Loss-Constrained Minimum Cost Flow under Arc Failure Uncertainty with Applications in Risk-Aware Kidney Exchange

Qipeng P. Zheng*  Siqian Shen†  Yuhui Shi‡

November 16, 2014

Abstract

In this paper, we study a stochastic minimum cost flow (SMCF) problem under arc failure uncertainty, where an arc flow solution may correspond to multiple path flow representations. We assume that the failure of an arc will cause flow losses on all paths using that arc, and for any path carrying positive flows, the failure of any arc on the path will lose all flows carried by the path. We formulate two SMCF variants to minimize the cost of arc flows, while respectively restricting the value-at-risk (VaR) and conditional value-at-risk (CVaR) of random path flow losses due to uncertain arc failure (reflected as network topological changes). We formulate a linear program to compute possible losses, yielding a mixed-integer programming formulation of SMCF-VaR and a linear programming formulation of SMCF-CVaR. We present a kidney exchange problem under uncertain match failure as an application, and use the two SMCF models to maximize the utility/social welfare of pairing kidneys subject to constrained risk of utility losses. Our results show the efficacy of our approaches, the conservatism of using CVaR, and optimal flow patterns given by VaR and CVaR models on diverse instances.

Key words: stochastic minimum cost flow, value-at-risk (VaR), conditional value-at-risk (CVaR), Benders decomposition, risk-aware kidney exchange

1 Introduction

The minimum cost flow (MCF) problem is fundamental to many classical network flow problems (Ahuja et al. 1993). Consider a directed graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a set of all nodes, and $\mathcal{A}$ is a set of all arcs. Set $\mathcal{S} \subset \mathcal{N}$ and set $\mathcal{T} \subset \mathcal{N}$ respectively contain supply and demand nodes. Denote an absolute value of supply or demand at node $i$ by $D_i$ for all $i \in \mathcal{N}$, a unit flow cost on

*Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL 32816; email: qipeng.zheng@ucf.edu;
†Corresponding author; Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109; email: siqian@umich.edu;
‡Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI
arc \((i,j)\) by \(C_{ij}\), and a capacity of arc \((i,j)\) by \(U_{ij}\), for all \((i,j)\) \(\in\) \(A\). Let decision variables \(x_{ij}\) be the amounts of flows on arcs \((i,j)\) \(\in\) \(A\). We formulate a generic MCF problem as

\[
[MCF]: \min \sum_{(i,j) \in A} C_{ij} x_{ij} \tag{1a}
\]

s.t.

\[
\sum_{j: (i,j) \in A} x_{ij} - \sum_{j: (j,i) \in A} x_{ji} = \begin{cases} 
D_i & \forall i \in S, \\
0 & \forall i \in N \setminus S \setminus T, \\
-D_i & \forall i \in T, 
\end{cases} \tag{1b}
\]

\[
0 \leq x_{ij} \leq U_{ij}, \quad \forall (i,j) \in A, \tag{1c}
\]

where the objective (1a) is to minimize the total flow cost, constraints (1b) ensure flow balance at each node, and constraints (1c) bound flows on arcs by their capacities.

In this paper we consider the stochastic minimum cost flow problem (SMCF), in which arcs may randomly fail given some known probabilities. The goal is to minimize the arc flow cost, while the losses of path flows are restricted by two types of risk measures under the uncertainty of 0-1 arc failure. Such network flows can represent traffic, product shipments, or telecommunication signals in a wide class of network applications (see, e.g., Shen and Chen 2013, Shen and Smith 2013). The problem is of great importance to the design and operations of critical infrastructures (Shen 2013). In particular, we let arc flows be fixed before knowing realizations of random arc failure. The problems are designed for applications where one needs to determine flow assignments \textit{a priori} in an uncertain network environment and cannot reroute flows once the uncertainty is realized.

### 1.1 Motivation and Problem Description

We investigate two SMCF problem variants, where we seek a feasible arc flow solution yielding the minimum cost subject to constrained risk of flow losses. We justify our problem settings in applications that typically involve (i) rapid changes of a network environment, (ii) extremely short reaction time for resource delivery and network recovery, or both (i) and (ii). For example, a shipment schedule of relief resources is needed right after the occurrence of a catastrophic event, when detailed information of road failure and traffic congestion is not available. Or a cyber-attacker interferes wireless communications and causes data losses among web users, while an operator seeks a fixed data routing plan to bound the risk of significant data losses against possible bandwidth disruptions.

In this paper, we assume that (i) the failure of an arc will cause flow losses on all paths using that arc, and (ii) for any path carrying positive flows, the failure of one or multiple arcs on the path will lead to losing the whole amount of flows it carries. The two assumptions take into account situations where flows are routed via paths and losses are incurred when the paths are disconnected.

In a related work, Boginski et al. (2009) study an SMCF variant under uncertain arc failure where they calculate flow losses as a summation of flows on all failed arcs and restricted the
conditional value-at-risk (CVaR) of the total losses. We consider network flow problems where flows are routed via paths between supply and demand nodes, and failed arcs will cause path disconnections and losses of path flows. Therefore, the summation of flows on failed arcs may not accurately reflect actual flow losses in the corresponding operations in practice. It is demonstrated later that the amount of arc flow losses used in Boginski et al. (2009) provides a conservative upper bound of the path flow losses considered in this paper, and the gap between the two depends on how many failed arcs exist in each path with positive flows.

Moreover, an arc flow solution possibly correspond to several path flow representations, yielding multiple possible values of flow losses. In this paper, we restrict the value-at-risk (VaR) and CVaR of any possible values of flow losses. The total loss is computed through a linear program formulated on a residual network of the original graph, given an arc flow solution and a realization of the uncertain arc failure. We show that the two SMCF variants are respectively equivalent to bounding the VaR and CVaR of the minimum value of flow losses among all possible representations using path flows. For finite scenarios of random arc failure, we formulate the VaR- and CVaR-based SMCF variants as a mixed-integer program and a linear program, respectively. We develop decomposition and cutting-plane algorithms for solving the two formulations. Our models and approaches are demonstrated on a class of kidney exchange problems, where the objective is to maximize the utility/social welfare of pairing kidneys subject to constrained risk of utility losses due to uncertain match failure of paired kidneys. We test a set of network instances to demonstrate the results and insights of VaR and CVaR variants of the SMCF problem. We also apply our approaches to the kidney exchange problems and characterize exchange solutions that are node-disjoint cycles while using different probability/risk measures.

1.2 Literature Review

SMCF problems have been extensively studied due to their close relationship with a variety of network applications involving data uncertainties. For instance, Loui (1983), Eiger et al. (1985), Fan et al. (2005), Hutson and Shier (2009) consider variants of a stochastic shortest path problem, as special cases of SMCF, where a decision maker seeks a path from an origin to a destination and minimizes either the expected value or a weighted sum of mean and variance of the random path length under uncertain travel time (e.g., Boyles and Waller 2010). Cheung and Powell (1996), Powell and Cheung (1994), Glockner and Nemhauser (2000), Glockner et al. (2001) analyze stochastic dynamic network flow problems under uncertain arc capacities and developed formulations based on large-scale time-expanded networks. In their models, network flows are recourse decisions determined after knowing the uncertainty, which are also used in the “last-mile delivery” problems of seeking dynamic flows according to network changes in humanitarian relief applications (e.g., Balcik et al. 2008, Salmerón and Apte 2010, Özdamar et al. 2004). Also by defining recourse flows, Prékopa and Boros (1991) compute the probability of the existence of feasible flows in a stochas-
tic transportation network with random demands and arc capabilities, while Hudson and Kapur (1983), Lin (1998), Lin (2007) study similar reliability problems of assigning feasible flows to meet uncertain demands given a budget of the total flow cost. In this paper, we minimize the cost of fixed arc flows subject to sufficiently low risk of flow losses evaluated by two risk measures with respect to uncertain arc failure in a network. Specifically, we use VaR and CVaR (cf. Rockafellar and Uryasev 2000, 2002) to bound possible losses of path flows. The former measures exact probabilities of random outcomes, corresponding to the study of chance-constrained programming (cf. Miller and Wagner 1965, Prékopa 1970) and its proliferating literature. The latter, also called “average value-at-risk” or “expected shortfall” (see, e.g., Chun et al. 2012), is a coherent risk measure that has been mainly and widely used for risk management in financial problems. The two measures have been compared on a variety of applications in the literature (e.g., Alexander and Baptista 2004). We acknowledge that there are other options of risk measures, especially coherent risk measures such as the entropic value-at-risk (EVaR) (Ahmadi-Javid 2012, Föllmer and Knispel 2011) that corresponds to the tightest possible upper bound obtained from the Chernoff inequality for VaR and CVaR. These coherent risk measures normally yield formulations that are computationally tractable but could yield solutions that are more conservative than the VaR measure. In this paper we use CVaR as one of popular coherent risk measures and as a comparison to VaR for analyzing the SMCF problem.

1.3 Contributions and Organization of the Paper

The contributions of the paper are summarized as follows. First, given any arc flow solution and an arc failure scenario, we formulate a linear programming (LP) model to compute flow losses of the corresponding path flow solutions. This results in an LP reformulation of the CVaR variant, and maintains the computational complexity of the VaR variant solved by integer programming techniques. We also apply our approaches to address a class of risk-aware kidney exchange problems under uncertain match failure, arising in state-of-the-art applications of SMCF. Second, we test a variety of SMCF instances to demonstrate the computational efficacy of our approach that computes path flow losses by using linear constraints, compared with direct reformulations using path flow representation variables. Third, we demonstrate the CVaR’s solution conservatism via calculating risk values associated with the corresponding VaR on all instances, and evaluate flow losses more accurately by using path flows so that we avoid duplicating losses of arc flows carried by the same path. Fourth, our results show the importance of trading off the use of paths with high travel cost and paths containing vulnerable arcs (i.e., arcs with relatively high failure rates). Moreover, optimal solutions to both VaR and CVaR variants intend to split flows to multiple paths with the minimum number of arcs shared by these paths.

The remainder of the paper is organized as follows. In Section 2, given an arc flow solution to the MCF, we formulate an LP model to compute flow losses in each arc failure scenario. In addition, we
develop polynomial-time algorithms for computing the maximum and minimum losses. Section 3 and Section 4 describe models and algorithms of the VaR and CVaR variants of SMCF, respectively, both of which impose risk constraints on the minimum possible flow losses. Section 5 demonstrates procedures of formulating a class of kidney exchange problems as SMCF variants under match failure uncertainty. Section 6 presents computational results by testing randomly generated diverse instances, and examines the tradeoff between solution robustness and the minimum flow cost. We conclude the paper and state future research directions in Section 7.

2 Computing Path Flow Losses in Polynomial Time

We first develop models and algorithms for computing losses of path flows for a given arc flow solution and a realized scenario of arc failure. We evaluate the total loss of an arc flow solution as the summation of flows on all paths that have arc failure. According to the flow representation theorem (Ahuja et al. 1993), an arc flow solution may correspond to multiple flow representations using paths and cycles. Given \( C_{ij} > 0 \), \( \forall (i,j) \in A \), we focus on only path flow solutions for representing any given arc flow solution, because a flow representation with positive cycle flows will yield a higher cost as well as a higher chance of having bigger flow losses and thus will not be optimal. The total flow losses corresponding to the same arc flow solution might vary, depending on which paths we consider to convey positive flows. Consider an example depicted in Figure 1, in which we route four units of flow from node 1 to node 8, with arc flow solutions being indicated along the side of each arc. Consider a scenario where both arcs (2, 3) and (6, 7) failed. Then two units of losses are incurred if we transport two units of flow via path “\( P_1: 1–2–3–4–5–6–7–8 \),” and the other two units via path “\( P_2: 1–4–5–8 \).” However, all four units of flow will be completely gone if we ship two units of flow via path “\( P_3: 1–2–3–4–5–8 \),” and the other two units via path “\( P_4: 1–4–5–6–7–8 \).”

![Figure 1: Illustrating multiple values of path flow losses yielded by one arc flow solution.](image)

Given random scenarios of arc failure, the corresponding maximum and minimum losses of all possible path flow representations are also random. For a fixed arc flow solution and an arc failure scenario, Section 2.1 formulates an LP for computing all possible losses, and Section 2.2 develops polynomial algorithms for computing the maximum and the minimum flow losses.
2.1 A Linear Programming Model for Computing Flow Losses

Given an arc flow solution $\hat{x} = [\hat{x}_{ij}, (i,j) \in A]^T$, we construct a reformulated residual network (RRN) $G(\hat{x}) = (N', A')$ with a new arc set $A'$, for which we delete all arcs in $A$ are currently with zero flow, create a new arc $(i,j)$ in $A'$ for each arc $(j,i) \in A$ that has $\hat{x}_{ji} > 0$, and designate $\hat{x}_{ji}$ as the capacity of the new arc.

![Figure 2: A path $s \rightarrow t$ with positive flows in an MCF solution.](image2.png)

Figure 2: A path $s \rightarrow t$ with positive flows in an MCF solution.

![Figure 3: The RRN $G(\hat{x})$ of path $s \rightarrow t$ illustrated in Figure 2.](image3.png)

Figure 3: The RRN $G(\hat{x})$ of path $s \rightarrow t$ illustrated in Figure 2.

Figure 2 presents a path flow solution, where nodes $s$ and $t$ are the origin and destination, straight arcs represent arcs (e.g., $(i,j)$ and $(k,l)$), and curvy arcs represent paths (e.g., $s \rightsquigarrow i$). For given $\hat{x}$ on path $s \rightarrow t$, we build $G(\hat{x})$ by deleting all zero-flow arcs in $A$ and reversing all positive-flow arcs in $A$. Figure 3 illustrates the corresponding RRN $G(\hat{x})$ for path $s \rightarrow t$ in Figure 2, given that flows are all positive in the path.

Let parameter $Y_{ij}$ denote the status of arc $(i,j) \in A$, such that

$$Y_{ij} = \begin{cases} 
1 & \text{if arc } (i,j) \in A \text{ fails}, \\
0 & \text{otherwise}.
\end{cases}$$

Consider network $G(\hat{x}, Y)$, where $Y = [Y_{ij}, (i,j) \in A]^T$, as a modification of $G(\hat{x})$, in which we delete all arcs $(i,j)$ that have $Y_{ij} = 1$. Furthermore, for any nodes $i$ and $j$ with $Y_{ij} = 1$, we add an additional fixed supply $\hat{x}_{ij}$ at node $i$, and an additional variable demand $\rho_j$ at node $j$. We also add a demand variable $\lambda_s$ as the accumulated withdrawn of flows at node $s$. Figure 4 illustrates the corresponding $G(\hat{x}, Y)$ for the $G(\hat{x})$ in Figure 3 given $Y_{ij} = Y_{kl} = 1$.

![Figure 4: The RRN $G(\hat{x}, Y)$ of path $s \rightarrow t$ in Figure 2 given $Y_{ij} = Y_{kl} = 1$](image4.png)
Given an MCF solution $x$ and parameter $Y$, we use the RRN $\mathcal{G}(x,Y)$ to compute all possible values of the total flow withdrawn (i.e., flow losses) in $\mathcal{G}$ due to arc failure. Define variable $f_{ij}$ for each arc $(i,j) \in \overline{A}$, which equals to the withdrawn flow on arc $(j,i) \in A$. Recall that set $S$ contains all supply nodes in $\mathcal{G}$. The following LP model maintains flow balances at all nodes in $\mathcal{G}(x,Y)$:

$$\sum_{j: (i,j) \in \overline{A}} f_{ij} - \sum_{j: (j,i) \in \overline{A}} f_{ji} = \begin{cases} -\lambda_i + \sum_{j: (i,j) \in A} Y_{ij}x_{ij} - \rho_i & \forall i \in S \\ \sum_{j: (i,j) \in A} Y_{ij}x_{ij} - \rho_i, & \forall i \in N \setminus S \end{cases} \quad (2a)$$

$$0 \leq f_{ij} \leq (1 - Y_{ji})x_{ji} \quad \forall (i,j) \in \overline{A} \quad (2b)$$

$$0 \leq \lambda_i \leq D_i \quad \forall i \in S \quad (2c)$$

$$0 \leq \rho_i \leq \sum_{j: (j,i) \in A} Y_{ji}x_{ji} \quad \forall i \in N. \quad (2d)$$

Constraints (2a) balance flows at all the nodes in set $S$ and all other nodes in $N \setminus S$, where the latter do not accumulate withdrawn path flows reflected by the values of $\lambda_i$, $\forall i \in S$. Constraints (2b) ensure $f_{ij} = 0$ for any arc $(i,j) \in \overline{A}$ if arc $(j,i) \in A$ fails in $\mathcal{G}$, and restrict $f_{ij}$ on other arcs $(i,j) \in \overline{A}$ by the values of $x_{ji}$ otherwise. The accumulated value of flow losses at any node $i \in S$ is bounded by its original supply $D_i$ according to (2c), while (2d) bound the values of variable demands $\rho_i$, $\forall i \in N$ by the total amount of flow inputs on failed arcs adjacent to node $i$ in $A$.

Let $L(x,Y)$ be the amount of flow losses given by a path flow representation of solution $x$ and uncertain parameter $Y$. The following theorem relates $L(x,Y)$ with the LP formulation (2).

**Theorem 1.** Given a feasible flow solution $x$ and parameter $Y$, $L(x,Y) = \sum_{i \in S} \lambda_i$, where $\lambda = [\lambda_i, \ i \in S]^T$ is a feasible solution to Formulation (2).

**Proof.** Given a feasible arc flow $x$, according to the flow representation theorem, we can decompose it into $O(m)$ distinct paths each of which has a positive flow and connects an excess node with a deficit node. The RRN $\mathcal{G}(x,Y)$ represents a flow withdrawn pattern, where by eliminating arcs with no flows (i.e., arcs $(i,j) \in A$ with $x_{ij} = 0$) and reversing arcs with positive flows, we ensure that the withdrawn flows only traverse arcs in those paths carrying positive flows in an opposite direction of the original path flow direction. For each node $i \in N \setminus S$, according to (2a), the original outgoing flows on its failed adjacent arcs are accumulated (i.e., $\sum_{j: (i,j) \in A} Y_{ij}x_{ij}$) and subtracted by the amount of variable demand $\rho_i$ (created if $Y_{ij} = 1$ exists for any $(i,j) \in A$), to generate a flow-withdrawn supply at node $i$. For any nodes in $N \setminus S$ that do not have any failed adjacent arcs, the associated flow withdrawn inputs and outputs are balanced as the right-hand sides of constraints (2a) are zero. Constraints (2b) forbid any withdrawn flows traversing failed arcs and also restrict the amount of withdrawn flows on any arc by its original flow amount. Therefore, variable $\rho_i$ in (2a) will carry the difference of accumulated flow withdrawn and the output capacities at each node $i \in N$, while its value as the amount of withdrawn “consumed” by node $i$ will be no more than the total amount of failed flow inputs in the original graph. Finally, the accumulated amount of
flow losses on each path is taken by variable $\lambda_i$ at origin $i$ of the corresponding path according to (2a), for all $i \in S$, and is bounded by the total supply $D_i$ according to (2c). The summation of all positive withdrawn flows $\lambda_i$ at all nodes $i \in S$ yields the total flow losses $L(x, Y)$, which completes the proof.

**Remark 1.** According to Theorem 1, the total amount of path flow losses is always bounded above by the summation of flows on failed arcs, i.e., $L(x, Y) \leq \sum_{(i,j) \in A} Y_{ij}x_{ij}$. This is because variables $\lambda_i$ are only linked to the failed arcs that are adjacent to nodes $i \in S$ in (2a). The difference between $L(x, Y)$ and $\sum_{(i,j) \in A} Y_{ij}x_{ij}$ will become larger if more failed arcs are present in the same path with a positive flow assignment. Note that the latter is another measure of flow losses used in the literature (e.g., Boginski et al. 2009) when flows are not considered to be generated via paths. Our approach uses a less conservative means of measuring flow losses and offers ways of calculating exact path flow losses in related applications.

As previously illustrated in Figure 1, an arc flow solution $x$ may correspond to multiple path flow representations, and thus might lead to different $L(x, Y)$-values when we solve Formulation (2) even for fixed parameter $Y$. Figure 5 illustrates the RRN of the network in Figure 1 under the assumption that both arcs $(2,3)$ and $(6,7)$ fail, resulting in complete flow losses on paths $P_1$, $P_3$, and $P_4$. Via the LP Formulation (2), we can also obtain solutions in the two cases with respective two and four units of flow losses: (i) $f_{87}^1 = f_{76}^1 = f_{32}^1 = f_{41}^1 = f_{85}^1 = 0$, $f_{65}^1 = f_{54}^1 = f_{43}^1 = f_{21}^1 = 2$, $\rho_7^1 = 0$, $\rho_3^1 = 2$, $\lambda_1^1 = 2$, and (ii) $f_{87}^2 = f_{76}^2 = f_{32}^2 = f_{85}^2 = f_{43}^2 = 0$, $f_{65}^2 = f_{54}^2 = f_{41}^2 = f_{21}^2 = 2$, $\rho_7^2 = 0$, $\rho_3^2 = 0$, $\lambda_1^2 = 4$.

![Figure 5: The RRN of the graph in Figure 1 and two possible values of $L(x, Y)$.](image)

### 2.2 Algorithms for Computing the Maximum and the Minimum Flow Losses

The SMCF problems studied in this paper aim to ensure the reliability of any arc flow solution $x$ by bounding possible losses $L(x, Y)$ under uncertainty $Y$. Given that multiple values of $L(x, Y)$ for the same combination of solution $x$ and parameter $Y$, here we focus on how to compute the minimum and the maximum flow losses, respectively denoted by $L_{\max,\min}(x, Y) = \min_{\lambda,\rho,f} \{ L(x, Y) | (2a)-(2d) \}$, and
\( \overline{L}(x, Y) = \max_{\lambda, \rho, f} \{ L(x, Y) | \text{(2a) - (2d)} \} \). We construct two algorithms in particular for computing \( L(x, Y) \) and \( \overline{L}(x, Y) \) in polynomial time.

**Algorithm 1 ALG(\( \mathcal{M} \))**: Routing withdrawn flows from set \( \mathcal{S} \) to set \( \mathcal{M} \) in the RRN \( G(x, Y) \).

1. **INPUTS**: \( \mathcal{M} \); \( g_i \), \( \forall i \in \mathcal{S} \); \( x_{ij}, Y_{ij}, f_{ji}, \forall (i, j) \in \mathcal{A} \).
2. Let \( Q = \{ \text{The set of all directed paths from } k \text{ to } l \text{ in } G(x, Y) \} \), for all \( k \in \mathcal{S} \), \( l \in \mathcal{M} \).
3. if \( \mathcal{M} = \mathcal{S} \) then
   4. \( \lambda_l = 0 \) and \( e_l = D_l \), \( \forall l \in \mathcal{S} \);
5. else if \( \mathcal{M} = \mathcal{T} \) then
   6. \( \rho_l = 0 \) and \( e_l = \sum_{j: (j, l) \in \mathcal{A}} Y_{jl} x_{jl}, \forall l \in \mathcal{T} \).
7. end if
8. while \( Q \neq \emptyset \) do
9. Pick an arbitrary path \( P \) from \( Q \).
10. Let \( k \) and \( l \) be the origin and destination nodes of \( P \) in the RRN \( G(x, Y) \).
11. Let \( \delta = \min \{ e_l, \min_{(i, j) \in P} \{ x_{ij} - f_{ij} \}, g_k \} \).
12. if \( \delta = 0 \) then
   13. \( Q \leftarrow Q \setminus \{ P \} \);
14. else
   15. \( f_{ij} = f_{ij} + \delta, \forall (i, j) \in P \), \( e_l = e_l - \delta \), and \( g_k = g_k - \delta \).
16. if \( \mathcal{M} = \mathcal{S} \) then
17. \( \lambda_l = \lambda_l + \delta \);
18. else if \( \mathcal{M} = \mathcal{T} \) then
19. \( \rho_l = \rho_l + \delta \).
20. end if
21. end if
22. end while
23. return \( \lambda_i, \forall i \in \mathcal{S}; \rho_i, \forall i \in \mathcal{T}; g_i, \forall i \in \mathcal{S}, f_{ij}, \forall (i, j) \in \mathcal{A} \).

For any MCF solution \( x \), define sets \( \mathcal{S} \) and \( \mathcal{T} \) as follows,

\[
\mathcal{S} \equiv \left\{ i \in \mathcal{N} \left| \sum_{j: (i, j) \in \mathcal{A}} Y_{ij} x_{ij} > 0 \right. \right\} \quad \text{and} \quad \mathcal{T} \equiv \left\{ i \in \mathcal{N} \left| \sum_{j: (j, i) \in \mathcal{A}} Y_{ji} x_{ji} > 0 \right. \right\},
\]

i.e., nodes in the corresponding RRN having excesses and deficits of withdrawn flows, respectively. We construct \( \text{ALG}(\mathcal{M}) \) (Algorithm 1) to calculate a feasible solution \( (\lambda, \rho, f) \) to Formulation (2) by rerouting flows from nodes in \( \mathcal{S} \) to nodes in \( \mathcal{M} \), where set \( \mathcal{M} \) can either be set \( \mathcal{S} \) or be set \( \mathcal{T} \), after knowing the failed arcs indicated by parameter \( Y \).

We demonstrate the steps in Algorithm 1 by using the example in Figure 5, where \( \mathcal{T} = \{3, 7\} \) and \( \mathcal{S} = \{2, 6\} \). The solution \( (\lambda^1, \rho^1, f^1) \), yielding two unit of flow losses, corresponds to decisions
of routing two units of flow from node 6 to node 3, and then two units of flow from node 2 to node 1. On the other hand, the solution \((\lambda^2, \rho^2, f^2)\) with four units of flow losses corresponds to routing two units of flow from node 6 to node 1, and the other two units from node 2 to node 1.

\textbf{ALG(}M\textbf{)} is an augmenting path algorithm variant for maximizing flows from multiple source nodes in \(\mathcal{S}\) (with certain supply capacities based on arc failure information) to sink nodes in \(\mathcal{M}\) (with bounded variable demands). Denote \(g^0_i \equiv \sum_{j:(i,j) \in A} Y_{ij} x_{ij}, \forall i \in \mathcal{S}\). Implementing \textbf{ALG(}S\textbf{)} (with \(\mathcal{M} = \mathcal{S}\), \(g_i = g^0_i, \forall k \in \mathcal{S}\), and \(f_{ij} = 0, \forall (i,j) \in \mathcal{A}\)) on the RRN \(\mathcal{G}(x, Y)\) will give the maximum amount of flow withdrawn (i.e., \(L(x, Y) = \sum_{i \in \mathcal{S}} \lambda_i\)). On the other hand, by summing up (2a) over all \(i \in \mathcal{N}\), we have \(\sum_{i \in \mathcal{S}} \lambda_i = \sum_{(i,j) \in \mathcal{A}} Y_{ij} x_{ij} - \sum_{i \in \mathcal{N}} \rho_i\), indicating that

\[
\min_{\lambda, \rho, f} \left\{ \sum_{i \in \mathcal{S}} \lambda_i : \text{(2a)-(2d)} \right\} \Leftrightarrow \sum_{(i,j) \in \mathcal{A}} Y_{ij} x_{ij} - \max_{\lambda, \rho, f} \left\{ \sum_{i \in \mathcal{N}} \rho_i : \text{(2a)-(2d)} \right\}.
\]

Thus, the following two steps calculate \(L(x, Y)\). First, we run \textbf{ALG(}T\textbf{)} with \(\mathcal{M} = \mathcal{T}\), \(g_i = g^0_i, \forall i \in \mathcal{S}\), and \(f_{ij} = 0, \forall (i,j) \in \mathcal{A}\) to update \(g_i, \forall i \in \mathcal{S}\) and \(f_{ij}, \forall (i,j) \in \mathcal{A}\). Then we run \textbf{ALG(}S\textbf{)} with solutions obtained from the first step to calculate \(L(x, Y)\), which is equal to \(\sum_{i \in \mathcal{S}} \lambda_i\).

\section{Models and Algorithms of SMCF-VaR}

Denote the random form of parameter \(Y\) by \(Y_\xi\). Consider set \(\Omega\) of a finite realizations of \(Y_\xi\), denoted by \(Y_\xi_s, \forall s \in \Omega\). We first restrict the VaR of random flow losses \(L(x, Y_\xi)\) and formulate

\[
\text{SMCF-VaR} : \min \left\{ \sum_{(i,j) \in \mathcal{A}} C_{ij} x_{ij} : \text{(1b), (1c)}, \mathbb{P}\{L(x, Y_\xi) \leq \eta\} \geq 1 - \theta \right\},
\]

where we keep MCF constraints (1b) and (1c), and require at least \(1 - \theta\) probability of having \(L(x, Y_\xi)\) being no more than a threshold loss limit, denoted by \(\eta\) (i.e., the VaR of \(L(x, Y_\xi)\)). Here \(\theta\) is a given small risk tolerance that is close to zero.

\textbf{Remark 2}. As \(L(x, Y_\xi_s)\) may take multiple values even for fixed solution \(x\) and realization \(Y_\xi_s\), decision makers may be interested in bounding the VaR of \(L(x, Y_\xi)\) or \(L(x, Y_\xi_s)\), respectively reflecting “risk-averse” and “risk-seeking” behavior against the uncertainty. Problem SMCF-VaR is in fact equivalent to its \(L(x, Y_\xi)\)-variant that possesses a probabilistic constraint \(\mathbb{P}\{L(x, Y_\xi) \leq \eta\} \geq 1 - \theta\), because \(L(x, Y_\xi)\) will always take on the smallest value for an optimal solution \(x\) to SMCF-VaR in any scenario \(s\) so that the violation probability with respect to the fixed VaR \(\eta\) is minimized.

This result cannot be generalized to the \(L(x, Y_\xi)\)-variant of SMCF-VaR, which needs to be solved as a bilevel program where the inner problem \(L(x, Y_\xi)\) is a maximization problem given fixed \(x\) and \(Y_\xi\). The \(L(x, Y_\xi)\)-variant represents a more conservative implementation of the SMCF problems we study, and the bilevel program requires developing significantly different approaches from the ones for solving SMCF-VaR that we will elaborate shortly. In this paper, we focus on SMCF-VaR and
later SMCF-CVaR that restrict the risk associated with the minimum flow losses $L(x, Y_{\xi})$, and will discuss future research in Section 7 for tackling bilevel SMCF problems involving $\mathcal{L}(x, Y_{\xi})$.

### 3.1 An Integer Programming Reformulation of SMCF-VaR

We reformulate SMCF-VaR as a mixed-integer program for $Y_{\xi}$ with finite realizations by defining an additional binary variable associated with each realized scenario. Let $\text{Prob}_{\xi s}$ be the probability of realizing $Y_{\xi s}$, $\forall s \in \Omega$ such that $\sum_{s \in \Omega} \text{Prob}_{\xi s} = 1$. Define binary variables $z^s$, $\forall s \in \Omega$, such that $z^s = 1$ if $L(x, Y_{\xi s})$ is larger than the threshold limit $\eta$, and $z^s = 0$ otherwise. The SMCF-VaR in (4) is equivalent to a deterministic model:

**SMCF-VaR-D:**

$$\begin{align*}
\text{min} & \quad \sum_{(i,j) \in A} C_{ij} x_{ij} \\
\text{s.t.} & \quad (1b)-(1c) \quad (2a)-(2d) \quad \text{with inputs } Y_{\xi s} \text{ and variables } f^s, \lambda^s, \text{ and } \rho^s, \forall s \in \Omega \\
& \quad L(x, Y_{\xi s}) = \sum_{i \in S} \lambda^s_i \leq Mz^s + \eta \quad \forall s \in \Omega \quad (5a) \\
& \quad \sum_{s \in \Omega} \text{Prob}_{\xi s} z^s \leq \theta \quad (5b) \\
& \quad z^s \in \{0, 1\} \quad \forall s \in \Omega, \quad (5c)
\end{align*}$$

where $M$ is an arbitrary large number to guarantee the validity of (5a) when $z^s = 1$ in some scenario $s$. Variable $\lambda^s_i$ represents the accumulated flow loss at node $i \in S$ of scenario $s$, and (5a) specify scenario-based values of $L(x, Y_{\xi s})$ via linear constraints (2a)-(2d). Constraints (5a) also enforce $z^s$ being 1 if $L(x, Y_{\xi s}) > \eta$. Constraint (5b) restricts the violation chance being no more than $\theta$.

**Remark 3.** The number of binary variables in Formulation (5) (as well as models discussed in the rest of the paper) depends on the cardinality of set $\Omega$. In general, Monte Carlo sampling (e.g., Mak et al. 1999, Norkin et al. 1998) is a common approach for generating independent scenarios from known distributions of the uncertainty. Shapiro and Homem-de-Mello (2000) developed a Sample Average Approximation (SAA) approach for solving stochastic linear programs, and Kleywegt et al. (2002) extended the approach to stochastic discrete optimization problems. In particular, Luedtke and Ahmed (2008) approximated optimization problems with probabilistic constraints by using SAA, and provided sufficient numbers of samples and scenarios to derive lower bounds or feasible solutions with certain confidence levels. In this paper, we focus on solving SMCF-VaR given set $\Omega$ and realizations $Y_{\xi s}$ with their probabilities $\text{Prob}_{\xi s}$, $\forall s \in \Omega$. Our approaches can be integrated into an SAA implementation that follows methods provided in Luedtke and Ahmed (2008) for generating independent samples and set $\Omega$ of scenarios in each sample.
3.2 Decomposition and a Cutting-Plane Algorithm

We view SMCF-VaR as a two-stage optimization problem that verifies whether or not the value of flow losses in each scenario \( s \) is no more than \( \eta \) after knowing an MCF solution \( x \) and \( Y_{\xi^s} \). We consider the Benders decomposition approach (Birge and Louveaux 1997, Benders 1962) which partitions all decision variables into two subsets that are made sequentially. (In the context of two-stage stochastic programming, we classify decisions into “here-and-now” decisions and “wait-and-see” decisions, with the former made before knowing realizations of the uncertainty and the latter depending on the former “here-and-now” decisions as well as uncertain realizations.) For SMCF-VaR, we formulate a master problem which decides an MCF solution \( x \), and subproblems which verify whether \( x \) violates the probabilistic constraint. We develop valid cutting planes to be generated into the first-stage relaxed master problem by using the optimal dual information of the second-stage subproblems. We then iteratively solve the master problem until no cuts are required.

Let \( p^s \) be the objective value associated with each subproblem \( s \) for all \( s \in \Omega \). The master problem is given by

\[
\min \left\{ \sum_{(i,j) \in A} C_{ij} x_{ij} : (1b)-(1c), \sum_{s \in \Omega} \text{Prob} \xi^s p^s \leq \theta, F(x, p^s) \geq 0, p^s \geq 0, \forall s \in \Omega \right\} \tag{6}
\]

where \( F(x, p^s) \geq 0 \) is a set of cutting planes derived from the \( s^{th} \) subproblem formulated as

\[
p^s = \min \left\{ z^s : (2a)-(2d) \text{ with } Y_{\xi^s}, f^s, \lambda^s, \text{ and } \rho^s, \sum_{i \in S} \lambda^s_i \leq M z^s + \eta, z^s \in \{0, 1\} \right\}. \tag{7}
\]

Algorithm 2 presents a cutting-plane approach for solving SMCF-VaR-D. We take the best solution, found by the Benders decomposition after a fixed number of iterative cut generations in the form of \( F(x, p^s) \geq 0 \), as a “good-quality” incumbent solution and change to off-the-shelf solvers for continuing optimizing SMCF-VaR-D as an integer program. To compute cut \( F(x, p^s) \geq 0 \) in each subproblem \( s \), we use an LP relaxation of Formulation (5) and its optimal linear duals to derive coefficients of the Benders cuts.

**Objective Bounding** After obtaining a first-stage solution \((\hat{x}, \hat{p})\), before solving each subproblem (7) in scenario \( s \), we compute \( \underline{L}(\hat{x}, Y_{\xi^s}) \) by running \( \text{ALG}(T) \) and then \( \text{ALG}(S) \). If \( \hat{p}^s = 0 \) and \( \underline{L}(\hat{x}, Y_{\xi^s}) \leq \eta \), then no cut is needed for subproblem \( s \), \( \forall s \in \Omega \). This pre-checking procedure, using the previously derived polynomial-time algorithm, permits us solving subproblem \( s \) only when \( \hat{p}^s = 0 \) and \( \underline{L}(\hat{x}, Y_{\xi^s}) > \eta \). That is, *a priori* to the implementation of Algorithm 2, we compare an estimated \( \hat{p}^s \) with a real violation status of the target VaR without explicitly solving the \( s^{th} \) subproblem to save the computational effort.
Algorithm 2 Solving SMCF-VaR-D via cutting-plane procedures.

1: Solve a relaxed master problem (6). Denote \((\hat{x}, \hat{p})\) as tentative optimal solutions.
2: Denote \(F(\hat{x}, \hat{p})\) as the set of cutting planes to be generated based on solution \((\hat{x}, \hat{p})\).
3: Set \(F(\hat{x}, \hat{p}) = \emptyset\).
4: for All scenarios \(s \in \Omega\) do
5: Fix \(x = \hat{x}\) in the \(s^{th}\) subproblem (7) (or an approximation of the subproblem, e.g., the LP relaxation), \(\forall s \in \Omega\). Denote \(\hat{z}^s\) as the optimal objective value.
6: if \(\hat{p}^s < \hat{z}^s\) then
7: Compute coefficients of cut \(F(x, p^s) \geq 0\) by using strong/weak optimality conditions.
8: Update \(F(\hat{x}, \hat{p}) = F(\hat{x}, \hat{p}) \cup \{F(x, p^s) \geq 0\}\).
9: end if
10: end for
11: if \(F(\hat{x}, \hat{p}) \neq \emptyset\) then
12: Generate all cuts in \(F(\hat{x}, \hat{p})\) to the relaxed master problem (6). Go to Step 1.
13: else
14: Solution \((\hat{x}, \hat{p})\) is the best we obtain from the decomposition procedure.
15: if \(\hat{p}\) is not integer-valued then
16: Use \((\hat{x}, \hat{p})\) as an incumbent solution, and solve SMCF-VaR-D in commercial solver, e.g., CPLEX.
17: else
18: Return \((\hat{x}, \hat{p})\) as an optimal solution.
19: end if
20: end if

4 Models and Algorithms of SMCF-CVaR

We also study a coherent risk measure, CVaR\(\theta\), to derive a solution that is feasible to SMCF-VaR for the same risk parameter \(\theta\). In general, CVaR\(\theta\) is known as a convex approximation of VaR\(\theta\), which measures the expected random amount of a random variable exceeding its VaR for a given risk level \(\theta\). Our focus is to analyze SMCF variants given by the two approaches under uncertain 0-1 arc failure. According to our definitions of flow losses, additional arc failure may or may not lead to an increase of flow losses and will depend on specific network topologies. We aim to formulate the two variants based on VaR and CVaR to seek optimal flow patterns against the topological uncertainty and examine their differences.

Similar to SMCF-VaR, we formulate SMCF-CVaR for given risk parameter \(\theta\) (equivalent to its
\[ L(x, Y_\xi) \text{-variant) as} \]

\[
\begin{align*}
\min & \quad \sum_{(i,j) \in A} C_{ij} x_{ij} \\
\text{s.t.} & \quad (1b)-(1c) \\
& \quad (2a)-(2d) \text{ with inputs } Y_{\xi s} \text{ and variables } f^s, \lambda^s, \text{ and } \rho^s, \forall s \in \Omega \\
& \quad \alpha + \sum_{s \in \Omega} \frac{\text{Prob}_{x, b_{\xi s}}}{\theta} \leq \eta \quad \text{(8a)} \\
& \quad \sum_{i \in S} \lambda^s_i \leq b_{\xi s} + \alpha \quad \forall s \in \Omega \quad \text{(8b)} \\
& \quad \alpha \geq 0, \quad b_{\xi s} \geq 0 \quad \forall s \in \Omega \quad \text{(8c)}
\end{align*}
\]

where we keep all previous constraints (1b) and (1c) which respectively maintain flow balance of nodes and calculate flow losses in each scenario \( s, \forall s \in \Omega \). Variable \( \alpha \) in (8a) is the actual VaR of flow losses \( L(x, Y_\xi) \), and is nonnegative ensured by (8c). Variable \( b_{\xi s} \) in (8b) represents the amount of flow losses (i.e., \( L(x, Y_\xi) = \sum_{i \in S} \lambda^s_i \)) being greater than the VaR value in scenario \( s, \forall s \in \Omega \).

SMCF-CVaR is a linear program with fixed risk parameter \( \theta \) and threshold \( \eta \). We apply the Bender decomposition. We relax constraints (8b) and formulate the first-stage master problem as

\[
\min \left\{ \sum_{(i,j) \in A} C_{ij} x_{ij} : (1b)-(1c), (8a), (8c) \right\}.
\] (9)

By solving the master problem (9), we obtain a tentative optimal solution \((\hat{x}, \hat{b}_{\xi s}, \hat{\alpha})\). Now for fixed solution \( \hat{x} \) and parameter \( Y_{\xi s} \), consider subproblems

\[
\min_{\lambda^s, \rho^s, f^s} \left\{ \sum_{i \in S} \lambda^s_i : (2a)-(2d) \right\}, \forall s \in \Omega,
\] (10)

yielding an optimal solution \((\hat{\lambda}^s, \hat{\rho}^s, \hat{f}^s)\) to each subproblem (10) associated with realization \( s \), for all \( s \in \Omega \). We evaluate whether or not \( \sum_{i \in S} \hat{\lambda}^s_i \) is greater than \( \hat{b}_s + \hat{\alpha} \) in each scenario \( s \in \Omega \). If yes, we generate into the master problem (9) a feasibility cut in the form of

\[
b_s + \alpha \geq \sum_{(i,j) \in A} Y^s_{ij} (\hat{\phi}^s_i + \hat{\tau}^s_j) x_{ij} + \sum_{(i,j) \in A} (1 - Y^s_{ji}) \hat{\gamma}^s_{ij} x_{ji} + \sum_{i \in S} D_i \hat{\zeta}^s_i \quad \text{(11)}
\]

where \( \hat{\phi}^s, \hat{\tau}^s, \hat{\zeta}^s, \hat{\tau}^s \) are optimal solutions of the dual variables associated with constraints (2a), (2b), (2b), (2d), respectively, with dual variables \( \phi^s \) being unrestricted, and all \( \gamma^s, \zeta^s, \tau^s \leq 0, \forall s \in \Omega \).

An optimal solution to SMCF-CVaR provides a feasible solution to the SMCF-VaR\( \theta \) for the same risk threshold value \( \theta \). The former is much easier to optimize as a linear program. Therefore, for solving SMCF-VaR as Formulation (5), one can first solve the corresponding SMCF-CVaR and employ the optimal result as an incumbent solution to compute bounds of SMCF-VaR and improve computational time.
5 Applications in Risk-Aware Kidney Exchange

In this section, we demonstrate how to use the developed SMCF-VaR and SMCF-CVaR for solving a class of risk- and failure-aware kidney exchange problems. The problems focus on pairing kidneys given by living donors who may be incompatible, due to blood type, tissue type, or other reasons, with their target patients. The method of kidney exchange has recently emerged to enable willing but incompatible donor-patient pairs to swap donors, so that each patient obtains a compatible kidney. Roth et al. (2004) initially propose to organize kidney exchange on a large scale, with the formation of the New England Program for Kidney Exchange (NEPKE) described in Roth et al. (2005).

With incompatible donors and their target patients entering a kidney exchange program, each incompatible donor-patient pair seeks to swap their donors with other pairs to obtain a compatible kidney. We encode a general kidney exchange problem as the graph $G(\mathcal{N}, \mathcal{A})$ considered in this paper as follows. We construct a node for each donor-patient pair in $\mathcal{N}$, and add an arc from one pair $i$ to another pair $j$ if the donor of pair $i$ is compatible with the patient of pair $j$ to form the arc set $\mathcal{A}$. We associate weight $w_{ij}$ with each arc $(i, j)$ in $\mathcal{A}$, representing the utility or social welfare attained if the transplant from $i$ to $j$ is implemented. Note that

- A cycle in this graph represents a possible swap among multiple pairs, with each pair in the cycle receiving the kidney from the next pair.

- A feasible exchange solution is a collection of node-disjoint cycles since each pair can give at most one kidney.

![Network encoding of a kidney exchange.](image)

Figure 6 illustrates an example of the network representation of a case with five donor-patient pairs. The graph contains 4 cycles, $c_1 = 1 \rightarrow 2 \rightarrow 1, c_2 = 2 \rightarrow 3 \rightarrow 2, c_3 = 3 \rightarrow 4 \rightarrow 3, c_4 = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$. Both $\{c_1, c_3\}$ and $\{c_4\}$ are feasible exchange solutions as they are maximal sets of node-disjoint cycles existing in the graph.
Therefore, a deterministic kidney exchange problem aims to identify a set of node-disjoint cycles with the maximum weights of all arcs contained in the cycles, representing the maximum social welfare/utility attained by pairing kidneys. Here we specify the general arc-flow decision variables $x_{ij}$ in [MCF] as binary variables $x'_{ij}$, for all $(i, j) \in A$, such that

\[
x'_{ij} = \begin{cases} 
1 & \text{arc } (i, j) \in A \text{ is contained in the exchange solution} \\
0 & \text{otherwise}
\end{cases}
\]

We specify a variant of the MCF for computing a feasible exchange that results in the maximum social welfare. A deterministic formulation is given by

\[
\begin{align*}
\text{max} & \quad \sum_{(i,j) \in A} w_{ij} x'_{ij} \\
\text{s.t.} & \quad \sum_{j: (i,j) \in A} x'_{ij} - \sum_{j: (j,i) \in A} x'_{ji} = 0 \quad \forall i \in N \\
& \quad \sum_{j: (i,j) \in A} x'_{ij} \leq 1 \quad \forall i \in N \\
& \quad x'_{ij} \in \{0, 1\} \quad \forall (i,j) \in A,
\end{align*}
\]

where (12a) maximizes the total utility yielded by successful exchanges of kidneys; (12b) ensure cycle solutions by maintaining zero flow-in and flow-out balance at each node, while these cycles are disjoint by node guaranteed by (12c), which indicate no more than one kidney is given out by the donor of pair $i$, for each node $i \in N$. It can be easily verified that the foregoing problem is equivalent to a matching problem in a transformed graph in which two copies of each node in $N$ are created, respectively representing the donor and the patient of the corresponding node. There exists an arc between a donor node and a patient node if they are compatible, and thus the resulting transformed graph is bipartite in which we can partition the donor nodes and the patient nodes into two subsets and arcs only exist between but not within the two subsets. Therefore, the problem of identifying node-disjoint cycles is equivalent to finding a perfect matching on a bipartite graph that is also equivalent to a maximum flow problem. We refer to the interested readers to Ahuja et al. (1993) for more detailed discussions of the three equivalent problems.

In a stochastic setting, we consider some previously compatible donors and receivers may be found incompatible after pairing the exchanges, which we interpret as arc failure (topological changes) in the compatible graph $G(N', A)$. An arc failure can be due to several reasons. For example, a last-minute testing in a kidney exchange often reveals new incompatibilities that were not detected in the initial testing. Or a donor may regret or fail to fulfill her obligation due to preference change. In all such cases, all pairs in those cycles containing incompatible pairs are affected since a planned transplant operation is no longer possible. Dickerson et al. (2013) address the problem of failure-aware kidney exchange by maximizing the expected utility while taking into
account match failure, and show that it can significantly increase the expected number of lives saved both in theory and computation, compared with using a deterministic model. Here we cast two types of risk-aware kidney exchange problems as variants of SMCF-VaR and SMCF-CVaR, elaborated as follows.

Recall our notation in earlier sections for defining the SMCF problems. Similarly, we assume random 0-1 match failure of arc \((i, j)\), denoted by a Bernoulli random variable \(Y'_{ij}\) such that \(Y'_{ij} = 1\) if it fails and \(Y'_{ij} = 0\) otherwise, for all \((i, j) \in \mathcal{A}\). The set \(\Omega\) now specifies a finite number of realizations of the uncertain \(Y' = [Y'_{ij}; (i, j) \in \mathcal{A}]^T\) and \(Y'_{ji}\) denotes a realized match failure status of scenario \(s \in \Omega\). We modify the definition of flow losses in general SMCF and for the kidney exchange application, the total utility losses due to any arc failure in a cycle equal to the sum of utility weights of all arcs in the cycle. The total amount of utility losses in \(\mathcal{G}(\mathcal{N}, \mathcal{A})\) can be calculated via constructing the RRN \(\mathcal{G}(\bar{x}', Y')\) as before given an exchange solution \(x'\) and random vector \(Y'\). Figure 7 illustrates the RRN of a 4-way exchange cycle with the failure of arc \((2, 1)\) and arc \((4, 3)\), meaning that the donor of pair 2 (pair 4) cannot give the kidney to the patient of pair 1 (pair 3), and therefore all exchanges involved in the cycle cannot be implemented.

![Figure 7: A 4-way exchange solution and the corresponding RRN](image)

We simply the LP formulation (2) for solving the kidney exchange problem on the RRN of a compatible graph for computing utility losses \(L(x', Y')\), given an exchange solution \(x'\) and match failure uncertainty \(Y'\). We reuse the previous notation \(\mathcal{A}\) and \(f_{ij}\) to indicate the set of arcs in the RRN and the amount of withdrawn flows on arc \((i, j) \in \mathcal{A}\), respectively, and formulate

\[
L(x', Y') = \sum_{(j,i) \in \mathcal{A}} Y'_{ji}w_{ji} + \min f \sum_{(i,j) \in \mathcal{A}} w_{ij}f_{ij}
\]

s.t.

\[
\sum_{j:(i,j) \in \mathcal{A}} f_{ij} - \sum_{j:(j,i) \in \mathcal{A}} f_{ji} = \sum_{j:(i,j) \in \mathcal{A}} Y'_{ij}x'_{ij} - \sum_{j:(j,i) \in \mathcal{A}} Y'_{ji}x'_{ji}, \quad \forall i \in \mathcal{N}
\]

\[
0 \leq f_{ij} \leq (1 - Y'_{ji})x'_{ji}, \quad \forall (i, j) \in \mathcal{A}.
\]

where the objective (13a) contains two parts of utility losses as the sum of utility weights of all failed
arcs and the sum of utility weights on the remaining arcs where exchanges have been withdrawn (indicated by the values of variables $f_{ij}$, $\forall (i, j) \in \mathcal{A}$); (13b) represent flow conservation constraints, where the withdrawn supply at a node $i$ is 1 if pair $i$ in $\mathcal{A}$ gives out its donor’s kidney but the match fails, and is -1 if there is a kidney received by the patient of pair $i$ but the match fails. Constraints (13c) allow a unit withdrawn flow on arc $(i,j)$, $\forall (i,j) \in \mathcal{A}$ only if arc $(j,i) \in \mathcal{A}$ is used (has an exchange) and it does not fail.

**Theorem 2.** The value of $L(x', Y')$, given $x'$ and $Y'$, measures exactly the total utility losses of affected exchanges due to match failure.

*Proof.* Because all cycles in an exchange solution $x'$ are node-disjoint, it is sufficient to show the result for a solution consisting of one cycle. By removing failed arcs from the cycle, the remaining arcs form a collection of paths. Consider the RRN of a compatible graph (see, e.g., Figure 7), with the cost of each arc $(i,j) \in \mathcal{A}$ set as the utility weight of arc $(j,i) \in \mathcal{A}$. The minimum-cost flow solution replicates a collection of affected exchanges in addition to the failed exchanges (arcs) indicated by $Y'$. The total utility losses of all unimplementable exchanges equal to the sum of utility weights of directly failed arcs and arcs in the remaining paths. This completes the proof.

Different from $L(x, Y)$ described before for general SMCF problems, the utility loss $L(x', Y')$ has a unique value given fixed $x'$ and $Y'$ because a feasible exchange only consists of node-disjoint cycles and therefore the flow representation in the remaining graph is unique. Note that Formulation (13a)–(13c) for calculating $L(x', Y')$ is also much more simplified compared with Formulation (2) in the general case. To formulate a risk-aware kidney exchange problem, one can follow the previous procedures to incorporate $L(x', Y'_\xi)$ with random $Y'_\xi$ to a SMCF-VaR setting and formulate:

$$\max \left\{ \sum_{(i,j) \in \mathcal{A}} w_{ij} x'_{ij} : (12b)–(12d), \ P \{ L(x', Y'_\xi) \leq \eta \} \geq 1 - \theta \right\},$$  \hspace{1cm} (14)$$

where the minimization of the objective value of the inner problem $L(x', Y'_\xi)$ will be naturally enforced at optimum and thus we can exactly follow the previous procedures to transform (14) into a mixed-integer program given the finite support set $\Omega$ of $\xi$. A SMCF-CVaR formulation of the risk-aware kidney exchange problem can be established in a similar way as before and we omit the details for the sake of brevity.

## 6 Computational Studies

This section mainly consists of three parts. First, we demonstrate the computational performance of SMCF-VaR and SMCF-CVaR on a set of randomly generated graph instances, and in particular emphasize computational-time savings by formulating path flow losses via linear constraints, rather than use path flow representation variables. Second, we report and compare the results of VaR and
CVaR variants, particularly focusing on (i) the relationship between their optimal objective values and (ii) their optimal solution patterns. The computation of the first two parts focuses on general network flow instances, while in the third part, we compute SMCF-VaR and SMCF-CVaR models of the risk-aware kidney exchange problem described in Section 5. Through computational studies in all three parts, we aim to demonstrate general insights and compare the results yielded by VaR and CVaR approaches so that one can apply our research for solving specific SMCF problems that could fit in the assumptions (e.g., flow decisions are made a priori to knowing the uncertainty; flows are delivered via paths from origins to destinations) made in this paper. We also analyze various optimal pairing decisions of exchanging kidneys by imposing VaR and CVaR risk measures on potential utility losses, compared with optimal results obtained from a deterministic exchange problem, so that we can demonstrate the necessity of considering match failure uncertainty in such applications.

6.1 Experimental Setup

We randomly generate instances of the directed graph $G = (N, A)$ in different sizes with 10-, 20-, and 30-nodes. For every node $i \in N$, we randomly pick an integer between the minimum and the maximum outgoing degrees required for every node in $G$, of which the values are indicated in Table 1 for the corresponding graph sizes and density levels. This integer value represents a target degree of node $i$. We then repeatedly generate nodes adjacent to node $i$ from all nodes in $N \setminus \{i\}$ until reaching the degree target. After this, to provide an initially connected graph and ensure the existence of paths between origins and destinations, we manually add arcs connect each pair of nodes if they are not connected, checked by using a search algorithm. The cost and capacity of each arc are randomly generated based on uniform distributions on [1, 20] and on [10, 30] respectively for all instances. Moreover, from the node set $N$ we randomly pick 1, 2, or 3 supply node(s) and also 1, 2, or 3 demand node(s), and generate random supply/demand values of each selected node, while ensuring that the sum of supplies at all nodes in $S$ equals to the sum of demands at all nodes in $D$ for each instance.

In this paper, we aim to propose, solve, and test general SMCF problems under uncertain arc failure with finite realizations. In our tests we generate 50 scenarios for each problem instance, i.e., $|\Omega| = 50$. All scenarios are considered equally likely, i.e., $\text{Prob}_{\xi} = 1/|\Omega|$. The scenarios
are generated by conducting independent Bernoulli trials on all arcs, using either 0.25 or 0.1 as a homogeneous failure probability of each arc. In this way, we can compare the performances of the CVaR and VaR approaches for handling SMCF instances on networks that are relatively more vulnerable (i.e., the ones generated using 0.25 failure probability) and more reliable (i.e., the ones generated using 0.1 failure probability). Depending on specific problems and applications, one can generate scenarios with either correlated or independent failures of arcs given specified arc-failure distributions, by using Monte-Carlo simulation-based sampling approaches reviewed in Remark 3.

It is also possible that no feasible flow assignment exists to the VaR or the CVaR models of the SMCF problem given specific \( \theta \) and \( \eta \) values we test. To address this issue, we adopt the following rules to keep all instances feasible in our numerical experiments. For any given \( \theta \) and \( \eta \), we first solve SMCF-CVaR (which can be easily solved as a linear program). Note that a CVaR formulation has a more restricted feasible region and thus its optimal solution always yields a feasible solution to the corresponding VaR for the same SMCF instance. Therefore, the feasibility of the VaR model is guaranteed if there exists a feasible solution to the CVaR model. We regenerate an instance if its CVaR formulation is infeasible.

All computations are implemented in Microsoft Visual C++ 2008, while calling CPLEX 12.2 to solve all instances. All programs are run in Microsoft Windows 7 Enterprise 64-bit operating system on a Dell Desktop with Intel(R) Pentium(R) CPU G6950 2.80GHz and 3 GB RAM.

6.2 Comparison of CPU Times and Objectives

We test various values of the risk parameter \( 1 - \theta \) from 50% to 90% increased by 5% each time. In addition, we test instances with high successful rates, and let \( 1 - \theta \) be 92.5%, 95%, 97.5%, and 99%. For every choice of \( 1 - \theta \), we run 10 instances for \( \eta = 1, \ldots, 10 \). The resulting CPU time is approximately the same for each instance for different \( \eta \)-values. Tables 2 and 3 report average CPU time (in seconds) for different value ranges of \( (1 - \theta) \) (indicated in the second line of each column) respectively under 0.25 and 0.1 homogeneous failure probabilities of each arc. (For example, Column “50–70%” reports average CPU seconds of the ten instances with \( 1 - \theta \) being > 50% and \( \leq 70\% \).) In these tables, \(|N|\)-l, -m, and -h indicate graph instances with low, medium, and high densities.

For both Tables 2 and 3, we provide the following observations. (i) CPU time increases as we increase graph sizes and density levels, but the time increase of solving the SMCF-VaR is much more significant. (ii) CPU time decreases as we increase \( 1 - \theta \), i.e., when both VaR and CVaR seek more robust solutions but with higher cost. (iii) Solving SMCF-CVaR is much quicker than solving SMCF-VaR. (iv) The computational time of SMCF-VaR on graph instances having lower failure probability (i.e., when failure rate = 0.1 of each arc) is much shorter than on relatively unreliable graphs.

We continue analyzing the computational time resulted from various demand settings of nodes in
Table 2: CPU seconds for solving instances with arc failure rate = 0.25.

<table>
<thead>
<tr>
<th></th>
<th>SMCF-VaR</th>
<th></th>
<th>SMCF-CVaR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inst.</td>
<td>=50%</td>
<td>50–70%</td>
<td>75–90%</td>
</tr>
<tr>
<td>10-l</td>
<td>1.08</td>
<td>0.52</td>
<td>0.33</td>
</tr>
<tr>
<td>10-m</td>
<td>5.03</td>
<td>4.86</td>
<td>4.25</td>
</tr>
<tr>
<td>10-h</td>
<td>7.96</td>
<td>22.25</td>
<td>15.74</td>
</tr>
<tr>
<td>20-l</td>
<td>92.83</td>
<td>187.44</td>
<td>61.70</td>
</tr>
<tr>
<td>20-m</td>
<td>857.70</td>
<td>986.27</td>
<td>379.32</td>
</tr>
<tr>
<td>20-h</td>
<td>161.55</td>
<td>338.32</td>
<td>576.64</td>
</tr>
<tr>
<td>30-l</td>
<td>3200.48</td>
<td>3440.52</td>
<td>201.14</td>
</tr>
<tr>
<td>30-m</td>
<td>2435.25</td>
<td>4256.39</td>
<td>1187.17</td>
</tr>
<tr>
<td>30-h</td>
<td>1844.21</td>
<td>7456.36</td>
<td>4408.97</td>
</tr>
</tbody>
</table>

Table 3: CPU seconds for solving instances with arc failure rate = 0.1.

<table>
<thead>
<tr>
<th></th>
<th>SMCF-VaR</th>
<th></th>
<th>SMCF-CVaR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inst.</td>
<td>=50%</td>
<td>50–70%</td>
<td>70–90%</td>
</tr>
<tr>
<td>10-l</td>
<td>0.23</td>
<td>0.61</td>
<td>0.37</td>
</tr>
<tr>
<td>10-m</td>
<td>0.20</td>
<td>0.34</td>
<td>0.82</td>
</tr>
<tr>
<td>10-h</td>
<td>0.26</td>
<td>0.52</td>
<td>1.92</td>
</tr>
<tr>
<td>20-l</td>
<td>2.54</td>
<td>26.44</td>
<td>22.75</td>
</tr>
<tr>
<td>20-m</td>
<td>0.63</td>
<td>1.20</td>
<td>2.54</td>
</tr>
<tr>
<td>20-h</td>
<td>0.95</td>
<td>1.54</td>
<td>18.17</td>
</tr>
<tr>
<td>30-l</td>
<td>90.92</td>
<td>197.86</td>
<td>319.71</td>
</tr>
<tr>
<td>30-m</td>
<td>1.60</td>
<td>3.07</td>
<td>51.36</td>
</tr>
<tr>
<td>30-h</td>
<td>26.28</td>
<td>157.31</td>
<td>407.46</td>
</tr>
</tbody>
</table>

D. Here we only use instances of “20-m” with the scenario sets generated from a failure rate = 0.25 on each arc, and generate demand values at nodes from uniform distributions with means of 20, 50, and 100 with the same variance. Table 4 provides the average CPU seconds and Table 5 presents the average optimal objectives of the ten 20-m instances with various demand means, where we denote each instance by the value of demand mean we use followed by “-nc” or “-c,” respectively representing instances without arc capacities (i.e., uncapacitated) and with arc capacities (i.e., capacitated).

From Table 4 and Table 5, we observe that (i) CPU time slightly increases as we increase the mean of demand; (ii) there are no significant computational-time or objective differences between capacitated and uncapacitated instances, and (iii) the optimal MCF cost dramatically increases when the demand average increases. This indicates that the difficulty of probabilistically satisfying demand in SMCF problems does not directly depend on average node demand. The patterns of CPU time reflected in Table 4 are similar to the ones reflected in Tables 2 and 3.
Table 4: CPU seconds of 20-m instances with means of demand = 20, 50, 100.

<table>
<thead>
<tr>
<th>Inst.</th>
<th>20-nc</th>
<th>20-c</th>
<th>50-nc</th>
<th>50-c</th>
<th>100-nc</th>
<th>100-c</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMCF-VaR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>857.70</td>
<td>642.82</td>
<td>984.67</td>
<td>1020.54</td>
<td>1246.71</td>
<td>1103.36</td>
</tr>
<tr>
<td>50–70%</td>
<td>986.27</td>
<td>751.29</td>
<td>1382.76</td>
<td>1336.66</td>
<td>1741.14</td>
<td>1770.01</td>
</tr>
<tr>
<td>70–90%</td>
<td>379.32</td>
<td>304.92</td>
<td>365.79</td>
<td>375.13</td>
<td>399.61</td>
<td>358.48</td>
</tr>
<tr>
<td>90–99%</td>
<td>16.80</td>
<td>14.68</td>
<td>20.88</td>
<td>20.05</td>
<td>23.54</td>
<td>21.32</td>
</tr>
<tr>
<td>SMCF-CVaR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>3.26</td>
<td>2.73</td>
<td>2.84</td>
<td>3.00</td>
<td>3.06</td>
<td>2.84</td>
</tr>
<tr>
<td>50–70%</td>
<td>3.05</td>
<td>2.57</td>
<td>2.66</td>
<td>2.91</td>
<td>2.81</td>
<td>2.63</td>
</tr>
<tr>
<td>70–90%</td>
<td>2.59</td>
<td>2.17</td>
<td>2.25</td>
<td>2.44</td>
<td>2.47</td>
<td>2.17</td>
</tr>
<tr>
<td>90–99%</td>
<td>2.23</td>
<td>1.86</td>
<td>1.93</td>
<td>2.16</td>
<td>2.16</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Table 5: Optimal flow cost of 20-m instances with means of demand = 20, 50, 100

<table>
<thead>
<tr>
<th>Inst.</th>
<th>20-nc</th>
<th>20-c</th>
<th>50-nc</th>
<th>50-c</th>
<th>100-nc</th>
<th>100-c</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMCF-VaR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>582.8</td>
<td>582.8</td>
<td>2142.6</td>
<td>2142.6</td>
<td>4943.0</td>
<td>4943.0</td>
</tr>
<tr>
<td>50–70%</td>
<td>790.8</td>
<td>790.8</td>
<td>2955.5</td>
<td>2955.5</td>
<td>6829.5</td>
<td>6829.5</td>
</tr>
<tr>
<td>70–90%</td>
<td>1281.5</td>
<td>1281.5</td>
<td>4670.3</td>
<td>4670.3</td>
<td>10610.2</td>
<td>10600.6</td>
</tr>
<tr>
<td>90–99%</td>
<td>1802.1</td>
<td>1802.1</td>
<td>6472.2</td>
<td>6472.2</td>
<td>14668.2</td>
<td>14603.2</td>
</tr>
<tr>
<td>SMCF-CVaR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>1663.3</td>
<td>1663.3</td>
<td>6183.6</td>
<td>6183.6</td>
<td>14490.4</td>
<td>14419.8</td>
</tr>
<tr>
<td>50–70%</td>
<td>1795.7</td>
<td>1795.7</td>
<td>6587.5</td>
<td>6587.5</td>
<td>15187.2</td>
<td>15108.6</td>
</tr>
<tr>
<td>70–90%</td>
<td>1996.5</td>
<td>1996.5</td>
<td>7107.9</td>
<td>7107.9</td>
<td>16137.2</td>
<td>16013.2</td>
</tr>
<tr>
<td>90–99%</td>
<td>2020.9</td>
<td>2020.9</td>
<td>7139.1</td>
<td>7139.1</td>
<td>16195.2</td>
<td>16060.7</td>
</tr>
</tbody>
</table>

Moreover, for both SMCF-VaR and SMCF-CVaR, their feasible regions are nondecreasing with respect to values of reliability $1 - \theta$ and tolerable losses $\eta$. The results also show that the optimal objective values in both models are very sensitive to changes made to $1 - \theta$ and $\eta$. Figure 8 shows the optimal objective values of both SMCF-VaR (in the lower surface) and SMCF-CVaR (in the upper surface) while perturbing the reliability $(1 - \theta)$ and the loss tolerance $(\eta)$. In Figure 8, the difference between the optima of the two models is zero at 99% reliability and keeps increasing when $1 - \theta$ decreases. The optimal objective values in both formulations are neither convex nor concave with respect to parameters $1 - \theta$ and $\eta$. This is because, for SMCF-VaR, there exist binary variables that cause the nonconvexity issue, and for SMCF-CVaR, the risk parameter $\theta$ appears in the denominator of the left-hand side. The lower surface (VaR) appears wavy with respect to the reliability $1 - \theta$. The upper surface (CVaR) appears linear before reaching the loss tolerance and then appears curvy and smooth. In addition, SMCF-CVaR is more sensitive to changes of the loss tolerance $\eta$, but SMCF-VaR is more sensitive to changes of the reliability $1 - \theta$.  

22
6.3 Results of SMCF-VaR and SMCF-CVaR

In all previous tables, the CPU time of SMCF-VaR is much longer than the time of SMCF-CVaR. In this section, we compare detailed solutions obtained by solving SMCF-VaR and SMCF-CVaR under different parameter settings. For the same $1 - \theta$, CVaR$_\theta$ is a conservative approximation of VaR$_\theta$, yielding a feasible solution and an objective upper bound for the latter. In Table 6, we report objective values of $(\text{CVaR-VaR})/\text{VaR} \times 100\%$, indicating gap percentages between the optimal SMCF-CVaR objective and the corresponding optimal objective value of SMCF-VaR. In particular, we specify results of $1 - \theta = 99\%$. Note that the feasible upper bound provided by CVaR is relatively loose when $1 - \theta$ is small, and becomes tighter as we increase $1 - \theta$.

We report $(\text{CVaR-VaR})/\text{VaR} \times 100\%$ values of all 20-h instances with arc failure rate = 0.1. Table 7 depicts results for $1 - \theta = 50\%, 55\%, \ldots, 95\%, 99\%$ (indicated in the first column) and loss tolerance $\eta = 1, \ldots, 10$ (indicated in the first row). The solution gaps are smaller when $\eta$ is larger given any $1 - \theta$ value. For each fixed tolerance $\eta$, the solution gaps first increase, reach a peak value and then decrease if we increase the reliability $1 - \theta$. The peak values are different for different values of $\eta$, and increase as $\eta$ increases. The same optimal solutions of CVaR and VaR appear in many instances as we allow larger loss tolerance, e.g., the columns corresponding to $\eta = 7, 8, 9, \text{and } 10$. 

Figure 8: Comparison of optimal objective values between SMCF-VaR and SMCF-CVaR
Table 6: (CVaR-VaR)/VaR×100% for different values of $1 - \theta$ and arc failure rates

<table>
<thead>
<tr>
<th>Arc Failure Rate = 0.25</th>
<th>Instances</th>
<th>$\geq$50%</th>
<th>50–70%</th>
<th>70–90%</th>
<th>90–99%</th>
<th>$\geq$99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-l</td>
<td>159.53%</td>
<td>119.51%</td>
<td>68.85%</td>
<td>32.54%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>20-m</td>
<td>183.57%</td>
<td>133.53%</td>
<td>59.55%</td>
<td>13.99%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>20-h</td>
<td>36.54%</td>
<td>36.23%</td>
<td>25.64%</td>
<td>9.63%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>30-l</td>
<td>117.43%</td>
<td>93.03%</td>
<td>87.07%</td>
<td>11.24%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>30-m</td>
<td>89.26%</td>
<td>42.7%</td>
<td>26.19%</td>
<td>5.55%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>30-h</td>
<td>91.38%</td>
<td>68.66%</td>
<td>19.91%</td>
<td>6.63%</td>
<td>0.00%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arc Failure Rate = 0.1</th>
<th>Instances</th>
<th>$\geq$50%</th>
<th>50–70%</th>
<th>70–90%</th>
<th>90–99%</th>
<th>$\geq$99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-l</td>
<td>64.18%</td>
<td>57.84%</td>
<td>42.73%</td>
<td>19.32%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>20-m</td>
<td>12.62%</td>
<td>18.03%</td>
<td>31.57%</td>
<td>16.35%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>20-h</td>
<td>8.50%</td>
<td>10.75%</td>
<td>10.96%</td>
<td>3.57%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>30-l</td>
<td>114.67%</td>
<td>84.98%</td>
<td>40.09%</td>
<td>41.51%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>30-m</td>
<td>52.85%</td>
<td>66.67%</td>
<td>53.22%</td>
<td>10.47%</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>30-h</td>
<td>24.88%</td>
<td>28.87%</td>
<td>22.44%</td>
<td>3.76%</td>
<td>0.00%</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: (CVaR-VaR)/VaR×100% values of instances 20-h with arc failure rate = 0.1

<table>
<thead>
<tr>
<th>$1 - \theta$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>44.74%</td>
<td>24.99%</td>
<td>12.22%</td>
<td>3.06%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>55%</td>
<td>47.14%</td>
<td>28.38%</td>
<td>14.03%</td>
<td>4.43%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>60%</td>
<td>49.56%</td>
<td>31.32%</td>
<td>16.20%</td>
<td>5.78%</td>
<td>1.02%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>65%</td>
<td>49.35%</td>
<td>33.59%</td>
<td>18.43%</td>
<td>7.49%</td>
<td>1.81%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>70%</td>
<td>50.06%</td>
<td>36.96%</td>
<td>21.98%</td>
<td>9.59%</td>
<td>3.06%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>75%</td>
<td>39.53%</td>
<td>33.86%</td>
<td>23.15%</td>
<td>11.51%</td>
<td>4.54%</td>
<td>0.57%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>80%</td>
<td>38.28%</td>
<td>32.83%</td>
<td>24.60%</td>
<td>13.34%</td>
<td>6.12%</td>
<td>1.70%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>85%</td>
<td>30.94%</td>
<td>27.77%</td>
<td>21.94%</td>
<td>14.39%</td>
<td>7.89%</td>
<td>3.31%</td>
<td>0.31%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>90%</td>
<td>29.09%</td>
<td>25.13%</td>
<td>19.90%</td>
<td>13.65%</td>
<td>9.66%</td>
<td>5.19%</td>
<td>2.43%</td>
<td>0.23%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>95%</td>
<td>9.80%</td>
<td>8.46%</td>
<td>5.69%</td>
<td>6.03%</td>
<td>7.49%</td>
<td>6.12%</td>
<td>4.44%</td>
<td>3.18%</td>
<td>2.04%</td>
<td>0.91%</td>
</tr>
<tr>
<td>99%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Finally, we check solution robustness and qualities guaranteed by SMCF-CVaR. In Table 8, for every demand-loss allowance $\eta$, each column depicts the real demand losses (i.e., $\alpha$ defined in CVaR) guaranteed by the corresponding SMCF-CVaR given $1 - \theta$. We also fix optimal MCF solutions from the CVaR models, check the satisfaction probabilities associated with the VaR constraints yielded by optimal solutions to SMCF-CVaR given fixed values of $1 - \theta$ and $\eta$. Table 9 reports the real probabilities of satisfaction.

For example, if the probability of flow losses being no more than 10 units is at least 50%, by solving a more conservative SMCF-CVaR with $1 - \theta = 50\%$, a real loss of flows is 5.23, and we guarantee a much more conservative reliability level as 80.67%. If the required reliability is 99%,
there are 10 units of flows lost and the satisfied reliability is 100.00% as no scenarios are violated. In general, when $1 - \theta$ and $\eta$ are small, SMCF-CVaR yields a much looser and a more conservative approximation of SMCF-VaR.

### 6.4 Results of Risk-Aware Kidney Exchange Problems

We apply our approaches to risk-aware kidney exchange (RAKE) problems under the uncertainty of certain donor-patient match failure after the pairing decisions have been made. In particular, we solve the VaR and CVaR models of the problem, of which the modeling and solution details are described in Section 5. We vary the number of donor-patient pairs involved in the compatibility graph $G(N, A)$ as 50, 100, and 200, i.e., $|N| = 50, 100, 200$. We randomly generated ten instances for each size of the graph with each node having an outgoing degree between $0.04|N|$ and $0.12|N|$, i.e.,
the outgoing-degree ranges of 50-, 100-, 200-node graphs are \([2, 6], [4, 12],\) and \([8, 24]\), respectively. We follow the same procedures in Section 6.1 to generate all RAKE instances by making sure that every node in \(\mathcal{N}\) is contained in at least one directed cycle formed by arcs in \(\mathcal{A}\) in each generated instance. We set uniform unit utilities \(w_{ij} = 1\) for all \((i, j) \in \mathcal{A}\). Moreover, we employ the results in Dickerson et al. (2013) to set the failure probabilities of each match (i.e., each arc in \(\mathcal{A}\)). According to their results, “patients tend to have either very high or very low sensitization,” and we sample randomly from a bimodal distribution with 30% of arcs having a low failure rate in \([0\%, 20\%]\) while 70% arcs having a high failure rate between \([80\%, 100\%]\), such that the average failure rate is 70% which has also been verified by the literature. For each instance, we generate 200 scenarios with equal probability 0.5% of realizing each scenario according to these arc-failure rates.

For the SMCF-VaR and SMCF-CVaR models of the RAKE problem, we test values of the reliability \(1 - \theta\) as 70%, 80%, 90%, and 99%. The threshold loss \(\eta\) is computed \textit{a priori} to solving all instances as follows. We first solve a deterministic RAKE problem that maximizes the total exchange utility without any arc failure. We call this computational scheme “MaxU” with an optimal exchange solution denoted by \(x_{\text{MaxU}}\). We then realize the uncertainty in each instance, and use Formulation (13a)–(13c) to compute the expected utility losses

\[
L_{\text{Max}} = \mathbb{E}_{\xi} \left[ L(x'_{\text{MaxU}}, Y'_{\xi}) \right] = \frac{1}{|\Omega|} \sum_{s \in \Omega} L(x'_{\text{MaxU}}, Y'_{\xi_s}).
\]  

Now consider another benchmark computational scheme, in which we seek a feasible exchange solution to maximize the total utility given by exchanges and meanwhile minimize the expected losses due to the uncertain compatibility, each with 50% weights. We call this scheme “MinEL” which solves

\[
\max_{x' \in \{0, 1\}^{|\mathcal{A}|}} \left\{ \sum_{s \in \Omega} w_{ij} x'_{ij} - \frac{1}{|\Omega|} \sum_{s \in \Omega} L(x', Y'_{\xi_s}) : \sum_{j: (i,j) \in \mathcal{A}} x'_{ij} - \sum_{j: (j,i) \in \mathcal{A}} x'_{ji} = 0, \sum_{j: (i,j) \in \mathcal{A}} x'_{ij} \leq 1, \forall i \in \mathcal{N} \right\}.
\]  

We denote its optimal objective (i.e., the minimum expected value of utility losses) by \(L_{\text{Min}}\). We set the threshold loss \(\eta\) as the middle point in \([L_{\text{Min}}, L_{\text{Max}}]\), i.e.,

\[
\eta = (L_{\text{Max}} + L_{\text{Min}})/2.
\]

We compute all randomly generated RAKE instances and compare the lengths of cycles in each optimal solution according to different settings of the models. Intuitively, we would prefer solutions containing shorter cycles as they generally have lower chances of including matches that could fail later. Abraham et al. (2007) consider a kidney exchange clearing problem which allows to use cycles up to length 3 and show that the problem is NP-hard. Saidman et al. (2006) and Roth et al. (2007) show that efficiency could be gained by incorporating larger cycle exchanges.

By using SMCF-VaR or SMCF-CVaR, we do not specify limits of the length of cycles used in our exchange solutions, which could help to improve the computation by avoiding combinatorial
structures of the problem. Meanwhile, we enforce optimal solutions to choose shorter cycles via the use of risk measures on utility losses. Table 5 provides averages of the average (Avg), maximum (Max), and minimum (Min) lengths of all cycles in optimal solutions to the ten instances we generate, solved by using different solution schemes indicated in each column.

Table 10: Comparison of cycle lengths given by optimal solutions using different approaches

<table>
<thead>
<tr>
<th></th>
<th>MaxU</th>
<th>MinEL</th>
<th>SMCF-VaR (1 − θ)</th>
<th>SMCF-CVaR (1 − θ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>70%</td>
<td>80%</td>
<td>90%</td>
<td>99%</td>
</tr>
<tr>
<td>Avg</td>
<td>10.4</td>
<td>2.7</td>
<td>4.8</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td></td>
<td>4.2</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td></td>
<td>4.8</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td></td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Max</td>
<td>18.2</td>
<td>4.2</td>
<td>6.8</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>4.8</td>
<td></td>
<td>4.4</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>4.4</td>
<td></td>
<td>2.4</td>
<td>2.0</td>
</tr>
<tr>
<td>Min</td>
<td>5.4</td>
<td>2.2</td>
<td>3.2</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td></td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td></td>
<td>2.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>

In Table 10, optimal solutions solved by using MaxU are the least conservative and contain significantly larger cycles compared with solutions to other models, whereas optimal exchange solutions to MinEL are the most conservative ones and all contain relatively small cycles. By using SMCF-VaR, we balance the total utility yielded by large-cycle exchanges and potential utility losses due to uncertain failure of any arc involved in the cycle solutions. Given the same reliability guarantee, SMCF-CVaR tends to result in more conservative and thus shorter-cycle solutions. For both SMCF-VaR and SMCF-CVaR, the optimal exchanges contain shorter cycles as we increase the reliability $1 − \theta$.

7 Conclusions

In this paper, we proposed a linear model to calculate possible path flow losses given a minimum cost flow solution subject to random arc failure, and also a polynomial algorithm to compute the minimum and the maximum losses. We used two types of risk measures, VaR and CVaR, to bound the minimum flow losses of an arc flow solution that solves MCF. We formulated SMCF-VaR as a mixed-integer program, and SMCF-CVaR as an LP model, both of which involved linear constraints for calculating the flow losses. We demonstrated formulations of the two SMCF problem variants on an application of risk-aware kidney exchange with random match failure after decisions of pairing kidneys have been made. We tested both SMCF-VaR and SMCF-CVaR on general network instances with diverse sizes and random arc failure probability, and reported computational results on both computational times and optimal values. In general, SMCF-CVaR yielded more conservative flow solutions that ensure much smaller flow losses with respect to the original reliability requirement. In an uncertain environment (e.g., with possible arc or node failures), both models can find risk-averse solutions by diversifying flow assignments (e.g., different paths) to ensure the successful passage of majority or all flow assignments with a high probability. Although this could
lead to a higher total cost, it keeps the risk of flow losses under control. Decision makers can use the proposed models to balance between the total flow cost and the risk of flow losses.

Future research tasks include developing algorithms for solving VaR and CVaR variants that bound the maximum flow losses $\mathcal{L}(x, Y_\xi)$ in SMCF, where the inner maximization problem cannot be simply eliminated. One idea is to take the dual of $\mathcal{L}(x, Y_\xi)$ that is a minimization problem, and solve SMCF-VaR or SMCF-CVaR by eliminating the “minimization” of the inner dual problem. However, the constraints in such a reformulation will contain bilinear terms “$x \times$” the dual variables associated with (2a)–(2d), which cannot be linearized due to the continuity of both variables $x$ and the dual variables. SMCF-VaR would then become both nonlinear and discrete (i.e., $z^* \in \{0, 1\}$), which is in general computationally intractable. We will investigate efficient algorithms for solving the $\mathcal{L}(x, Y_\xi)$-variant in our future research. Meanwhile, we are interested in developing approximation algorithms or meta-heuristics to derive good objective bounds for accelerating the computation of SMCF-VaR.

Acknowledgments: The authors are grateful to two anonymous reviewers, an associate editor, and the department editor, Dr. Shabir Ahmed, for their helpful suggestions that led to a much improved version of this paper. Dr. Zheng gratefully acknowledges the support from NASA WV Space Grant Consortium. Dr. Shen gratefully acknowledges the support of IBM Smarter Planet Innovation Faculty Award. This material is based upon work supported by the AFRL Mathematical Modeling and Optimization Institute.

References


