

Loss-constrained minimum cost flow under arc failure uncertainty with applications in risk-aware kidney exchange

Qipeng P. Zheng, Siqian Shen & Yuhui Shi

To cite this article: Qipeng P. Zheng, Siqian Shen & Yuhui Shi (2015) Loss-constrained minimum cost flow under arc failure uncertainty with applications in risk-aware kidney exchange, IIE Transactions, 47:9, 961-977, DOI: [10.1080/0740817X.2014.991476](https://doi.org/10.1080/0740817X.2014.991476)

To link to this article: <http://dx.doi.org/10.1080/0740817X.2014.991476>



Accepted author version posted online: 13 Jan 2015.
Published online: 13 Jan 2015.



Submit your article to this journal [↗](#)



Article views: 124



View related articles [↗](#)



View Crossmark data [↗](#)

Loss-constrained minimum cost flow under arc failure uncertainty with applications in risk-aware kidney exchange

QIPENG P. ZHENG¹, SIQIAN SHEN^{2,*} and YUHUI SHI²

¹Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL 32816, USA

²Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA
 E-mail: siqian@umich.edu

Received October 2012 and accepted November 2014

In this article, 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.

Keywords: Stochastic minimum cost flow, value-at-risk, conditional value-at-risk, Benders decomposition, risk-aware kidney exchange

1. Introduction

The Minimum Cost Flow (MCF) problem is fundamental to many classic network flow problems (Ahuja *et al.*, 1993). Consider a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, 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 arc (i, j) by C_{ij} , and a capacity of arc (i, j) by U_{ij} , for all $(i, j) \in \mathcal{A}$. Let decision variables x_{ij} be the amounts of flows on arcs $(i, j) \in \mathcal{A}$. We formulate a generic MCF problem as

$$\text{MCF: } \min \sum_{(i,j) \in \mathcal{A}} C_{ij} x_{ij} \quad (1a)$$

$$\text{s.t. } \sum_{j:(i,j) \in \mathcal{A}} x_{ij} - \sum_{j:(j,i) \in \mathcal{A}} x_{ji} = \begin{cases} D_i & \forall i \in \mathcal{S}, \\ 0 & \forall i \in \mathcal{N} \setminus \mathcal{S} \setminus \mathcal{T}, \\ -D_i & \forall i \in \mathcal{T}, \end{cases} \quad (1b)$$

$$0 \leq x_{ij} \leq U_{ij}, \quad \forall (i, j) \in \mathcal{A}, \quad (1c)$$

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 article, 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) and 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 *a priori* in an uncertain network environment and cannot re-route 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 immediately after the

*Corresponding author

occurrence of a catastrophic event, when detailed information of road failure and traffic congestion is not available. Or a cyber-attacker interferes with 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 article, 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 all the flows that 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 article, 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 article, 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 are of considerable interest 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), and 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), and Glockner *et al.* (2001) analyze stochastic dynamic network flow problems under uncertain arc capacities and develop 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 stochastic transportation network with random demands and arc capabilities, whereas Hudson and Kapur (1983), Lin (1998), and Lin (2007) study similar reliability problems of assigning feasible flows to meet uncertain demands given a budget of the total flow cost. In this article, we minimize the cost of fixed arc flows subject to a 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) and 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 are 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 (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 article we use CVaR as one of popular coherent risk measures and as a comparison to VaR for analyzing the SMCF problem.

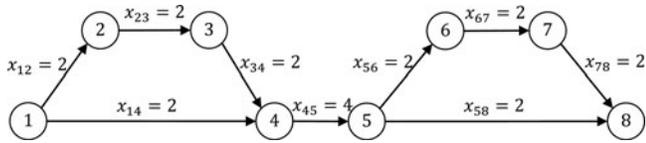


Fig. 1. Illustrating multiple values of path flow losses yielded by one arc flow solution.

1.3. Contributions and organization of this paper

The contributions of the article 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 article 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 article 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



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

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 $C_{ij} > 0, \forall(i, j) \in \mathcal{A}$, 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 the example depicted in Fig. 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) fail. Then two units of losses are incurred if we transport two units of flow via path “P1: 1–2–3–4–5–6–7–8” and the other two units via path “P2: 1–4–5–8.” However, all four units of flow will be completely gone if we ship two units of flow via path “P3: 1–2–3–4–5–8,” and the other two units via path “P4: 1–4–5–6–7–8.”

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 model for computing all possible losses, and Section 2.2 develops polynomial algorithms for computing the maximum and the minimum flow losses.

2.1. An LP model for computing flow losses

Given an arc flow solution $\hat{x} = [\hat{x}_{ij}, (i, j) \in \mathcal{A}]^T$, we construct a Reformulated Residual Network (RRN) $\mathcal{G}(\hat{x}) = (\mathcal{N}, \bar{\mathcal{A}})$ with a new arc set $\bar{\mathcal{A}}$, for which we delete all arcs in \mathcal{A} that are currently with zero flow, create a new arc (i, j) in $\bar{\mathcal{A}}$ for each arc $(j, i) \in \mathcal{A}$ that has $\hat{x}_{ji} > 0$, and designate \hat{x}_{ji} as the capacity of the new arc.

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 $\mathcal{G}(\hat{x})$ by deleting all zero-flow arcs in \mathcal{A} and reversing all positive-flow arcs in \mathcal{A} . Figure 3 illustrates the corresponding RRN $\mathcal{G}(\hat{x})$ for path $s \rightarrow t$ in Fig. 2, given that flows are all positive in the path.



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

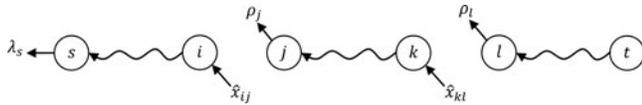


Fig. 4. The RRN $\mathcal{G}(\hat{x}, Y)$ of path $s \rightarrow t$ in Fig. 2 given $Y_{ij} = Y_{kl} = 1$.

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

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

Consider network $\mathcal{G}(\hat{x}, Y)$, where $Y = [Y_{ij}, (i, j) \in \mathcal{A}]^T$, as a modification of $\mathcal{G}(\hat{x})$, 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 ρ_j at node j . We also add a demand variable λ_s as the accumulated withdrawn of flows at node s . Figure 4 illustrates the corresponding $\mathcal{G}(\hat{x}, Y)$ for the $\mathcal{G}(\hat{x})$ in Fig. 3 given $Y_{ij} = Y_{kl} = 1$.

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 $(i, j) \in \bar{\mathcal{A}}$, which is equal to the withdrawn flow on arc $(j, i) \in \mathcal{A}$. Recall that set \mathcal{S} contains all supply nodes in \mathcal{G} . The following LP model maintains flow balances at all nodes in $\mathcal{G}(x, Y)$:

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

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

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

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

Constraints (2a) balance flows at all nodes in set \mathcal{S} and all other nodes in $\mathcal{N} \setminus \mathcal{S}$, where the latter do not accumulate withdrawn path flows reflected by the values of $\lambda_i, \forall i \in \mathcal{S}$. Constraints (2b) ensure $f_{ij} = 0$ for any arc $(i, j) \in \bar{\mathcal{A}}$ if arc $(j, i) \in \mathcal{A}$ fails in \mathcal{G} , and restrict f_{ij} on other arcs $(i, j) \in \bar{\mathcal{A}}$ by the values of x_{ji} otherwise. The accumulated value of flow losses at any node $i \in \mathcal{S}$ is bounded by its original supply D_i according to constraints (2c), while constraints (2d) bound the values of variable demands $\rho_i, \forall i \in \mathcal{N}$ by the total amount of flow inputs on failed arcs adjacent to node i in \mathcal{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 , $L(x, Y) = \sum_{i \in \mathcal{S}} \lambda_i$, where $\lambda = [\lambda_i, i \in \mathcal{S}]^T$ 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 $(i, j) \in \mathcal{A}$ 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 $i \in \mathcal{N} \setminus \mathcal{S}$, according to constraints (2a), the original outgoing flows on its failed adjacent arcs are accumulated (i.e., $\sum_{j:(i,j) \in \mathcal{A}} Y_{ij} x_{ij}$) and subtracted by the amount of variable demand ρ_i (created if $Y_{ij} = 1$ exists for any $(i, j) \in \mathcal{A}$), to generate a flow-withdrawn supply at node i . For any nodes in $\mathcal{N} \setminus \mathcal{S}$ 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 ρ_i in constraints (2a) will carry the difference of accumulated flow withdrawn and the output capacities at each node $i \in \mathcal{N}$, 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 λ_i at origin i of the corresponding path according to constraints (2a), for all $i \in \mathcal{S}$, and is bounded by the total supply D_i according to constraints (2c). The summation of all positive withdrawn flows λ_i at all nodes $i \in \mathcal{S}$ 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., $L(x, Y) \leq \sum_{(i,j) \in \mathcal{A}} Y_{ij} x_{ij}$. This is because the variables λ_i are only linked to the failed arcs that are adjacent to nodes $i \in \mathcal{S}$ in constraints (2a). The difference between $L(x, Y)$ and $\sum_{(i,j) \in \mathcal{A}} Y_{ij} x_{ij}$ 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 Fig. 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 Fig. 1 under the assumption that both arcs (2, 3) and (6, 7) fail,

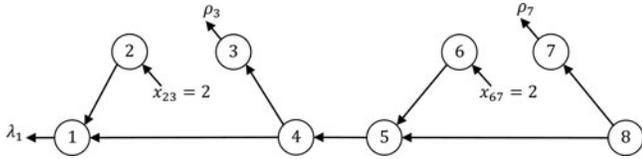


Fig. 5. The RRN of the graph in Fig. 1 and two possible values of $L(x, Y)$.

resulting in complete flow losses on paths $\mathcal{P}1$, $\mathcal{P}3$, and $\mathcal{P}4$. Via the LP formulation (2), we can also obtain solutions in the two cases with respectively 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) $f_{87}^2 = f_{76}^2 = f_{32}^2 = f_{85}^2 = f_{43}^2 = 0$, $f_{65}^2 = f_{54}^2 = f_{41}^2 = f_{21}^2 = 2$, $\rho_7^2 = 0$, $\rho_3^2 = 0$, $\lambda_1^2 = 4$.

2.2. Algorithms for computing the maximum and the minimum flow losses

The SMCF problems studied in this article 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 $\underline{L}(x, Y) = \min_{\lambda, \rho, f} \{L(x, Y) | (2a)-(2d)\}$, and $\bar{L}(x, Y) = \max_{\lambda, \rho, f} \{L(x, Y) | (2a)-(2d)\}$. We construct two algorithms in particular for computing $\underline{L}(x, Y)$ and $\bar{L}(x, Y)$ in polynomial time.

For any MCF solution x , define sets $\bar{\mathcal{S}}$ and $\bar{\mathcal{T}}$ as follows:

$$\bar{\mathcal{S}} \equiv \left\{ i \in \mathcal{N} \mid \sum_{j:(i,j) \in \mathcal{A}} Y_{ij} x_{ij} > 0 \right\} \text{ and}$$

$$\bar{\mathcal{T}} \equiv \left\{ i \in \mathcal{N} \mid \sum_{j:(j,i) \in \mathcal{A}} Y_{ji} x_{ji} > 0 \right\},$$

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 (λ, ρ, f) to Formulation (2) by rerouting flows from nodes in $\bar{\mathcal{S}}$ to nodes in \mathcal{M} , where set \mathcal{M} can either be set \mathcal{S} or be set $\bar{\mathcal{T}}$, after knowing the failed arcs indicated by parameter Y .

We demonstrate the steps in Algorithm 1 by using the example in Fig. 5, where $\bar{\mathcal{T}} = \{3, 7\}$ and $\bar{\mathcal{S}} = \{2, 6\}$. The solution (λ^1, ρ^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 (λ^2, ρ^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.

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

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

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

$$\min_{\lambda, \rho, f} \left\{ \sum_{i \in \bar{\mathcal{S}}} \lambda_i : (2a)-(2d) \right\} \Leftrightarrow \sum_{(i,j) \in \mathcal{A}} Y_{ij} x_{ij} - \max_{\lambda, \rho, f} \left\{ \sum_{i \in \mathcal{N}} \rho_i : (2a)-(2d) \right\}. \quad (3)$$

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

3. Models and algorithms of SMCF-VaR

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

$$\text{SMCF-VaR} : \min \left\{ \sum_{(i,j) \in \mathcal{A}} C_{ij} x_{ij} : (1b), (1c), \right. \\ \left. \mathbb{P}\{L(x, Y_\xi) \leq \eta\} \geq 1 - \theta \right\}, \quad (4)$$

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

Remark 2. As $L(x, Y_{\xi^s})$ may take multiple values even for fixed solution x and realization Y_{ξ^s} , decision-makers may be interested in bounding the VaR of $\bar{L}(x, Y_\xi)$ or $\underline{L}(x, Y_\xi)$, respectively reflecting “risk-averse” and “risk-seeking” behavior against the uncertainty. Problem SMCF-VaR is in fact equivalent to its $\underline{L}(x, Y_\xi)$ -variant that possesses a probabilistic constraint $\mathbb{P}\{\underline{L}(x, Y_\xi) \leq \eta\} \geq 1 - \theta$, because $L(x, Y_{\xi^s})$ 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 η is minimized. This result cannot be generalized to the $\bar{L}(x, Y_\xi)$ -variant of SMCF-VaR, which needs to be solved as a bilevel program where the inner problem $\bar{L}(x, Y_\xi)$ is a maximization problem given fixed x and Y_ξ . The $\bar{L}(x, Y_\xi)$ -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 article, we focus on SMCF-VaR and later SMCF-CVaR that restrict the risk associated with the minimum flow losses $\underline{L}(x, Y_\xi)$ and will discuss future research in Section 7 for tackling bilevel SMCF problems involving $\bar{L}(x, Y_\xi)$.

3.1. An integer programming reformulation of SMCF-VaR

We reformulate SMCF-VaR as a mixed-integer program for Y_ξ with finite realizations by defining an additional binary variable associated with each realized scenario. Let Prob_{ξ^s} be the probability of realizing Y_{ξ^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 η , and $z^s = 0$ otherwise. The SMCF-VaR in Formulation (4) is equivalent to a deterministic model:

SMCF-VaR-D:

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

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

Remark 3. The number of binary variables in constraint (5) (as well as models discussed in the rest of this article) depends on the cardinality of set Ω . In general, Monte Carlo sampling (e.g., Norkin *et al.* (1998) and Mak *et al.* (1999)) is a common approach for generating independent scenarios from known distributions of the uncertainty. Shapiro and Homem-de-Mello (2000) propose a Sample Average Approximation (SAA) approach for solving stochastic linear programs, and Kleywegt *et al.* (2002) extend the approach to stochastic discrete optimization problems. In particular, Luedtke and Ahmed (2008) approximate optimization problems with probabilistic constraints by using SAA and provide sufficient numbers of samples and scenarios to derive lower bounds or feasible solutions with certain confidence levels. In this article, we focus on solving SMCF-VaR given set Ω and realizations Y_{ξ^s} with their probabilities Prob_{ξ^s} , $\forall s \in \Omega$. Our approaches can be integrated into an SAA implementation that follows the methods provided in Luedtke and Ahmed (2008) for generating independent samples and set Ω 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 η after knowing an MCF solution x and Y_{ξ^s} . We consider the Benders decomposition approach (Benders, 1962; Birge and Louveaux, 1997), 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 that decides an MCF solution x and subproblems that 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

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

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

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

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 $\mathbb{F}(x, p^s) \geq 0$ 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 $\mathbb{F}(x, p^s) \geq 0$ in each subproblem s , we use an LP relaxation of constraint (5) and its optimal linear duals to derive coefficients of the Benders cuts.

3.2.1. 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 $\text{ALG}(\overline{T})$ 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, allows us to solve 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.

4. Models and algorithms of SMCF-CVaR

We also study a coherent risk measure, CVaR_θ , to derive a solution that is feasible to SMCF-VaR for the same risk parameter θ . In general, CVaR_θ is known as a convex approximation of VaR_θ , which measures the expected

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 $\mathcal{F}(\hat{x}, \hat{p})$ as the set of cutting planes to be generated based on solution (\hat{x}, \hat{p}) .
- 3: Set $\mathcal{F}(\hat{x}, \hat{p}) = \emptyset$.
- 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 $\mathbb{F}(x, p^s) \geq 0$ by using strong/weak optimality conditions.
- 8: Update $\mathcal{F}(\hat{x}, \hat{p}) = \mathcal{F}(\hat{x}, \hat{p}) \cup \{\mathbb{F}(x, p^s) \geq 0\}$.
- 9: **end if**
- 10: **end for**
- 11: **if** $\mathcal{F}(\hat{x}, \hat{p}) \neq \emptyset$ **then**
- 12: Generate all cuts in $\mathcal{F}(\hat{x}, \hat{p})$ to the relaxed master problem (6). Go to Step 2.
- 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**

random amount of a random variable exceeding its VaR for a given risk level θ . 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 θ (equivalent to its $\underline{L}(x, Y_\xi)$ -variant) as

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

$$\sum_{i \in S} \lambda_i^s \leq b_{\xi^s} + \alpha \quad \forall s \in \Omega \quad (8b)$$

$$\alpha \geq 0, \quad b_{\xi^s} \geq 0 \quad \forall s \in \Omega \quad (8c)$$

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

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

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

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_{ξ^s} , consider subproblems

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

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 $\sum_{i \in S} \hat{\lambda}_i^s$ 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

$$b_s + \alpha \geq \sum_{(i,j) \in \mathcal{A}} Y_{ij}^s (\hat{\phi}_i^s + \hat{\tau}_j^s) x_{ij} + \sum_{(i,j) \in \bar{\mathcal{A}}} (1 - Y_{ji}^s) \hat{\gamma}_{ij}^s x_{ji} + \sum_{i \in S} D_i \hat{\zeta}_i^s, \quad (11)$$

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

An optimal solution to SCMF-CVaR $_{\theta}$ provides a feasible solution to the SMCF-VaR $_{\theta}$ for the same risk threshold value θ . The former is much easier to optimize as a linear program. Therefore, for solving SMCF-VaR as constraint (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 the computational time.

5. Applications in risk-aware kidney exchange

In this section, we demonstrate how to use the developed SMCF-VaR and SMCF-CVaR to solve a class of risk- and

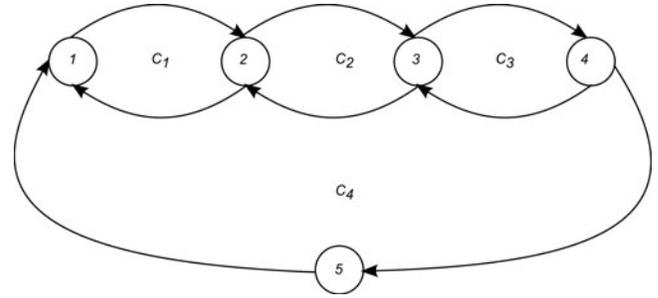


Fig. 6. Network encoding of a kidney exchange.

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 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 $\mathcal{G}(\mathcal{N}, \mathcal{A})$ considered in this article 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.

Figure 6 illustrates an example of the network representation of a case with five donor-patient pairs. The graph contains four 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 problem MCF as binary variables x'_{ij} , for all $(i, j) \in \mathcal{A}$, such that

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

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

$$\max \sum_{(i,j) \in \mathcal{A}} w_{ij} x'_{ij} \tag{12a}$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in \mathcal{A}} x'_{ij} - \sum_{j:(j,i) \in \mathcal{A}} x'_{ji} = 0 \quad \forall i \in \mathcal{N} \tag{12b}$$

$$\sum_{j:(i,j) \in \mathcal{A}} x'_{ij} \leq 1 \quad \forall i \in \mathcal{N} \tag{12c}$$

$$x'_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}, \tag{12d}$$

where the objective (12a) maximizes the total utility yielded by successful exchanges of kidneys; constraints (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 constraints (12c), which indicate that no more than one kidney is given out by the donor of pair i , for each node $i \in \mathcal{N}$. 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 \mathcal{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 $\mathcal{G}(\mathcal{N}, \mathcal{A})$. 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 his/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

in both 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} otherwise, for all $(i, j) \in \mathcal{A}$. The set Ω now specifies a finite number of realizations of the uncertain $Y = [Y_{ij}, (i, j) \in \mathcal{A}]^T$ and Y_{ξ}^s 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 $\mathcal{G}(x', Y)$ as before given an exchange solution x' and random vector Y . Figure 7 illustrates the RRN of a four-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.

We simplify 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 $\bar{\mathcal{A}}$ and f_{ij} to indicate the set of arcs in the RRN and the amount of withdrawn flows on arc $(i, j) \in \bar{\mathcal{A}}$, respectively, and formulate

$$L(x', Y) = \sum_{(j,i) \in \mathcal{A}} Y_{ji} w_{ji} + \min_f \sum_{(i,j) \in \bar{\mathcal{A}}} w_{ij} f_{ij}, \tag{13a}$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in \bar{\mathcal{A}}} f_{ij} - \sum_{j:(j,i) \in \bar{\mathcal{A}}} f_{ji} = \sum_{j:(i,j) \in \mathcal{A}} Y_{ij} x'_{ij} - \sum_{j:(j,i) \in \mathcal{A}} Y_{ji} x'_{ji}, \quad \forall i \in \mathcal{N}. \tag{13b}$$

$$0 \leq f_{ij} \leq (1 - Y_{ji}) x'_{ji}, \quad \forall (i, j) \in \bar{\mathcal{A}}, \tag{13c}$$

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} , $\forall (i, j) \in \bar{\mathcal{A}}$); constraints (13b) represent flow conservation constraints, where the withdrawn supply at a node i is one 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) , $\forall (i, j) \in \bar{\mathcal{A}}$ 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

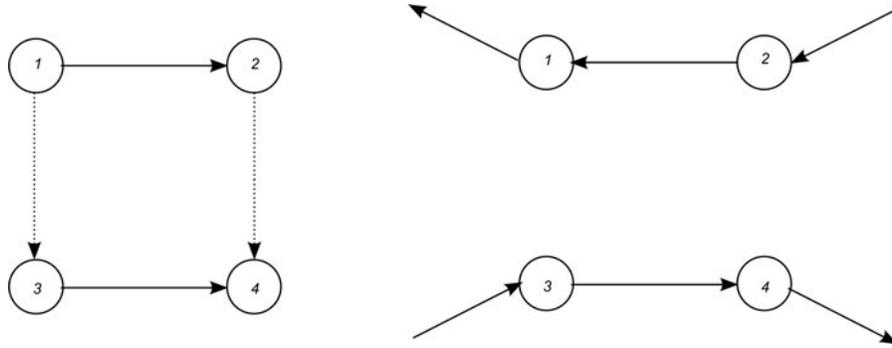


Fig. 7. A four-way exchange solution and the corresponding RRN.

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., Fig. 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 MCF 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'_ξ to a SMCF-VaR setting and formulate

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

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 formulation (14) into a mixed-integer program given the finite support set Ω of ξ . 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, in particular emphasize computational-time savings by formulating path flow losses via linear constraints, rather than using 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, whereas 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 to solve 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 article. 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 $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ in different sizes with 10, 20, and 30 nodes. For every node $i \in \mathcal{N}$, we randomly pick an integer between the minimum and the maximum outgoing degrees required for every node in \mathcal{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 .

Table 1. The [minimum, maximum] outgoing degree assumptions

Node	Low density	Medium density	High density
10	[2, 4]	[3, 6]	[5, 8]
20	[3, 6]	[5, 12]	[10, 15]
30	[3, 6]	[7, 15]	[12, 21]

We then repeatedly generate nodes adjacent to node i from all nodes in $\mathcal{N} \setminus \{i\}$ 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 connecting 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 \mathcal{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 \mathcal{S} equals to the sum of demands at all nodes in \mathcal{D} for each instance.

In this article, 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., $\text{Prob}_{\xi^s} = 1/|\Omega|$. 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 a 0.25 failure probability) and more reliable (i.e., the ones generated using a 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 θ and η values we test. To address this issue, we adopt the following rules to keep all instances feasible in our numerical experiments. For any given θ and η , 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 $\eta = 1, \dots, 10$. The resulting CPU time is approximately the same for each instance for different η -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 the average CPU seconds of the 10 instances with $1 - \theta$ being $> 50\%$ and $\leq 70\%$.) In these tables, $|\mathcal{N}|$ -l, -m, and -h indicate graph instances with low, medium, and high densities.

For both Tables 2 and 3, we provide the following observations.

1. CPU time increases as we increase graph sizes and density levels, but the time increase of solving the SMCF-VaR is much more significant.
2. CPU time decreases as we increase $1 - \theta$, i.e., when both VaR and CVaR seek more robust solutions but with higher cost.
3. Solving SMCF-CVaR is much quicker than solving SMCF-VaR.
4. 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 \mathcal{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 10 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.

Moreover, for both SMCF-VaR and SMCF-CVaR, their feasible regions are nondecreasing with respect to values of reliability $1 - \theta$ and tolerable losses η . The results also show that the optimal objective values in both models are very sensitive to changes made to $1 - \theta$ and η . 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 (η).

Table 2. CPU seconds for solving instances with arc failure rate = 0.25

<i>Inst.</i>	<i>SMCF-VaR</i>				<i>SMCF-CVaR</i>			
	=50%	50–70%	75–90%	90–99%	=50%	50–70%	70–90%	90–99%
10-l	1.08	0.52	0.33	0.21	0.15	0.12	0.11	0.10
10-m	5.03	4.86	4.25	0.54	0.27	0.26	0.22	0.19
10-h	7.96	22.25	15.74	2.82	0.41	0.43	0.38	0.34
20-l	92.83	187.44	61.70	3.23	1.45	1.34	1.07	0.79
20-m	857.70	986.27	379.32	16.80	3.26	3.05	2.59	2.23
20-h	161.55	338.32	576.64	33.80	3.81	4.01	3.55	3.30
30-l	3200.48	3440.52	201.14	3.66	2.91	2.62	1.31	0.57
30-m	2435.25	4256.39	1187.17	194.20	13.53	13.00	12.33	12.47
30-h	1844.21	7456.36	4408.97	231.82	17.89	18.45	17.50	17.51

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

<i>Inst.</i>	<i>SMCF-VaR</i>				<i>SMCF-CVaR</i>			
	=50%	50–70%	70–90%	90–99%	=50%	50–70%	70–90%	90–99%
10-l	0.23	0.61	0.37	0.23	0.11	0.10	0.09	0.08
10-m	0.20	0.34	0.82	0.50	0.14	0.12	0.11	0.12
10-h	0.26	0.52	1.92	0.92	0.19	0.19	0.16	0.14
20-l	2.54	26.44	22.75	2.46	0.76	0.74	0.56	0.48
20-m	0.63	1.20	2.54	4.54	0.79	0.89	1.02	0.84
20-h	0.95	1.54	18.17	7.94	1.42	1.37	1.25	1.05
30-l	90.92	197.86	319.71	10.97	2.30	2.06	1.72	0.95
30-m	1.60	3.07	51.36	63.60	7.99	8.75	7.07	6.06
30-h	26.28	157.31	407.46	97.35	15.04	15.89	15.12	14.79

Table 4. CPU seconds of 20-m instances with means of demand = 20, 50, 100

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

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

	<i>Inst.</i>	<i>20-nc</i>	<i>20-c</i>	<i>50-nc</i>	<i>50-c</i>	<i>100-nc</i>	<i>100-c</i>
<i>SMCF-VaR</i>	=50%	582.8	582.8	2142.6	2142.6	4943.0	4943.0
	50–70%	790.8	790.8	2955.5	2955.5	6829.5	6829.5
	70–90%	1281.5	1281.5	4670.3	4670.3	10 610.2	10 600.6
	90–99%	1802.1	1802.1	6472.2	6472.2	14 668.2	14 603.2
<i>SMCF-CVaR</i>	=50%	1663.3	1663.3	6183.6	6183.6	14 490.4	14 419.8
	50–70%	1795.7	1795.7	6587.5	6587.5	15 187.2	15 108.6
	70–90%	1996.5	1996.5	7107.9	7107.9	16 137.2	16 013.2
	90–99%	2020.9	2020.9	7139.1	7139.1	16 195.2	16 060.7

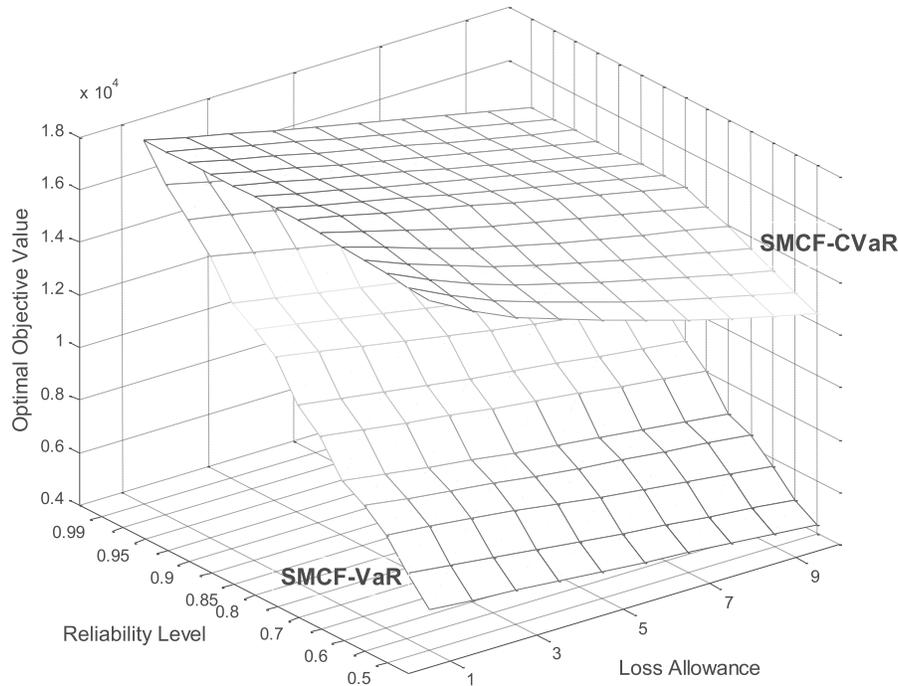


Fig. 8. Comparison of optimal objective values between SMCF-VaR and SMCF-CVaR.

In Fig. 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 the parameters $1 - \theta$ and η . This is because, for SMCF-VaR, there exist binary variables that cause the nonconvexity issue, and for SMCF-CVaR, the risk parameter θ 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 η , but SMCF-VaR is more sensitive to changes of the reliability $1 - \theta$.

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_θ , yielding a feasible solution and an objective upper bound for the latter. In Table 6, we report objective values of $(CVaR - VaR) / VaR \times 100\%$, 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 $(CVaR - VaR) / VaR \times 100\%$ values of all 20-h instances with arc failure rate = 0.1. Table 7 depicts results for $1 - \theta = 50\%, 55\%, \dots, 95\%, 99\%$ (indicated in the first column) and loss tolerance $\eta = 1, \dots, 10$ (indicated in the first row). The solution gaps are smaller when η is larger given any $1 - \theta$ value. For each fixed tolerance η , 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 η , and increase as η 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 $\eta = 7, 8, 9$, and 10.

Finally, we check solution robustness and qualities guaranteed by SMCF-CVaR. In Table 8, for every demand-loss allowance η , each column depicts the real demand losses (i.e., α defined in CVaR) guaranteed by the corresponding SMCF-CVaR given $1 - \theta$. We also fix optimal MCF solutions from the CVaR models, and check the satisfaction probabilities associated with the VaR constraints yielded by optimal solutions to SMCF-CVaR given fixed values of $1 - \theta$ and η . 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 of 80.67%. If the required reliability is 99%, there are 10 units of flows lost and the satisfied reliability

Table 6. $(CVaR-VaR)/VaR \times 100\%$ values (in %) for different values of $1 - \theta$ and arc failure rates

Arc failure rate	Instances	=50%	50–70%	70–90%	90–99%	=99%
0.25	20-l	159.53	119.51	68.85	32.54	0.00
	20-m	183.57	133.53	59.55	13.99	0.00
	20-h	36.54	36.23	25.64	9.63	0.00
	30-l	117.43	93.03	87.07	11.24	0.00
	30-m	89.26	42.7	26.19	5.55	0.00
	30-h	91.38	68.66	19.91	6.63	0.00
0.1	20-l	64.18	57.84	42.73	19.32	0.00
	20-m	12.62	18.03	31.57	16.35	0.00
	20-h	8.50	10.75	10.96	3.57	0.00
	30-l	114.67	84.98	40.09	41.51	0.00
	30-m	52.85	66.67	53.22	10.47	0.00
	30-h	24.88	28.87	22.44	3.76	0.00

is 100.00% as no scenarios are violated. In general, when $1 - \theta$ and η 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 $\mathcal{G}(\mathcal{N}, \mathcal{A})$ as 50, 100, and 200; i.e., $|\mathcal{N}| = 50, 100, 200$. We randomly generate 10 instances for each size of the graph with each node having an outgoing degree between $0.04|\mathcal{N}|$ and $0.12|\mathcal{N}|$; 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 η is computed *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'_{MaxU} . We then realize the uncertainty in each instance, and use Equations (13a)–(13c) to compute the expected utility losses:

$$L_{\text{Max}} = \mathbb{E}_{\xi} [L(x'_{\text{MaxU}}, Y'_{\xi})] = \frac{1}{|\Omega|} \sum_{s \in \Omega} L(x'_{\text{MaxU}}, Y'_{\xi^s}). \quad (15)$$

Now consider another benchmark computational scheme, in which we seek a feasible exchange solution to maxi-

Table 7. $(CVaR-VaR)/VaR \times 100\%$ values (in %) of instances 20-h with arc failure rate = 0.1

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

Table 8. Real demand losses (α) guaranteed by SMCF-CVaR for various η and $1 - \theta$

$1 - \theta$ (%)	1	2	3	4	5	6	7	8	9	10
50	0.00	0.04	0.35	0.71	1.39	2.03	2.79	3.62	4.35	5.23
55	0.00	0.07	0.41	0.84	1.39	2.21	2.95	3.81	4.62	5.37
60	0.00	0.10	0.44	1.03	1.59	2.36	3.12	4.01	4.85	5.60
65	0.02	0.18	0.57	1.12	1.81	2.75	3.54	4.43	5.20	6.17
70	0.03	0.30	0.75	1.39	2.19	2.96	4.00	5.01	6.04	7.04
75	0.11	0.60	1.21	1.91	2.51	3.58	4.60	5.47	6.45	7.37
80	0.64	1.48	2.41	3.19	3.98	4.94	5.81	6.62	7.43	8.52
85	0.82	1.82	2.80	3.76	4.70	5.62	6.44	7.44	8.39	9.37
90	0.95	1.93	2.90	3.90	4.86	5.77	6.60	7.50	8.59	9.65
95	1.00	2.00	3.00	4.00	5.00	5.96	6.86	7.73	8.62	9.41
99	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00

mize 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” that solves:

$$\max_{x' \in \{0,1\}^{|\mathcal{A}|}} \left\{ \sum_{s \in \Omega} w_{ij} x'_{ij} - \frac{1}{|\Omega|} \sum_{s \in \Omega} L(x', Y_{\xi^s}) : \sum_{j:(i,j) \in \mathcal{A}} x'_{ij} - \sum_{j:(j,i) \in \mathcal{A}} x'_{ji} = 0, \sum_{j:(i,j) \in \mathcal{A}} x'_{ij} \leq 1, \forall i \in \mathcal{N} \right\}. \quad (16)$$

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

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

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 10 provides averages of the average (Avg), maximum (Max), and minimum (Min) lengths of all cycles in optimal solutions to the 10 instances we generate, solved by using different solution schemes indicated in each column.

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 $1 - \theta$.

Table 9. Real successful probabilities guaranteed by SMCF-CVaR for various η and $1 - \theta$

$1 - \theta$ (%)	1	2	3	4	5	6	7	8	9	10
50	90.67	87.33	83.33	83.67	83.00	80.67	79.33	80.00	80.67	80.67
55	91.00	88.00	86.67	85.33	85.00	85.00	84.33	82.67	84.00	84.00
60	93.00	89.33	88.00	86.67	86.67	85.33	86.00	85.67	84.67	86.00
65	93.00	90.67	89.33	88.00	88.00	87.33	87.33	86.67	87.33	87.67
70	93.33	91.67	92.00	91.00	89.33	90.00	90.67	90.00	90.00	90.67
75	93.33	93.00	92.67	92.00	91.33	91.33	90.33	91.33	90.67	90.67
80	99.33	98.00	97.33	96.00	96.00	95.00	95.67	95.00	93.00	93.67
85	100.00	100.00	100.00	100.00	99.33	99.33	98.67	98.00	98.00	99.33
90	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
95	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
99	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

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

	<i>MaxU</i>	<i>MinEL</i>	<i>SMCF-VaR</i> ($1 - \theta$)				<i>SMCF-CVaR</i> ($1 - \theta$)			
			70%	80%	90%	99%	70%	80%	90%	99%
Avg	10.4	2.7	4.8	3.8	3.2	2.2	4.2	3.6	2.6	2.2
Max	18.2	4.2	6.8	5.0	4.8	2.8	6.4	4.4	4.0	2.4
Min	5.4	2.2	3.2	2.2	2.0	2.0	3.0	2.0	2.0	2.0

7. Conclusions

In this article, we proposed a linear model to calculate possible path flow losses given a MCF 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 RAKE 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 $\bar{L}(x, Y_\xi)$ in SMCF, where the inner maximization problem cannot be simply eliminated. One idea is to take the dual of $\bar{L}(x, Y_\xi)$ 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 constraints (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., $z^s \in \{0, 1\}$), which is in general computationally intractable. We will investigate efficient algorithms for solving the $\bar{L}(x, Y_\xi)$ -variant in our future research. Meanwhile, we are interested in developing approximation algorithms

or metaheuristics to derive good objective bounds for accelerating the computation of SMCF-VaR.

Acknowledgements

The authors are grateful to two anonymous reviewers, an Associate Editor, and the Department Editor for their helpful suggestions that led to a much improved version of this article.

Funding

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

- Abraham, D.J., Blum, A., and Sandholm, T. (2007) Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges, in *Proceedings of the Eighth ACM Conference on Electronic Commerce, EC '07*, ACM, Philadelphia, pp. 295–304.
- Ahmadi-Javid, A. (2012) Entropic value-at-risk: a new coherent risk measure. *Journal of Optimization Theory and Applications*, **155**(3), 1105–1123.
- Ahuja, R., Magnanti, T.L. and Orlin, J.B. (1993) *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, NJ.
- Alexander, G.J. and Baptista, A.M. (2004) A comparison of VAR and CVaR constraints on portfolio selection with the mean-variance model. *Management Science*, **50**(9), 1261–1273.
- Balcik, B., Beamon, B.M. and Smilowitz, K. (2008) Last mile distribution in humanitarian relief. *Journal of Intelligent Transportation Systems*, **12**(2), 51–63.
- Benders, J.F. (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, **4**(1), 238–252.
- Birge, J.R. and Louveaux, F.V. (1997) *Introduction to Stochastic Programming*, Springer, New York, NY.
- Boginski, V.L., Commander, C.W. and Turko, T. (2009) Polynomial-time identification of robust network flows under uncertain arc failures. *Optimization Letters*, **3**, 461–473.
- Boyles, S.D. and Waller, S.T. (2010) A mean-variance model for the minimum cost flow problem with stochastic arc costs. *Networks*, **56**(3), 215–227.

- Cheung, R.K. and Powell, W.B. (1996) An algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic fleet management. *Operations Research*, **44**, 951–963.
- Chun, S.Y., Shapiro, A. and Uryasev, S. (2012) Conditional value-at-risk and average value-at-risk: estimation and asymptotics. *Operations Research*, **60**(4), 739–756.
- Dickerson, J.P., Procaccia, A.D. and Sandholm, T. (2013) Failure-aware kidney exchange, in *Proceedings of the 14th ACM Conference on Electronic Commerce*, ACM, Philadelphia, pp. 323–340.
- Eiger, A., Mirchandani, P.B. and Soroush, H. (1985) Path preferences and optimal paths in probabilistic networks. *Transportation Science*, **19**(1), 75–84.
- Fan, Y.Y., Kalaba, R.E. and Moore II, J.E.M. (2005) Shortest paths in stochastic networks with correlated link costs. *Computers and Mathematics with Applications*, **49**, 1549–1564.
- Föllmer, H. and Knispel, T. (2011) Entropic risk measures: coherence vs. convexity, model ambiguity and robust large deviations. *Stochastics and Dynamics*, **11**, 333–351.
- Glockner, G.D. and Nemhauser, G.L. (2000) A dynamic network flow problem with uncertain arc capacities: formulation and problem structure. *Operations Research*, **48**(2), 233–242.
- Glockner, G.D., Nemhauser, G.L. and Tovey, C.A. (2001) Dynamic network flow with uncertain arc capacities: decomposition algorithm and computational results. *Computational Optimization and Applications*, **18**(3), 233–250.
- Hudson, J.C. and Kapur, K.C. (1983) Reliability analysis for multistate systems with multistate components. *IIE Transactions*, **15**(2), 127–135.
- Hutson, K.R. and Shier, D.R. (2009) Extended dominance and a stochastic shortest path problem. *Computers & Operations Research*, **36**, 584–596.
- Kleywegt, A.J., Shapiro, A. and Homem-de-Mello, T. (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, **12**(2), 479–502.
- Lin, J.-S. (1998) Reliability evaluation of capacitated-flow network with budget constraint. *IIE Transactions*, **30**(12), 1175–1180.
- Lin, Y.-K. (2007) Reliability evaluation for an information network with node failure under cost constraint. *IEEE Transactions on, Systems, Man and Cybernetics, Part A: Systems and Humans*, **37**(2), 180–188.
- Loui, R.P. (1983) Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, **26**, 670–676.
- Luedtke, J. and Ahmed, S. (2008) A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, **19**(2), 674–699.
- Mak, W.K., Morton, D.P. and Wood, R.K. (1999) Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, **24**(1), 47–56.
- Miller, B.L. and Wagner, H.M. (1965) Chance constrained programming with joint constraints. *Operations Research*, **13**(6), 930–945.
- Norkin, V.I., Pflug, G.C. and Ruszczyński, A. (1998) A branch and bound method for stochastic global optimization. *Mathematical Programming*, **83**(1), 425–450.
- Özdamar, L., Ekinci, E. and Küçükyazici, B. (2004) Emergency logistics planning in natural disasters. *Annals of Operations Research*, **129**(1–4), 217–245.
- Powell, W.B. and Cheung, R.K.-M. (1994) A network recourse decomposition method for dynamic networks with random arc capacities. *Networks*, **24**, 369–384.
- Prékopa, A. (1970) On probabilistic constrained programming, in *Proceedings of the Princeton Symposium on Mathematical Programming*, Princeton University Press, Princeton, NJ, pp. 113–138.
- Prékopa, A. and Boros, E. (1991) On the existence of a feasible flow in a stochastic transportation network. *Operations Research*, **39**(1), 119–129.
- Rockafellar, R.T. and Uryasev, S. (2000) Optimization of conditional value-at-risk. *Journal of Risk*, **2**(3), 21–41.
- Rockafellar, R.T. and Uryasev, S.P. (2002) Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, **26**(7), 1443–1471.
- Roth, A.E., Sönmez, T. and Ünver, M.U. (2004) Kidney exchange. *The Quarterly Journal of Economics*, **119**(2), 457–488.
- Roth, A.E., Sönmez, T. and Ünver, M.U. (2005) A kidney exchange clearinghouse in New England. *American Economic Review*, **95**(2), 376–380.
- Roth, A.E., Sönmez, T. and Ünver, M.U. (2007) Efficient kidney exchange: coincidence of wants in markets with compatibility-based preferences. *The American Economic Review*, **97**(3), 828–851.
- Saidman, S.L., Roth, A.E., Sönmez, T., Ünver, M.U. and Delmonico, F.L. (2006) Increasing the opportunity of live kidney donation by matching for two-and three-way exchanges. *Transplantation*, **81**(5), 773–782.
- Salmerón, J. and Apte, A. (2010) Stochastic optimization for natural disaster asset prepositioning. *Production and Operations Management*, **19**(5), 561–574.
- Shapiro, A. and Homem-de-Mello, T. (2000) On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *SIAM Journal of Optimization*, **11**(1), 70–86.
- Shen, S. (2013) Optimizing designs and operations of a single network or multiple interdependent infrastructures under stochastic arc disruptions. *Computers & Operations Research*, **40**(11), 2677–2688.
- Shen, S. and Chen, Z. (2013) Optimization models for differentiating quality of service levels in probabilistic network capacity design problems. *Transportation Research Part B: Methodological*, **58**(1), 71–91.
- Shen, S. and Smith, J. (2013) A decomposition approach for solving a broadcast domination network design problem. *Annals of Operations Research*, **210**(1), 333–360.

Biographies

Qipeng P. Zheng is an Assistant Professor in the Department of Industrial Engineering and Management Systems at the University of Central Florida. He received his Ph.D. in Industrial & Systems Engineering from the University of Florida in 2010 and his master's degree in Automation from Tsinghua University, China, in 2005, and his bachelor degree in Automation from North China University of Technology in 2001. His main research area is optimization and applications especially under uncertainty. He is an Associate Editor of the journal *Energy Systems*.

Siqian Shen is an Assistant Professor of Industrial and Operations Engineering at the University of Michigan. She obtained a B.S. degree from Tsinghua University in China in 2007 and a Ph.D. from the University of Florida in Industrial and Systems Engineering in 2011. Her research interests are in mathematical optimization, particularly in stochastic programming, network optimization, and integer programming. She was awarded First Place of the 2012 IIE Pritsker Doctoral Dissertation Award and was a recipient of the 2012 IBM Smarter Planet Innovation Faculty Award.

Yuhui Shi is a Ph.D. candidate in the Department of Industrial and Operations Engineering of the University of Michigan. He obtained his B.S. degree in Management Science from Fudan University in China in 2011 and joined the department of IOE in the same year. His research focuses on scheduling and various combinatorial problems.