_1, \ldots, e_L)$ is expressed as

$$\begin{aligned}\mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{f}) \mid \boldsymbol{q}, \boldsymbol{r}\right] &= \sum_{k=1}^{L}\left(\sum_{j=1}^{k} w_j\right) \mathbb{E}\left[\prod_{j=1}^{k}(1-\boldsymbol{r}_{e_j}-\boldsymbol{f}_{e_j}) \mid \boldsymbol{q}, \boldsymbol{r}\right] \\ &= \sum_{k=1}^{L}\left(\sum_{j=1}^{k} w_j\right) \prod_{j=1}^{k}\left(1-\mathbb{E}\left[\boldsymbol{r}_{e_j}+\boldsymbol{f}_{e_j} \mid \boldsymbol{q}, \boldsymbol{r}\right]\right) ,\end{aligned}$$

where the second step is due to the fact that $\boldsymbol{f}_e$ are independent. Similarly,

$$\mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f}) \mid \boldsymbol{q}+\boldsymbol{q}', \boldsymbol{r}\right] = \sum_{k=1}^{L}\left(\sum_{j=1}^{k} w_j\right) \prod_{j=1}^{k}\left(1-\mathbb{E}\left[\boldsymbol{r}_{e_j}+\boldsymbol{r}'_{e_j}+\boldsymbol{f}_{e_j} \mid \boldsymbol{q}+\boldsymbol{q}', \boldsymbol{r}\right]\right) .$$

as before, due to Assumption D.1 it follows that

$$\mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{f}) \mid \boldsymbol{q}, \boldsymbol{r}\right] \leq \mathbb{E}\left[F(c, \boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f}) \mid \boldsymbol{q}+\boldsymbol{q}', \boldsymbol{r}\right].$$

□

Lemma D.4. *With a failure-aware matching policy, and if all edges are independent, adding a single edge to any edge query set weakly improves the objective of Problem 1. Formally, for any $\boldsymbol{q}, \boldsymbol{q}' \in \{0,1\}^{|E|}$ with $\boldsymbol{q}+\boldsymbol{q}' \in \{0,1\}^{|E|}$ and $|\boldsymbol{q}'| = 1$, and $M(\boldsymbol{r}) \equiv M^{\texttt{FA}}(\boldsymbol{r})$,*

$$V^S(\boldsymbol{q}) \leq V^S(\boldsymbol{q}+\boldsymbol{q}')$$

Proof. The objective of Problem 1 for edge set $\boldsymbol{q}$ is expressed as

$$\begin{aligned}V^S(\boldsymbol{q}) &= \mathop{\mathbb{E}}_{\boldsymbol{r}|\boldsymbol{q}}\left[\mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q},\boldsymbol{r}}\left[\sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r}) F(c, \boldsymbol{r}+\boldsymbol{f})\right]\right] \\ &= \sum_{\boldsymbol{r}\in\{0,1\}^{|\boldsymbol{q}|}} P_{\boldsymbol{q}}(\boldsymbol{r}) \mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q},\boldsymbol{r}}\left[\sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r}) F(c, \boldsymbol{r}+\boldsymbol{f})\right] \\ &= \sum_{\boldsymbol{r}\in\{0,1\}^{|\boldsymbol{q}|}} P_{\boldsymbol{q}}(\boldsymbol{r}) \sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r}) \mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q},\boldsymbol{r}}\left[F(c, \boldsymbol{r}+\boldsymbol{f})\right]\end{aligned}$$

For edge set $\boldsymbol{q}+\boldsymbol{q}'$ we partition response variables into $\boldsymbol{r}, \boldsymbol{r}' \in \{0,1\}^{|E|}$, where $\boldsymbol{r}_e$ is the response variable for all edges $e \in \boldsymbol{q}$, and $\boldsymbol{r}_e = 0$ for all other edges (including the edge in $\boldsymbol{q}'$). Similarly, $\boldsymbol{r}'_e$ is

the response variable for edge $\boldsymbol{q}'$, and $\boldsymbol{r}'_e = 0$ for all other edges. The objective of $\boldsymbol{q} + \boldsymbol{q}'$ is expressed as

$$\begin{aligned} V^S(\boldsymbol{q}+\boldsymbol{q}') &= \mathop{\mathbb{E}}_{\boldsymbol{r},\boldsymbol{r}'|\boldsymbol{q}+\boldsymbol{q}'}\left[\mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q}+\boldsymbol{q}',\boldsymbol{r}+\boldsymbol{r}'}\left[\sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r}+\boldsymbol{r}')F(c,\boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f})\right]\right] \\ &= \sum_{\boldsymbol{r}\in\{0,1\}^{|\boldsymbol{q}|}} P_{\boldsymbol{q}+\boldsymbol{q}'}(\boldsymbol{r}) \mathop{\mathbb{E}}_{\boldsymbol{r}'|\boldsymbol{q}+\boldsymbol{q}'}\left[\mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q}+\boldsymbol{q}',\boldsymbol{r}+\boldsymbol{r}'}\left[\sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r}+\boldsymbol{r}')^\top F(c,\boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f})\right]\right] \\ &= \sum_{\boldsymbol{r}\in\{0,1\}^{|\boldsymbol{q}|}} P_{\boldsymbol{q}}(\boldsymbol{r}) \mathop{\mathbb{E}}_{\boldsymbol{r}'|\boldsymbol{q}+\boldsymbol{q}'}\left[\mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q}+\boldsymbol{q}',\boldsymbol{r}+\boldsymbol{r}'}\left[\sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r}+\boldsymbol{r}')F(c,\boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f})\right]\right], \end{aligned}$$

where in the final line we replace $P_{\boldsymbol{q}+\boldsymbol{q}'}(\boldsymbol{r})$ with $P_{\boldsymbol{q}}(\boldsymbol{r})$, because each $\boldsymbol{r}_e$ is conditionally independent, given $\boldsymbol{q}_e$.

Next, by definition

$$\mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q}+\boldsymbol{q}',\boldsymbol{r}+\boldsymbol{r}'}\left[\sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r}+\boldsymbol{r}')F(c,\boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f})\right] \geq \mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q}+\boldsymbol{q}',\boldsymbol{r}+\boldsymbol{r}'}\left[\sum_{c\in\mathcal{C}} \boldsymbol{x}_c F(c,\boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f})\right] \quad \forall \boldsymbol{x}\in\mathcal{M}.$$

That is, M^{FA} is guaranteed to maximize this expectation, and thus

$$\begin{aligned} V^S(\boldsymbol{q}+\boldsymbol{q}') &\geq \sum_{\boldsymbol{r}\in\{0,1\}^{|\boldsymbol{q}|}} P_{\boldsymbol{q}}(\boldsymbol{r}) \mathop{\mathbb{E}}_{\boldsymbol{r}'|\boldsymbol{q}+\boldsymbol{q}'}\left[\mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q}+\boldsymbol{q}',\boldsymbol{r}+\boldsymbol{r}'}\left[\sum_{c\in\mathcal{C}} M_c^{\texttt{FA}}(\boldsymbol{r})F(c,\boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f})\right]\right] \quad \text{(B)} \\ &= \sum_{\boldsymbol{r}\in\{0,1\}^{|\boldsymbol{q}|}} P_{\boldsymbol{q}}(\boldsymbol{r}) \sum_{c\in\mathcal{C}} M_v^{\texttt{FA}}(\boldsymbol{r}) \mathop{\mathbb{E}}_{\boldsymbol{r}'|\boldsymbol{q}+\boldsymbol{q}'}\left[\mathop{\mathbb{E}}_{\boldsymbol{f}|\boldsymbol{q}+\boldsymbol{q}',\boldsymbol{r}+\boldsymbol{r}'}\left[F(c,\boldsymbol{r}+\boldsymbol{r}'+\boldsymbol{f})\right]\right] \quad \text{(C)} \end{aligned}$$

Finally, combining (B) and (C) with Lemma D.3, the following inequality holds

$$V^S(\boldsymbol{q}) \leq V^S(\boldsymbol{q}+\boldsymbol{q}').$$

□

Using the above lemmas, the proof of Proposition 2.3 is straightforward:

Proposition 2.3 *With a failure-aware matching policy, if all edges are independent, the objective of Problem 1 is monotonic in the set of queried edges.*

Proof. Let $\boldsymbol{q}', \boldsymbol{q}'' \in \mathcal{E}$ be two edge sets such that $\boldsymbol{q}' \subseteq \boldsymbol{q}''$. It remains to show that, with matching policy $M(\boldsymbol{r}) \equiv M^{\texttt{FA}}(\boldsymbol{r})$,

$$V^S(\boldsymbol{q}'') \leq V^S(\boldsymbol{q}').$$

First note that because $\mathcal{E}$ is a matroid, there is a sequence of edges $(\boldsymbol{q}^{e_1}, \ldots, \boldsymbol{q}^{e_L})$ (with each $|\boldsymbol{q}^{e_i}| = 1$) such that $\boldsymbol{q}'' + \boldsymbol{q}^{e_1} + \cdots + \boldsymbol{q}^{e_L} = \boldsymbol{q}'$. Due to Lemma D.4, the following sequence of inequalities hold:

$$\begin{aligned} V(\boldsymbol{q}'') &\leq V(\boldsymbol{q}'' + \boldsymbol{q}^{e_1}) \\ &\leq V(\boldsymbol{q}'' + \boldsymbol{q}^{e_1} + \boldsymbol{q}^{e_2}) \\ &\cdots \\ &\leq V(\boldsymbol{q}'' + \boldsymbol{q}^{e_1} + \cdots + \boldsymbol{q}^{e_L}) \\ &= V(\boldsymbol{q}') \end{aligned}$$

which concludes the proof. □

E Algorithm Descriptions

Here we describe more explicitly the algorithms for `Greedy` and `MCTS`, for both the single-stage and multi-stage settings.

ALGORITHM 3: `Greedy`: Greedy Search Heuristic for Single-Stage Edge Selection

(input) $\mathcal{E}$: legal edge sets

$\boldsymbol{q}^R \leftarrow \mathbf{0}$ the root node (no edges)
$V^* \leftarrow$ objective value of $\boldsymbol{q}^R$ Problem 1
while $\boldsymbol{q}^R$ *has children* **do**
  $\boldsymbol{q}' \leftarrow$ child node of $\boldsymbol{q}^R$ with maximal objective value in Problem 1
  $\boldsymbol{q}^R \leftarrow \boldsymbol{q}'$
return $\boldsymbol{q}^R$

E.1 Algorithms for Edge Selection

Algorithm 3 gives a pseudocode description of `Greedy` for the single-stage setting.

E.2 Multi-Stage Edge Selection

Multi-Stage `MCTS`. The multi-stage version of `MCTS` differs from the single-stage version in that each node of the search tree corresponds to both a set of queried edges and a set of observed rejections.

ALGORITHM 4: Multi-Stage `MCTS`: Tree Search Heuristic for Multi-Stage Edge Selection

TODO: this has not been updated yet!!! (input) $\mathcal{E}$: legal edge sets
(input) K: maximum size of any legal edge set
(input) T: time limit per level
(input) L: number of look-ahead levels

$\boldsymbol{q}^R \leftarrow \mathbf{0}$ root node (no edges)
$\boldsymbol{q}^* \leftarrow \mathbf{0}$ the best visited node
$V^* \leftarrow$ objective value of $\boldsymbol{q}^*$
for $N = 1, \ldots, K$ **do**
  $M \leftarrow \min\{N + L, K\}$
  $Q \leftarrow$ all nodes in levels N to M
  $U[\boldsymbol{q}] \leftarrow 0\ \forall \boldsymbol{q} \in Q$ UCB value estimate
  $V[\boldsymbol{q}] \leftarrow 0\ \forall \boldsymbol{q} \in Q$ objective value
  $N[\boldsymbol{q}] \leftarrow 0\ \forall \boldsymbol{q} \in Q$ number of visits
  while *less than time* T *has passed* **do**
    `Sample`$(\boldsymbol{q}^R, M)$
  $\boldsymbol{q}^R \leftarrow \arg\max_{\boldsymbol{q} \in C(\boldsymbol{q}^R)} U[\boldsymbol{q}]$
  Delete $U[\cdot]$, $V[\cdot]$, and $N[\cdot]$
return $\boldsymbol{q}^*$

ALGORITHM 5: `MultiSample`: Sampling function used by Multi-Stage `MCTS`

TODO: this has not been updated yet!!!
(input) $\boldsymbol{q}$: tree node
(input) M: maximum level to sample from
$N[\boldsymbol{q}] \leftarrow N[\boldsymbol{q}] + 1$
$V[\boldsymbol{q}] \leftarrow$ objective of edge set $\boldsymbol{q}$ in Problem 1
if $V[\boldsymbol{q}] > V^*$ **then**
  $\boldsymbol{q}^* \leftarrow \boldsymbol{q}$, $V^* \leftarrow V[\boldsymbol{q}]$
if $\boldsymbol{q}$ *has no children* **then**
  return $V[\boldsymbol{q}]$
if $\boldsymbol{q}$ *has children* **then**
  if $|\boldsymbol{q}| < M$ **then**
    $\boldsymbol{q}' \leftarrow \arg\max_{\boldsymbol{q} \in C(\boldsymbol{q}^R)} U[\boldsymbol{q}]$
    $U[\boldsymbol{q}] \leftarrow U[\boldsymbol{q}] +$ `Sample`$(\boldsymbol{q}', M)$
  else
    $\boldsymbol{q}' \leftarrow$ a random descendent of $\boldsymbol{q}$ at any level
    $V' \leftarrow$ objective value of $\boldsymbol{q}'$ in Problem 1
    if $V' > V^*$ **then**
      $\boldsymbol{q}^* \leftarrow \boldsymbol{q}'$, $V^* \leftarrow V'$
    $U[\boldsymbol{q}] \leftarrow U[\boldsymbol{q}] + V'$

Multi-Stage `Greedy`. Algorithm 6 gives a pseudocode description of the multi-stage version of `Greedy`. This search heuristic returns the next edge to query with the highest expected final matching weight, *ignoring all future queries*. In other words, this approach treats every edge as the *last* edge; one might call this heuristic "myopic" as well as greedy.

ALGORITHM 6: `Greedy` Heuristic for Multi-Stage Edge Selection

(input) $\mathcal{E}$: legal edge sets
(input) $\boldsymbol{q}$: previously-queried edges
(input) $\boldsymbol{r}$: previously-observed rejections

$e^* \leftarrow \emptyset$ $V^* \leftarrow 0$
for *all* $\boldsymbol{q}'$ *in* $\boldsymbol{q}$*'s children* **do**
    $e' \leftarrow$ the new edge queried in child node $\boldsymbol{q}'$ $\boldsymbol{r}^A \leftarrow \boldsymbol{r}$
    $\boldsymbol{r}^R \leftarrow \boldsymbol{r}$
    $\boldsymbol{r}_e^A \leftarrow 1$ (response scenario where e' is accepted)
    $\boldsymbol{r}_e^R \leftarrow 1$ (response scenario where e' is rejected)
    $p^A \leftarrow$ probability that e is accepted, conditional on $\boldsymbol{r}$
    $p^R \leftarrow$ probability that e is accepted, conditional on $\boldsymbol{r}$
    $V' \leftarrow p^A \cdot W(M(\boldsymbol{r}^A); \boldsymbol{q}', \boldsymbol{r}^A) + p^R W(M(\boldsymbol{r}^R); \boldsymbol{q}', \boldsymbol{q}^R)$ (value of querying edge e')
    if $V' > V^*$ **then**
        $\boldsymbol{e}^* \leftarrow e'$
        $V^* \leftarrow V'$
return $\boldsymbol{e}^*$