Mean-Standard Deviation Model For Minimum Cost Flow Problem

Can Gokalp[1], Stephen D. Boyles[1], and Avinash Unnikrishnan[2]

[1]Operations Research and Industrial Engineering, The University of Texas at Austin

[2]Department of Civil and Environmental Engineering, Portland State University

April 11, 2020

ABSTRACT

We study the mean-standard deviation minimum cost flow (MSDMCF) problem, where the objective is minimizing a linear combination of the mean and standard deviation of flow costs. Due to nonlinearity and non-separability of the objective, the problem is not amenable to the standard algorithms developed for network flow problems. We prove that the solution for the MSDMCF problem coincides with the solution for a particular mean-variance minimum cost flow (MVMCF) problem. The latter problem is separable, and therefore can be solved more efficiently. Leveraging this result, we propose three methods — namely, Newton-Raphson (NR), bisection (BSC) and an hybrid of the two 'NR-BSC' — to solve the MSDMCF problem by solving multiple MVMCF subproblems. While the methods seek to find the specific parameter for the MVMCF problem for which the optimal solution is also optimal to the given MSDMCF problem, they differ in the root-finding method they use for the parameter search. We further show that this approach can be extended to solve more generalized non-separable parametric minimum cost flow problems under certain conditions. The performance of the algorithms are compared to CPLEX on benchmark MCF networks generated with the well known NETGEN generator. Computational experiments show that the proposed algorithms provide significant advantages over the off-the-shelf solver. NR method converges in about half the time it takes solver to converge. Meanwhile BSC method performs competitively when compared to the solver.

1 INTRODUCTION

The minimum cost flow (MCF) problem is to find the flow in a network that minimizes total cost while satisfying node demands and arc capacities. Many other flow and circulation problems are special cases of MCF, including the shortest path and maximum flow problems. Decision-making problems in a variety of industries — transportation, manufacturing, medicine, health care, energy, and defense, to name a few — can be formulated as MCF problems. In the traditional MCF formulation, the arc costs are assumed to be deterministic. This setting is well studied and several families of efficient algorithms have been developed for it [1].

When the arc costs are stochastic, the decision maker is often concerned with solution reliability in addition to minimizing the expected cost. Results in the travel choice literature show that travel time reliability is of comparable importance as mean travel costs [2, 3, 4, 5], motivating the incorporation of reliability-based objectives into specific network optimization problems with transportation applications. There is a rich body of literature on stochastic shortest path variants using different reliability specifications — minimizing variance or standard deviation in addition to expected travel times [6, 7, 8, 9, 10, 11, 12], maximizing probability of arrival or disutility associated with a pre-specified arrival time [13, 14, 15, 16, 17, 18, 19, 20, 21], percentiles [21, 22], risk aversion [23, 24, 25], and so forth. Reliability and risk related objectives have also been incorporated into traffic assignment models [26, 27, 28, 29, 30, 31, 32, 33].

There has been relatively less research on incorporating reliability objectives into other traditional minimum cost flow and max flow network problems. Boyles and Waller [34] study the MCF problem with uncertain arc costs where the aim is to minimize the mean as well as the variance of the total flow cost termed as the mean-variance minimum cost flow problem (MVMCF). In their model, the decision maker chooses a weight parameter indicating the relative importance of the mean and variance. They define arc marginal costs and use them to modify the generic cycle canceling algorithm. The objective is non-linear, but separable by arcs. This separability property is critical for their algorithm. In this paper, we study the MCF problem with uncertain arc costs where the objective is to minimize the mean and the standard deviation of the total flow cost. While the objective function is still convex, it is not separable by arc. Therefore, the approaches given in Boyles and Waller [34] are not applicable. In this paper, we prove that the solution to the mean-standard deviation minimum cost flow (MSDMCF) problem can be obtained by solving the MVMCF problem for an appropriate choice of weight parameter. We provide three root-finding based algorithms (bisection, Newton-Raphson, and hybrid) to determine the appropriate weight parameter. A network flow based sensitivity analysis procedure is developed to determine the derivatives for the Newton-Raphson and hybrid procedures. The MSDMCF problem is a special case of the more generalized non-separable parametric MCF (GNPMCF) problem where the objective consists of a linear additive function of flow and a weighted non-linear, non-separable function of flow. Our algorithms can be extended to the GNPMCF problem as long as the non-additive component of the objective function is an invertible, differentiable, monotonically increasing, and convex function of an additive and differentiable criterion.

Other researchers have applied robust optimization approaches to account for uncertainties in network parameters such as demands [35], costs and capacity [36], and network structure [37]. In the robust optimization paradigm, the uncertain parameters are assumed to vary in a pre-specified uncertainty set. The aim is to arrive at the best solution which is feasible for all possible realization of the uncertain parameters from their pre-specified sets. The shape of the uncertainty set indicates the decision makers risk preference and affects the tractability of the model [38]. Birge [39] and Glockner [40] apply a multi-stage stochastic programming approach to model uncertainties in network parameters in a stochastic and dynamic network flow setting. Stochastic programming approaches require knowledge of the probability distribution of the uncertain parameters. In contrast, we assume that the decision maker knows the mean and standard deviation of arc costs and is interested in minimizing the mean and standard deviation of total network flow costs. We do not focus on worst-case scenarios which can lead to overly conservative solutions.

There has been a separate body of work focusing on the impact of node and arc disruptions on the ability of a network to sustain a specific amount of flow [41, 42, 43]. Lin et al. [44] study the stochastic maximum flow problem where the nodes and arcs have uncertain discrete capacities and develop an algorithm to compute the system reliability defined as the probability that the maximum flow is greater than the given demand. Lin [45] focuses on the multi-commodity variant of [44] and defines system reliability objective as the probability of upper bound of system capacity equals a given pattern subject to budget constraints on flows. Along similar lines, Lin [46] adopts a throughput style definition of system reliability as the probability of sending a pre-specified amount of flow through the network under a cost constraint. Kuipers [47] formulate two stochastic maximum flow models: (i) maximum flow in stochastic networks (MFSN) - where the bandwidth or capacity has a log-concave probability distribution, (ii) maximum delay constrained flow problem (MDCF) where an additional stochastic delay constraint is imposed on the flows. A convex formulation and polynomial time algorithm is provided for the MFSN problem. The MDCF formulation is shown to be NP-hard and solved using an approximation algorithm. The MSDMCF model presented in this paper does not consider disruptions, failures, or uncertainties in capacity. Our model has a cost minimization perspective, whereas the above studies are concerned with maximizing flows and require full knowledge of the probability distributions.

The remainder of the paper is organized as follows. We introduce the problem formulation of the MSDMCF and show the relevance to MVMCF in Section 2. Section 3 describes the algorithm developed for solving the MSDMCF. In Section 4, we extend the results to a more general class of GNPMCF problems. We demonstrate the efficiency of our methods on randomly generated networks in Section 5, and finally, we conclude and discuss future directions in Section 6.

2 PROBLEM STATEMENT

2.1 Problem Formulation

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ represent a directed network with $\mathcal{N}$ and $\mathcal{A}$ denoting the set of nodes and arcs, respectively. The arc costs, c_{ij}, are stochastic, but independent, with known means, $E[c_{ij}]$, and variances, $Var[c_{ij}]$. Nodes and arcs are assumed to have deterministic demands

d_j and finite capacities u_{ij} respectively. Let x_{ij} denote the flow on arc (i,j) and $\mathbf{x}$ the vector of all flows. The MSDMCF problem considered in this paper has the following form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{(i,j)\in\mathcal{A}} E[c_{ij}]x_{ij} + \bar{\lambda}\sqrt{\sum_{(i,j)\in\mathcal{A}} Var[c_{ij}]x_{ij}^2} & \\ \text{s.t.} \quad & \sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j & \forall j \in \mathcal{N} \\ & 0 \leq x_{ij} \leq u_{ij} & \forall (i,j) \in \mathcal{A} \end{aligned} \tag{MSDMCF($\bar{\lambda}$)}$$

or, more compactly,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \boldsymbol{\mu}^T\mathbf{x} + \bar{\lambda}\sqrt{\mathbf{x}^T\mathbf{V}\mathbf{x}} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{MSDMCF($\bar{\lambda}$)}$$

with

$$\boldsymbol{\mu} = \begin{bmatrix} E_1 \\ \vdots \\ E_m \end{bmatrix}, \quad \mathbf{V} = diag(\mathbf{Var}) = \begin{bmatrix} Var_1 & 0 & 0 \\ & \ddots & \\ 0 & 0 & Var_m \end{bmatrix},$$

and $\mathbf{A}\mathbf{x} = \mathbf{b}$ representing the flow conversation and capacity constraints. Since $Var[c_{ij}] \geq 0$ for all $(i,j) \in \mathcal{A}$, the matrix $\mathbf{V}$ is positive semidefinite. Moreover, despite the square root, the objective is convex, as can be seen by writing

$$\sqrt{\mathbf{x}^T\mathbf{V}\mathbf{x}} = \left\|\mathbf{V}^{\frac{1}{2}}\mathbf{x}\right\|_2$$

and applying the triangle inequality.

The proposed algorithm exploits the relationship between MSDMCF and MVMCF, with the latter given by:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{(i,j)\in\mathcal{A}} E[c_{ij}]x_{ij} + \lambda \sum_{(i,j)\in\mathcal{A}} Var[c_{ij}]x_{ij}^2 & \\ \text{s.t.} \quad & \sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j & \forall j \in \mathcal{N} \\ & 0 \leq x_{ij} \leq u_{ij} & \forall (i,j) \in \mathcal{A}. \end{aligned} \tag{MVMCF(λ)}$$

or, compactly,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \boldsymbol{\mu}^T\mathbf{x} + \lambda \mathbf{x}^T \mathbf{V} \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \qquad (\text{MSDMCF}(\bar{\lambda}))$$

Unlike MSMCF, the MVMCF problem is separable by arc. Both of the problems are convex; however, the separability structure is exploitable by the solver and also other solution algorithms e.g. [34].

2.2 Proposed Approach

We adopt a parametric search method to solve the MSDMCF problem. Given $\bar{\lambda}$, we show that there exists some λ, for which optimal solutions for the mean-variance problem with λ are also optimal for the mean-standard deviation problem with $\bar{\lambda}$. In the remainder of the paper, we will refer to this parameter that produces the optimal solution to MSDMCF($\bar{\lambda}$) as λ^*. Our approach is similar in spirit to methods that have previously been applied for the mean-standard deviation shortest path problem (MSSPP) [48], [49]. However, since the MSDMCF is a continuous optimization problem, rather than a combinatorial optimization problem, existing results on MSSPP do not directly apply to the problem studied in this paper.

We first derive the relationship between the two weight parameters of these problems, which will guide the search. By applying root-finding methods to the function that defines this relationship, we find λ^* iteratively. To this end, we propose three algorithms, one based on bisection (BSC), one based on the Newton-Raphson (NR) method, and another using a combination of the two (NR-BSC). In order to obtain derivative information required for the NR algorithm, we perform sensitivity analysis on the solution of the MVMCF problem.

All of the results can be extended to a more general class of MCF problems — the generalized non-separable parametric MCF (GNPMCF), shown below:

$$\min\{\mu(\mathbf{x}) + \bar{\lambda} g(v(\mathbf{x})) : \mathbf{x} \in \mathcal{X}\}.$$

where μ and v are linear, with the composition $g \circ v(\mathbf{x}) = g(v(\mathbf{x}))$ invertible, monotonically increasing, convex; and differentiable over the interval where $x > 0$, with respect to $v(x)$.

In such cases, we can transform the function $g(\cdot)$ such that the problem becomes additive, therefore simpler. The same procedure proposed for the mean-standard deviation model can then be used to solve the GNPMCF problems with the stated assumptions above. Our main contributions are as follows:

1. We prove that the optimal solution to the MSDMCF problem is also optimal to the MVMCF problem for a particular chosen weight parameter. We also show that the converse of this claim is true, unlike the mean-standard deviation shortest path problem.

2. By deriving a key equation characterizing the relationship between the optimal solutions of the two problems, we develop three algorithms for finding the particular weight parameter λ^* to MVMCF problem for which the optimal solution is also optimal to the MSDMCF problem for a given $\bar{\lambda}$.

3. We further show that all of our results can be extended to a more general class of MCF problems.

This paper differs from the bi-objective MCF literature [50, 51, 52, 53, 54, 55, 56, 57, 58] in two aspects. The bi-objective MCF research mentioned above primarily focuses on two linear objectives whereas we have a non-separable non-linear component in our objective function. A key focus of the bi-objective MCF literature is determining the non-dominated solution set. In our paper, the two objectives can be collapsed into a single objective using a weight parameter, and we do not attempt to find the set of non-dominated solutions.

2.3 Relevance to the MVMCF

In this section, we will show that given an instance of the MSDMCF, there exists λ^* for which the optimal solution for the MVMCF(λ^*) is also optimal for the MSDMCF problem. The proof for this claim relies on the Karush-Kuhn-Tucker (KKT) necessary conditions. Therefore, we first derive these conditions for both problems below.

Let $\mathbf{h}(\mathbf{x})$ and $\boldsymbol{\ell}(\mathbf{x})$ represent the capacity and flow constraints, respectively. The feasible solution sets for the two problems are identical since their constraints are the same. Then the complementary slackness, primal feasibility, and dual feasibility conditions for both problems are given by:

$$
\begin{aligned}
& k_{ij} h_{ij}(\mathbf{x}) = 0 \quad \forall (i,j) \in \mathcal{A} \\
& h_{ij}(\mathbf{x}) \leq 0 \quad \forall (i,j) \in \mathcal{A} \\
& \ell_i(\mathbf{x}) = 0 \quad \forall i \in \mathcal{N} \\
& k_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{A} \\
& p_i \quad \text{free} \quad \forall i \in \mathcal{N}
\end{aligned} \tag{1}
$$

where k_{ij} and p_i are the dual variables for the capacity and flow balance constraints, respectively. Next, the stationary conditions are:

$$\mathbf{0} \in \nabla_{\mathbf{x}} \left(\boldsymbol{\mu}^T \mathbf{x} + \bar{\lambda} \sqrt{\mathbf{x}^T \mathbf{V} \mathbf{x}} + \sum_{i \in \mathcal{N}} p_i \ell_i(\mathbf{x}) + \sum_{(i,j) \in \mathcal{A}} k_{ij} h_{ij}(\mathbf{x}) \right), \tag{2}$$

$$\mathbf{0} \in \nabla_{\mathbf{x}} \left(\boldsymbol{\mu}^T \mathbf{x} + \lambda \mathbf{x}^T \mathbf{V} \mathbf{x} + \sum_{i \in \mathcal{N}} p_i \ell_i(\mathbf{x}) + \sum_{(i,j) \in \mathcal{A}} k_{ij} h_{ij}(\mathbf{x}) \right). \tag{3}$$

Equations (1) & (2) and (1) & (3) are the necessary conditions for optimality for MSDMCF($\bar{\lambda}$) and MVMCF(λ), respectively. Since the objective functions are also convex, these necessary conditions are also sufficient [59]. Our main result now follows.

Proposition 1. *Let $\mathbf{x}(\lambda)$ denote an optimal solution vector to the problem MVMCF(λ). This vector is also optimal to MSDMCF($\bar{\lambda}$), if λ satisfies*

$$\lambda = \frac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}. \tag{4}$$

Proof. The vector $\mathbf{x}(\lambda)$ must satisfy the KKT necessary conditions for MVMCF(λ) as it is an optimal solution. The constraint system for both problems is the same, and hence it is immediate that $\mathbf{x}(\lambda)$ will satisfy the conditions (1) for MSDMCF($\bar{\lambda}$).

As $\mathbf{x}(\lambda)$ satisfies the necessary conditions (3) for MVMCF(λ), then we must have

$$-\mu - \sum_{(i,j) \in \mathcal{A}} k_{ij} \nabla h_{ij}(\mathbf{x}) - \sum_{i \in \mathcal{N}} p_i \nabla l_i(\mathbf{x}) \in \lambda 2 \mathbf{V} \mathbf{x}(\lambda). \tag{5}$$

If equation (4) holds, then

$$\bar{\lambda} = 2\lambda\sqrt{\mathbf{x}(\lambda)^T\mathbf{V}\mathbf{x}(\lambda)}. \tag{6}$$

Substituting $\bar{\lambda}$ in (6) in the necessary conditions (2) for MSDMCF($\bar{\lambda}$), we get

$$-\mu - \sum_{(i,j)\in\mathcal{A}} k_{ij}\nabla h_{ij}(\mathbf{x}) - \sum_{i\in\mathcal{N}} p_i\nabla l_i(\mathbf{x}) \in \frac{2\lambda\sqrt{\mathbf{x}(\lambda)^T\mathbf{V}\mathbf{x}(\lambda)}\mathbf{V}\mathbf{x}(\lambda)}{\sqrt{\mathbf{x}(\lambda)^T\mathbf{V}\mathbf{x}(\lambda)}}, \tag{7}$$

Simplifying this equation yields (5), which is true by hypothesis. Therefore $\mathbf{x}(\lambda)$ satisfies the KKT necessary conditions for MSMCF($\bar{\lambda}$) with λ satisfying (4). Since the MSMCF($\bar{\lambda}$) objective function is convex, the KKT necessary conditions are also sufficient for optimality.

□

A similar argument gives the reverse direction;

Remark 1. *The converse of Proposition 1 is also true. That is, if a vector* $\mathbf{x}(\bar{\lambda})$ *is optimal for MSDMCF(*$\bar{\lambda}$*), then it is also optimal for MVMCF(*λ*) with*

$$\lambda = \frac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\bar{\lambda})^T\mathbf{V}\mathbf{x}(\bar{\lambda})}}. \tag{8}$$

Remark 1 distinguishes this setting from the mean-standard deviation shortest path problem, where the analogous statement fails [48]. Therefore, the MVMCF and MSDMCF have a closer relationship than the corresponding shortest path problems.

3 ALGORITHM

We use equation (4) to devise algorithms to solve the MSDMCF($\bar{\lambda}$) problem. If we can identify a weight parameter λ such that

$$f(\lambda) = \lambda - \frac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\lambda)^T\mathbf{V}\mathbf{x}(\lambda)}} = 0,$$

solving MVMCF(λ) will solve the original MSDMCF($\bar{\lambda}$). Therefore, the MSDMCF($\bar{\lambda}$) problem reduces to the problem of finding a root of $f(\lambda)$.

3.1 Bisection

A straightforward method to find the root is bisection. In order to use this method, we first need to show that there exists at least one root for $f(\lambda)$ in the domain $\lambda \in [0, \infty)$. To this end, we show that the function takes values of opposite signs when evaluated at the endpoints of the domain, and it is continuous for all $\lambda \in [0, \infty)$. It is trivial to see that it takes a nonpositive value as λ approaches 0, since $\bar{\lambda}$ is nonnegative and all feasible solutions are assumed to have positive variance. Moreover, the variance term in the denominator in $f(\lambda)$ is finite for any value of λ. Let Var_ℓ represent the minimum variance of any feasible flow[1]. Then the variance term is bounded below by Var_ℓ, which is positive by assumption. We can thus conclude that $f(\lambda)$ takes a positive value as λ approaches ∞ since the negative term is finite. A finite upper endpoint of the interval can simply be found by doubling an initial guess λ until $f(\lambda) \geq 0$. Further, $f(\lambda)$ is negative for $\lambda = 0$, so we can set the lower endpoint of the interval to 0.

We rely on the following variant of a standard result which allows us to show continuity of $f(\lambda)$, and therefore the correctness of the bisection method.

Lemma 1 (Maximum Value Theorem [60] - Variant)**.** *Let $X \subset \mathbb{R}^L$ and $Y \subset \mathbb{R}$, let $g : X \times Y \to \mathbb{R}$ be a continuous function, let the set $\mathcal{X}$ be a compact-valued and continuous correspondence, and further let*

$$v(y) = \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}, y).$$

Then the function $v : y \to \mathbb{R}$ is continuous.

The continuity of the function $f(\lambda)$ requires the continuity of the value function for the MVMCF(λ) problem, defined as

$$V(\lambda) = \min_{\mathbf{x} \in \mathcal{X}} f_{MV}(x, \lambda) = \sum_{(i,j) \in \mathcal{A}} E[c_{ij}] x_{ij} + \lambda \sum_{(i,j) \in \mathcal{A}} Var[c_{ij}] x_{ij}^2.$$

Note that $f_{MV}(\mathbf{x}; \lambda)$ is jointly continuous in x and λ. The feasible set $\mathcal{X}$ is compact and continuous as it is defined by a system of linear equations. Then, by Lemma 1, the value

[1] The minimum variance flow's variance cost Var_ℓ can be obtained by solving MCF where the objective only consists of the variance criterion or effectively setting λ to ∞

function $V(\lambda)$ is continuous. If the value function is continuous, then the variance term must be continuous, and therefore $f(\lambda)$ is continuous as well. Thus there exists at least one nonnegative λ^* in the domain for which $f(\lambda^*) = 0$, and the bisection method will converge to such a root.

The pseudocode for this algorithm is given in Figure 1.

FIGURE 1 Pseudocode for BSC ($\bar{\lambda}$, *TOL)*

$\lambda_{low} \leftarrow 0, \quad \lambda_{high} \leftarrow 2^k \quad \text{where} \quad k \leftarrow \min_{k' \in \mathbb{Z}^+} \left\{k' : f(2^k) \geq 0\right\}$

$Found \leftarrow False$

while not $Found$ **do**

$$\lambda \leftarrow \frac{(\lambda_{high} + \lambda_{low})}{2}$$

$$\mathbf{x}(\lambda) \leftarrow \arg\min\left(\text{MVMCF}(\lambda)\right)$$

$$f(\lambda) = \lambda - \frac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}$$

if $|f(\lambda)| \leq \text{TOL}$ **then**

$Found \leftarrow True$

else

if $f(\lambda) > 0$ **then**

$\lambda_{high} \leftarrow \lambda$

else

$\lambda_{low} \leftarrow \lambda$

3.2 Newton's Algorithm

Although the bisection method is guaranteed to converge, it only has a linear convergence rate and may need many iterations to converge, each of which requires solving a mean-variance problem. An alternative is to seek a root for $f(\lambda)$ with the Newton-Raphson method. This method is simple to implement, and under certain conditions has quadratic convergence [59]. However, this method requires calculating the derivative of $f(\lambda)$, which involves solving an auxiliary optimization problem. The Newton update for $f(\lambda)$ is given by:

$$\lambda_{n+1} = \left[\lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)}\right]^+ \tag{9}$$

Let $\boldsymbol{\xi}$ represent the vector of derivatives of the optimal solution $\mathbf{x}$ with respect to λ $\boldsymbol{\xi} = d\mathbf{x}/d\lambda$. We can then write $f'(\lambda)$ as

$$f'(\lambda) = 1 + \frac{\bar{\lambda}\mathbf{x}(\lambda)^T \mathbf{V}\boldsymbol{\xi}}{2\sqrt[3]{\mathbf{x}(\lambda)^T \mathbf{V}\mathbf{x}(\lambda)}} \tag{10}$$

In this section, we show that sensitivity analysis can be conducted for the MVMCF(λ) problem to identify the derivatives $\boldsymbol{\xi}$, as in other applications [61, 62]. The derivative of the optimal solution vector with respect to the weight parameter λ can be interpreted as the sensitivity of the solution to changes in λ.

Introducing KKT multipliers $\mathbf{p}$ and $\boldsymbol{\eta}$ for the flow conservation and capacity constraints in the MVMCF(λ) problem, respectively, we get the necessary conditions for optimality;

$$E[c_{ij}] + \lambda 2Var[c_{ij}]x_{ij} + \eta_{ij} + p_i - p_j = 0 \quad \forall (i,j) \in \mathcal{A} \tag{11a}$$

$$\sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j \quad \forall j \in \mathcal{N} \tag{11b}$$

$$0 \le x_{ij} \le u_{ij} \quad \forall (i,j) \in \mathcal{A} \tag{11c}$$

$$\eta_{ij}(x_{ij} - u_{ij}) = 0 \quad \forall (i,j) \in \mathcal{A} \tag{11d}$$

$$\eta_{ij} \ge 0 \quad \forall (i,j) \in \mathcal{A}. \tag{11e}$$

These conditions are also sufficient as this is a convex minimization problem and therefore represent the conditions for optimality.

Let $\mathcal{J}$ denote the set of arcs for which the reduced costs at optimality $\bar{c}_{ij} = c_{ij} + p_i - p_j$ are positive, let $\mathcal{K}$ be the set of arcs for which the reduced costs are negative, and let $\mathcal{L}$ denote the set of arcs where the reduced costs are zero:

$$\mathcal{J} = \{(i,j) : \bar{c}_{ij} > 0\} \tag{12a}$$

$$\mathcal{K} = \{(i,j) : \bar{c}_{ij} < 0\} \tag{12b}$$

$$\mathcal{L} = \{(i,j) : \bar{c}_{ij} = 0\}\,. \tag{12c}$$

By complementary slackness, at optimality we know that all links in $\mathcal{J}$ have zero flow, and all links in $\mathcal{K}$ have flows equal to their capacities. For small perturbations in λ, all reduced costs for links in $\mathcal{J}$ will remain positive, and all reduced costs for links in $\mathcal{K}$ will remain negative. Therefore, such links will see no change in flow, and for the purpose of sensitivity analysis, it is sufficient to consider only the links $(i,j) \in \mathcal{L}$. For these arcs, complementary slackness constraints (11d) force η_{ij} to be 0, reducing the optimality conditions to

$$E[c_{ij}] + 2\lambda x_{ij} Var[c_{ij}] + p_i - p_j = 0 \quad \forall (i,j) \in \mathcal{A} \tag{13a}$$

$$\sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j \quad \forall j \in \mathcal{N} \tag{13b}$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in \mathcal{A}. \tag{13c}$$

Let $\boldsymbol{\varphi}$ represent the marginal change in $\mathbf{p}$, when the weight parameter λ is perturbed, $\boldsymbol{\varphi} = d\mathbf{p}/d\lambda$. We can differentiate the equality conditions (13a–13c) to get a new set of equations (14a–14c) that must hold as λ is varied.

$$2Var[c_{ij}](x_{ij} + \lambda\xi_{ij}) + \varphi_i - \varphi_j = 0 \quad \forall (i,j) \in \mathcal{L} \tag{14a}$$

$$\sum_{(j,k)\in\mathcal{L}} \xi_{jk} - \sum_{(i,j)\in\mathcal{L}} \xi_{ij} = 0 \quad \forall j \in \mathcal{N} \tag{14b}$$

$$\xi_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{L} : x_{ij} = 0 \tag{14c}$$

$$\xi_{ij} \leq 0 \quad \forall (i,j) \in \mathcal{L} : x_{ij} = u_{ij} \tag{14d}$$

The constraints (14c) and (14d) address the possibility of degenerate solutions, with zero-reduced cost links at an upper or lower flow bound. Otherwise, the variables $\boldsymbol{\xi}$ are free and may take or positive values, since the flow on non-degenerate links in $\mathcal{L}$ can either increase or decrease when λ is perturbed.

A solution to this problem can be obtained by solving the set of linear equations above. However, there is a more efficient approach. Equations (14) are the KKT conditions of the network flow problem:

$$\begin{aligned} \min_{\boldsymbol{\xi}} \quad & 2\sum_{(i,j)\in\mathcal{L}} Var[c_{ij}]x_{ij}\xi_{ij} + \lambda \sum_{(i,j)\in\mathcal{L}} Var[c_{ij}]\xi_{ij}^2 \\ \text{s.t.} \quad & \sum_{(j,k)\in L} \xi_{jk} - \sum_{(i,j)\in\mathcal{L}} \xi_{ij} = 0 \qquad \forall j \in \mathcal{N} \\ & \xi_{ij} \geq 0 \qquad\qquad\qquad\qquad \forall (i,j) \in \mathcal{L} : x_{ij} = 0 \\ & \xi_{ij} \leq 0 \qquad\qquad\qquad\qquad \forall (i,j) \in \mathcal{L} : x_{ij} = u_{ij} \end{aligned} \tag{$\Delta(\lambda, \mathbf{x})$}$$

The vector $\mathbf{x}$ is the solution of the (MVMCF(λ)) problem at λ, that is, the only decision variable in this problem is the $\boldsymbol{\xi}$ vector. This is essentially another MVMCF problem where $E[c_{ij}] = 2Var[c_{ij}]x_{ij}$ where solution variables ξ_{ij} represent the rate of change in the original link flows as λ varies, and it is possible for these changes to be negative. This auxiliary problem has fewer variables than the MVMCF problem, as one only needs to consider the links in set $\mathcal{L}$.

The pseudocode in Figure 2 outlines the Newton-Raphson based search procedure which uses the flow sensitivity procedure to determine the derivatives. Since evaluating $f(\lambda)$ requires solving an optimization problem, it is unclear what conditions guarantee convergence of the algorithm. In the next subsection, we provide a fail-safe to alleviate the lack of convergence guarantee for the pure Newton algorithm. We note that the method converged for all the test instances in our experiments, despite the lack of convergence proof.

3.3 Hybrid Algorithm

The third algorithm (NR-BSC) is a hybrid of the first two, primarily using a Newton step size with bisection as a fallback to ensure convergence, as in Press et al. [63]. Specifically, we switch to a bisection step whenever the current Newton-Raphson step suggests a solution

FIGURE 2 Pseudocode for NR ($\bar{\lambda}$, *TOL)*

$\lambda_{low} \leftarrow 0, \quad \lambda_{high} \leftarrow 2^k \quad \text{where} \quad k \leftarrow \min_{k' \in \mathbb{Z}^+} \left\{ k' : f(2^k) \geq 0 \right\}$

$\lambda \leftarrow \dfrac{\lambda_{low} + \lambda_{high}}{2}, \quad Found \leftarrow False$

while not $Found$ **do**

$\quad \mathbf{x}(\lambda) \leftarrow \arg\min \left(\text{MVMCF}(\lambda) \right)$

$\quad f(\lambda) = \lambda - \dfrac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}$

$\quad$ **if** $|f(\lambda)| \leq \text{TOL}$ **then**

$\quad\quad Found \leftarrow True$

$\quad$ **else**

$\quad\quad \boldsymbol{\xi} \leftarrow \arg\min \Delta(\lambda, \mathbf{x})$

$\quad\quad f'(\lambda) = 1 + \dfrac{\bar{\lambda} \mathbf{x}(\lambda)^T \mathbf{V} \boldsymbol{\xi}}{2\sqrt[3]{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}$

$\quad\quad \lambda = \left[\lambda - \dfrac{f(\lambda)}{f'(\lambda)} \right]^+$

that is out of the bracket, or whenever the bracket size is not reducing rapidly enough. The resulting procedure has a best-case quadratic convergence rate and worst-case linear convergence rate.

It is easy to check for the first condition to see if the step would take the solution out of bounds. However, to check the second condition, a definition for 'rapidly enough' is needed. Press et al. [63] define such a condition: if $|2f(\lambda)| > |\Delta_\lambda f'(\lambda)|$ then the bracket size is not reduced rapidly enough, where $\Delta_\lambda = |\lambda_{n+1} - \lambda_n|$. In our implementation, we use a simpler condition, and check if $|f(\lambda)|$ is smaller than in the previous iteration. This approach prevents possible divergent behaviors in the pure NR algorithm. The pseudocode of the algorithm is provided in Figure 3.

FIGURE 3 Pseudocode for NR-BSC ($\bar{\lambda}$, *TOL)*

$\lambda_{low} \leftarrow 0, \quad \lambda_{high} \leftarrow 2^k \quad$ where $\quad k \leftarrow \min_{k' \in \mathbb{Z}^+} \left\{k' : f(2^k) \geq 0\right\}$

$\lambda \leftarrow \dfrac{\lambda_{low} + \lambda_{high}}{2}, \quad f|(\lambda_{prev})| \leftarrow \infty, \quad Found \leftarrow False$

while not $Found$ **do**

    $\mathbf{x}(\lambda) \leftarrow \arg\min\left(\mathrm{MVMCF}(\lambda)\right)$

    $f(\lambda) = \lambda - \dfrac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}$

    if $|f(\lambda)| \leq \mathrm{TOL}$ **then**

        $Found \leftarrow True$

    else

        $\lambda_{prev} \leftarrow \lambda$

        if $|f(\lambda)| \leq |f(\lambda_{prev})|$ **then**

            $\boldsymbol{\xi} \leftarrow \arg\min \Delta(\lambda, \mathbf{x})$

            $f'(\lambda) = 1 + \dfrac{\bar{\lambda}\mathbf{x}(\lambda)^T \mathbf{V} \boldsymbol{\xi}}{2\sqrt[3]{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}$

            $f(\lambda_{prev}) \leftarrow f(\lambda)$

            $\lambda \leftarrow \lambda - \dfrac{f(\lambda)}{f'(\lambda)}$

            if $\lambda_{low} < \lambda < \lambda_{high}$ **then**

                if $f(\lambda) > 0$ **then**

                    $\lambda_{high} \leftarrow \lambda$

                else

                    $\lambda_{low} \leftarrow \lambda$

            else

                Update the bounds using λ_{prev} **and perform Bisection step**

        else

            Update the bounds using λ_{prev} **and perform Bisection step**

4 GENERALIZATION TO NON-SEPARABLE PARAMETRIC MINIMUM COST FLOW

In this section, we will consider the generalized non-separable parametric MCF (GNPMCF) given by:

$$
\begin{aligned}
\min_{\mathbf{x}} \sum_{(i,j)\in\mathcal{A}} \mu[c_{ij}x_{ij}] + \bar{\lambda} g\left(\sum_{(i,j)\in\mathcal{A}} v[t_{ij}x_{ij}]\right) & \\
\text{s.t.} \sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j \qquad & \forall j \in \mathcal{N} \\
0 \le x_{ij} \le u_{ij} \qquad & \forall (i,j) \in \mathcal{A}
\end{aligned} \tag{15}
$$

where μ is a function that is linear in $\mathbf{x}$ and additive, v is a differentiable function, and the composition $g \circ v = g(v(\mathbf{x}))$ is a monotonically increasing, convex, and differentiable function over the region $x > 0$. (Examples of such functions are discussed below). Such a function is necessarily invertible, and applying g^{-1} as a transformation will yield an additive problem as a result:

$$
\begin{aligned}
\min_{\mathbf{x}} \sum_{(i,j)\in\mathcal{A}} \mu[c_{ij}x_{ij}] + \lambda g^{-1}\left(g\left(\sum_{(i,j)\in\mathcal{A}} v[c_{ij}x_{ij}]\right)\right) & \\
\text{s.t.} \sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j \qquad & \forall j \in \mathcal{N} \\
0 \le x_{ij} \le u_{ij} \qquad & \forall (i,j) \in \mathcal{A}
\end{aligned} \tag{16}
$$

By following the same procedure as in the proof of Proposition 1, we can find a relation between the two problems. Rewrite the stationary conditions for (15) and (16) as

$$
-\mu[\mathbf{c}] - \sum_{(i,j)\in\mathcal{A}} k_{ij} \nabla_{\mathbf{x}} h_{ij}(\mathbf{x}) - \sum_{i\in\mathcal{N}} p_i \nabla_{\mathbf{x}} l_i(\mathbf{x}) \in \bar{\lambda} \sum_{(i,j)\in\mathcal{A}} v'[c_{ij}x_{ij}] g'\Big(\sum_{(i,j)\in\mathcal{A}} v[c_{ij}x_{ij}]\Big), \tag{17}
$$

$$
-\mu[\mathbf{c}] - \sum_{(i,j)\in\mathcal{A}} k_{ij} \nabla_{\mathbf{x}} h_{ij}(\mathbf{x}) - \sum_{i\in\mathcal{N}} p_i \nabla_{\mathbf{x}} l_i(\mathbf{x}) \in \lambda \sum_{(i,j)\in\mathcal{A}} v'[c_{ij}x_{ij}], \tag{18}
$$

and observe that the left-hand sides of both equations are the same. Therefore, for a given $\bar{\lambda}$ we seek a λ such that;

$$\lambda = \bar{\lambda} g' \left(\sum_{(i,j)\in\mathcal{A}} v([c_{ij} x_{ij}]) \right). \tag{19}$$

In the above equations, the vector $\mathbf{x}$ is the solution vector obtained by solving (16) at λ — in order to not the clutter the notation, instead of $\mathbf{x}(\lambda)$ we referred to it as $\mathbf{x}$. Note that differentiability of both $g \circ v$ and v are necessary for Proposition 1 to follow. Moreover, as $g \circ v$ is convex by assumption, the KKT necessary conditions are also sufficient for optimality of (15) and therefore the optimal solution for (16) with λ satisfying (19) is also optimal for (15). To solve (15), it is sufficient to find a root of the function

$$f(\lambda) = \lambda - \bar{\lambda} g' \left(\sum_{(i,j)\in\mathcal{A}} v([c_{ij} x_{ij}]) \right). \tag{20}$$

Next, we show that a root exists for this function. By similar arguments to those in the proof of Lemma 1, the function $f(\lambda)$ is continuous. Moreover, it takes values of opposite signs when evaluated at the endpoints of the domain. By assumption, $g \circ v$ is monotonically increasing, and therefore the derivative is positive for any λ in the domain. Then, $f(\lambda)$ takes a nonpositive value as λ approaches 0. The criterion v term in $f(\lambda)$ is finite for any value of λ, so $f(\lambda)$ takes a positive value as λ approaches ∞, as the negative term will be finite with this assumption. To use the Newton-Raphson or the hybrid method we need the function g to be twice differentiable, otherwise one can always use BSC to solve the problem.

In practice, optimization problems of this form might arise when capturing the utilities with an exponential function. Other functions such as quadratic, Ackley, Brent, and Brown fall in to the class of functions for g that satisfies the required conditions to use in this framework. We refer the reader to an extensive survey of benchmark functions [64] for more applicable functions that fall into this class. It is also possible to arrive at this form starting from other optimization problems. For instance, the single criterion MCF problem — such as minimizing only the expected value — with nonlinearly evaluated budget constraints

$$
\begin{aligned}
\min_{\mathbf{x}} & \sum_{(i,j)\in\mathcal{A}} \mu[c_{ij}x_{ij}] \\
\text{s.t.} & \sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j \quad \forall j \in \mathcal{N} \\
& 0 \le x_{ij} \le u_{ij} \qquad\qquad \forall (i,j) \in \mathcal{A} \\
& g\left(\textstyle\sum_{(i,j)\in\mathcal{A}} v[c_{ij}x_{ij}]\right) \le B,
\end{aligned}
\tag{21}
$$

can be cast into the form

$$
\begin{aligned}
\min_{\mathbf{x}} \max_{\bar\lambda\ge 0} & \sum_{(i,j)\in\mathcal{A}} \mu[c_{ij}x_{ij}] + \bar\lambda g\left(\sum_{(i,j)\in\mathcal{A}} v[c_{ij}x_{ij}]\right) - \bar\lambda B \\
\text{s.t.} & \sum_{(j,k)\in A} x_{jk} - \sum_{(i,j)\in\mathcal{A}} x_{ij} = d_j \qquad \forall j \in \mathcal{N} \\
& 0 \le x_{ij} \le u_{ij} \qquad\qquad \forall (i,j) \in \mathcal{A},
\end{aligned}
\tag{22}
$$

by Langrangianizing the budget constraints. We then have an outer minimization problem for a fixed value of $\bar\lambda$ which is of the form we consider in this section.

5 NUMERICAL EXPERIMENTS

In this section, we assess the performance of the proposed algorithms and compare them to the performance of the CPLEX solver. We compare the methods using the same benchmark suite, and thus provide intuition into their performance on networks with different characteristics, including, how dense the network is, how restricting are the capacities on the arcs. All of the computational experiments are performed on a quad-core 2.8 GHz computer with 16 GB RAM.

5.1 Benchmark Networks

The performance of the methods are evaluated on the networks generated with the well-known random generator NETGEN [65]. We use the benchmark suite created in [66], which was designed to compare linear MCF solution methods.

In the NETGEN problem families, the arc costs and capacities are uniformly drawn from [1, 10,000] and [1, 1,000], respectively. There are approximately $\sqrt{n}$ supply and demand nodes, and the average supply per supply node is set to 1000.

There are four problem families created with above characteristics:

- **NETGEN-8.** Sparse networks, with average node outdegree of 8 ($m = 8n$).
- **NETGEN-SR.** Dense networks, with average node outdegree of $\sqrt{n}$ ($m \approx n\sqrt{n}$).
- **NETGEN-LO-8.** Same as NETGEN-8, except the average supply per supply node is 10.
- **NETGEN-LO-SR.** Same as NETGEN-SR, except the average supply per supply node is 10.

Arc capacities in NETGEN-LO-8 and NETGEN-LO-SR impose only loose bounds for feasible flows, as the average supply per supply node is small.

We use the arc costs created in the instances as the mean arc cost. We sample a coefficient of variation COV_{ij} for each link, drawn uniformly from $[0.15, 0.3]$, and thus set the standard deviation as $\sigma_{ij} = COV_{ij}\mu_{ij}$. This interval for COV_{ij} represents typical variation in transportation networks [67].

5.2 Finding an initial λ

It is possible to reduce the search space for the root-finding algorithms by finding a smaller initial interval which includes the root. Let Var_h represent an upper bound on the variance of optimal solutions with any λ to the MVMCF(λ) problem. An efficient way to obtain such a bound is to solve a linear minimum cost flow problem with the mean costs, essentially setting λ to 0, and setting Var_h to be the variance of such a solution. This is an upper bound on the variance of the optimal solution. Therefore, any λ with

$$\lambda \leq \frac{\bar{\lambda}}{2\sqrt{Var_h}},$$

also satisfies

$$\lambda \leq \frac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}},$$

where $\mathbf{x}$ is obtained from solving the MVMCF(λ) at that λ. Hence, we can set the lower bound for the interval that includes the root to $\lambda_{low} = \bar{\lambda}/2\sqrt{Var_h}$.

It is also possible to find an upper bound on the interval in a similar fashion. Doing so would require solving a quadratic MCF. There is an alternative, simpler procedure which provides a looser upper bound: if we set λ to $\bar{\lambda}/2$, and if $\sqrt{\mathbf{x}(\lambda)^T\mathbf{V}\mathbf{x}(\lambda)} > 1$ then $f(\lambda) > 0$. By changing units one can always satisfy the condition $\sqrt{\mathbf{x}(\lambda)^T\mathbf{V}\mathbf{x}(\lambda)} > 1$, and re-solve the problem after scaling. Therefore we can set $\lambda_{high} = \bar{\lambda}/2$.

However, our computational experiments show that the former approach based on solving a quadratic MCF performs better. Specifically, let Var_ℓ represent a lower bound on the variance of optimal solutions with any λ to the MVMCF(λ) problem. In order to obtain such a bound, one needs to solve a quadratic minimum cost flow problem with the variance term alone. Var_ℓ will lead to the maximum possible value for the negative term in $f(\lambda)$. Then, any λ with

$$\lambda \geq \frac{\bar{\lambda}}{2\sqrt{Var_\ell}},$$

also satisfies

$$\lambda \geq \frac{\bar{\lambda}}{2\sqrt{\mathbf{x}(\lambda)^T\mathbf{V}\mathbf{x}(\lambda)}},$$

where the $\mathbf{x}$ is obtained from solving the MVMCF(λ) at that λ. Hence, we can set the upper bound for the interval that includes the root to $\lambda_{high} = \bar{\lambda}/2\sqrt{Var_\ell}$

In the experiments, the Newton-Raphson method was given an initial λ of λ_{low}, whereas bisection method was given both λ_{high} and λ_{low} as described above.

5.3 Comparison of algorithms

The reported running times for the algorithms NR and BSC include the time elapsed for finding the interval for λ. We do not report the hybrid algorithm separately in the tables and figures below as its performance is almost identical to the NR method since the "failsafe" bisection steps were rarely used. Both of the line search methods used convergence criterion of $TOL = 10^{-8}$. The MVMCF subproblems are solved using CPLEX solver. All comparisons

	Initialization		Time (s)		Iteration #	
	λ_{low}	λ_{high}	BSC	NR	NR-iter	BSC-iter
Naive	0	$\bar{\lambda}/2$	154.85	19.15	2	21
Custom	$\bar{\lambda}/2\sqrt{Var_h}$	$\bar{\lambda}/2\sqrt{Var_\ell}$	26.77	11.63	1	1

TABLE 1: Initialization procedure benefits.

were done using $\bar{\lambda} = 10$. We also address how the performance changes for different values of $\bar{\lambda}$ later in this section.

For each graph family, each method's performance was measured by seconds needed to achieve 10^{-2}% "optimality gap" – the percentage gap between the method's objective and the best objective found by all three algorithms. The reported running times are averaged over 5 instances for each problem.

Table 1 illustrates the benefits of using custom bounds found with the procedure described in Subsection 5.2. It compares iteration numbers and the running time of the algorithms for both naive and custom bounds, on a dense network with 4096 nodes and degree 64. The custom initialization help the algorithms to start very close to $\lambda^\star$, and therefore iteration numbers and running times are much lower.

Tables 2–5 provide the absolute running times in seconds, and the best running times are bolded. Figures 4–7 provide corresponding plots, using logarithmic scales so the relative difference between methods is clearly apparent across all problem sizes tested.

In the tables, the size of the network is indicated by the number of nodes and the average number of degree per node in each row. NR method outperforms the other methods on every experiment. While the BSC method outperforms CPLEX on dense networks for smaller problem sizes, it has a worse trend than CPLEX in all cases. All of the methods' solution times increase about an order of magnitude when the number of nodes is held fixed and the density of the network increased.

Additionally, Tables 2–5 also provide the average number of iterations for the proposed algorithms to achieve the gap level. The NR method requires fewer iterations for all fami-

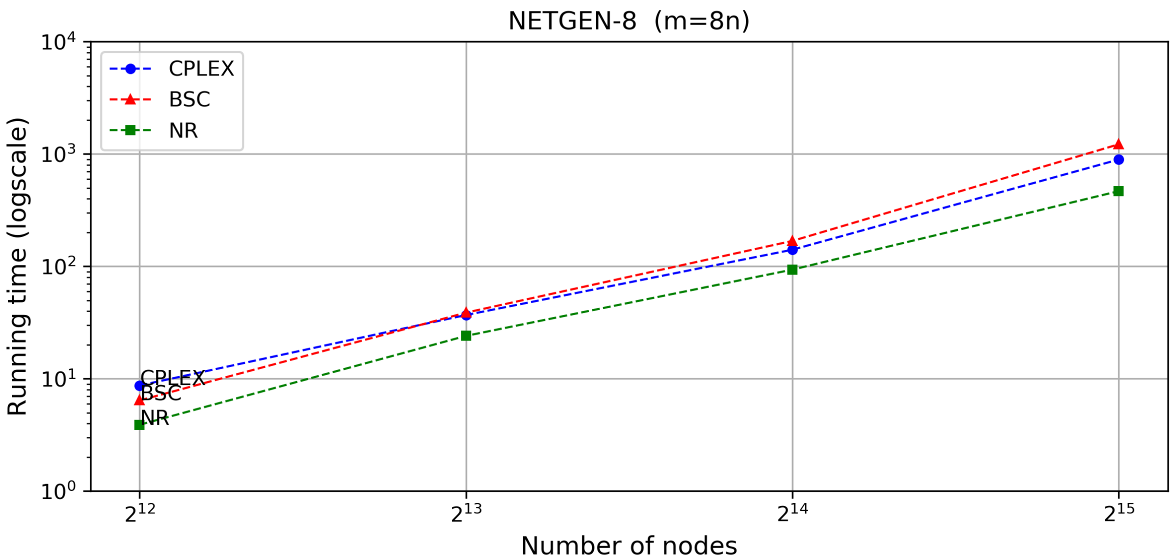

FIGURE 4: Comparison of the algorithms on NETGEN-8 families (logarithmic scale).

Size		Time (s)			Avg. Iteration #	
n	deg	CPLEX	BSC	NR	NR-iter	BSC-iter
2^{12}	8	8.70	6.40	**3.89**	2.0	2.2
2^{13}	8	36.90	38.74	**24.06**	2.0	2.0
2^{14}	8	140.64	168.51	**93.55**	1.2	1.2
2^{15}	8	893.78	1220.60	**467.02**	1.0	1.0

TABLE 2: Comparison on NETGEN-8 instances.

lies except NETGEN-LO-SR. The solution time of the NR method is better than the BSC method, despite requiring more iterations for this family. This is due to each method requiring a different amount of time to find the initial λ. The time for each method to achieve its first objective includes only the time elapsed for finding initial λ. For the NR method, initial λ is set to λ_{low}, and to find this lower bound, a linear MCF problem needs to be solved. On the other hand, for the BSC method, initial λ is set to $(\lambda_{low} + \lambda_{high})/2$ which requires finding both the lower bound and the upper bound. The latter requires solving a quadratic MCF problem and thus is more costly.

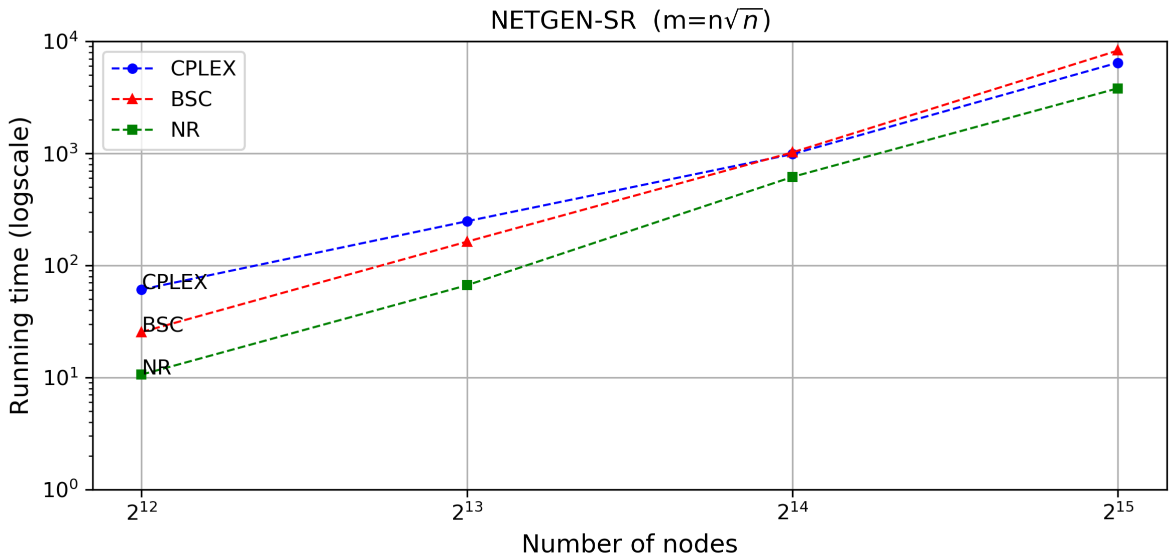

FIGURE 5: Comparison of the algorithms on NETGEN-SR families (logarithmic scale).

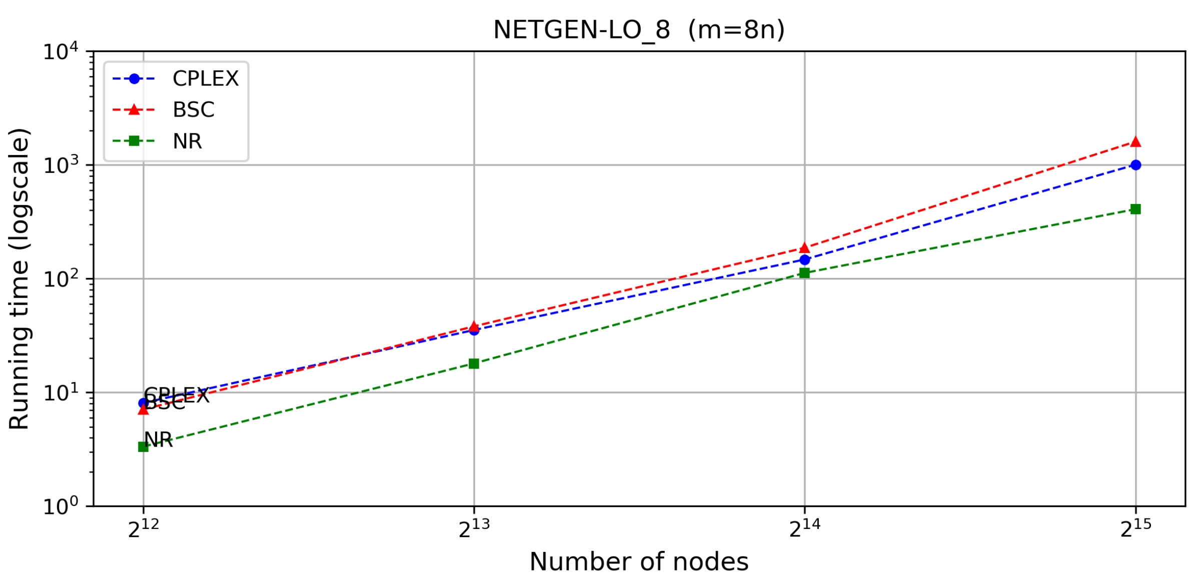

FIGURE 6: Comparison of the algorithms on NETGEN-LO-8 families (logarithmic scale).

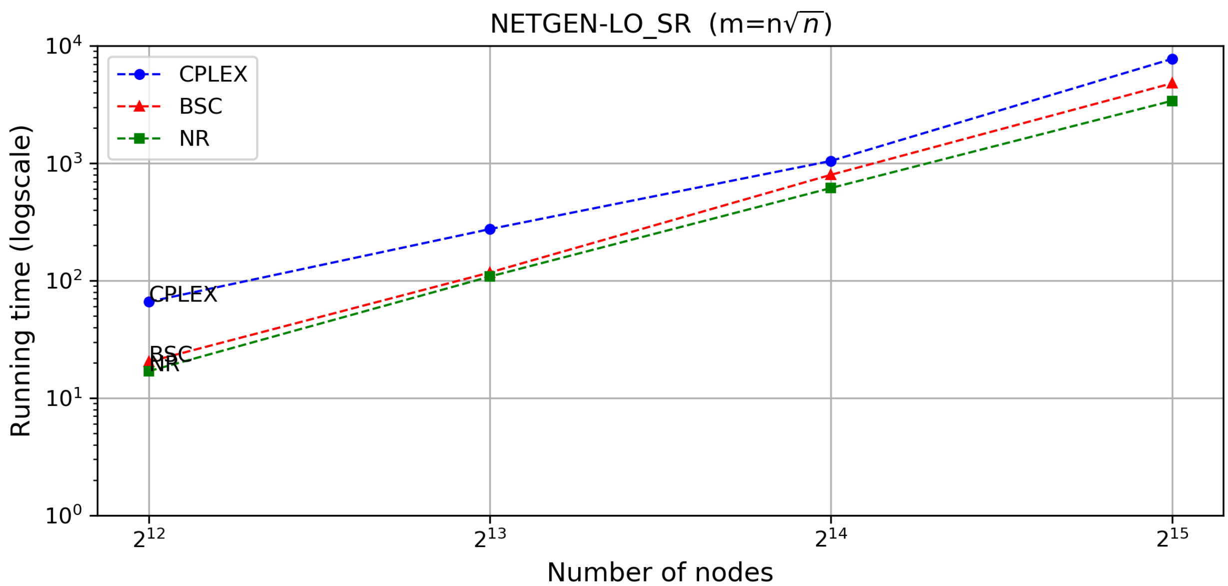

FIGURE 7: Comparison of the algorithms on NETGEN-LO-SR families (logarithmic scale).

Size		Time (s)			Avg. Iteration #	
n	deg	CPLEX	BSC	NR	NR-iter	BSC-iter
2^{12}	64	60.96	25.40	**10.60**	1.0	1.2
2^{13}	90	247.85	162.85	**66.53**	1.0	1.2
2^{14}	128	984.32	1019.80	**616.12**	1.2	1.2
2^{15}	181	6441.52	8265.37	**3804.74**	1.2	1.0

TABLE 3: Comparison on NETGEN-SR instances.

Size		Time (s)			Avg. Iteration #	
n	deg	CPLEX	BSC	NR	NR-iter	BSC-iter
2^{12}	8	8.08	7.08	**3.34**	2.0	2.0
2^{13}	8	35.38	38.11	**17.99**	1.8	2.0
2^{14}	8	147.63	187.12	**112.51**	1.6	2.0
2^{15}	8	1005.59	1604.47	**408.43**	1.0	2.0

TABLE 4: Comparison on NETGEN-LO-8 instances.

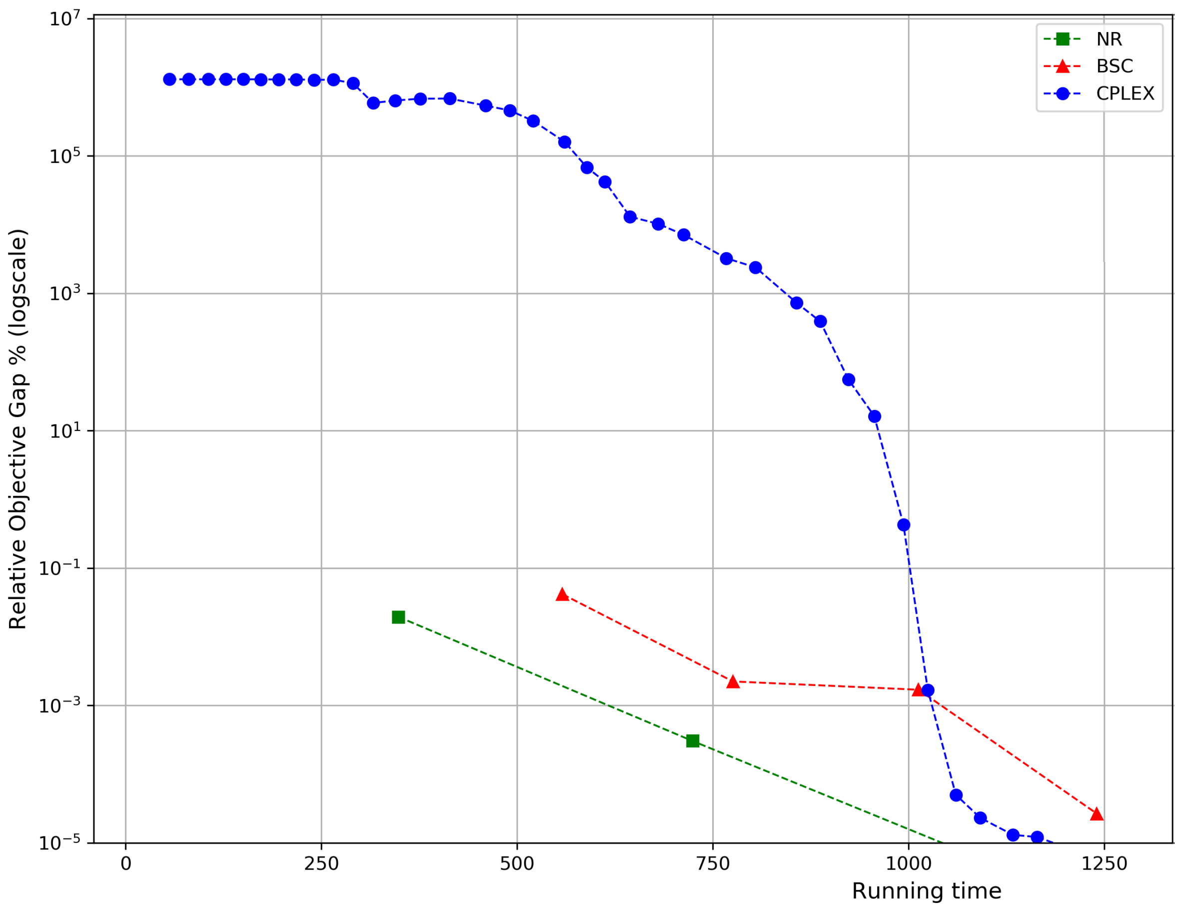

FIGURE 8: Convergence behavior.

Size		Time (s)			Avg. Iteration #	
n	deg	CPLEX	BSC	NR	NR-iter	BSC-iter
2^{12}	64	65.94	20.34	**16.91**	2.0	1.2
2^{13}	90	274.49	117.14	**108.20**	2.0	1.2
2^{14}	128	1044.87	796.01	**614.13**	1.8	2.0
2^{15}	181	7727.51	4789.31	**3397.46**	1.4	1.4

TABLE 5: Comparison on NETGEN-LO-SR instances.

		NETGEN-8				NETGEN-SR				NETGEN-LO-8				NETGEN-LO-SR			
Gap	Method	2^{12}	2^{13}	2^{14}	2^{15}	2^{12}	2^{13}	2^{14}	2^{15}	2^{12}	2^{13}	2^{14}	2^{15}	2^{12}	2^{13}	2^{14}	2^{15}
10^{-1}	CPLEX	8.70	36.90	138.02	872.25	59.47	242.40	961.95	6266.61	7.97	34.54	144.63	981.43	63.01	265.72	1031.35	7528.59
	BSC	6.40	38.74	168.51	1220.60	24.12	153.43	953.44	8265.37	5.92	28.62	139.49	1218.74	19.40	110.43	552.81	3920.18
	NR	3.89	24.06	93.55	467.02	10.60	66.53	513.04	3184.48	1.52	8.94	65.73	408.43	10.42	56.61	340.59	2922.85
10^{-2}	CPLEX	8.70	36.90	140.64	893.78	60.96	247.85	984.32	6441.52	8.08	35.38	147.63	1005.59	65.94	274.49	1044.87	7727.51
	BSC	6.40	38.74	168.51	1220.60	25.40	162.85	1019.80	3804.74	7.08	38.11	187.12	1604.47	20.34	117.14	796.015	4789.31
	NR	3.89	24.06	93.55	467.02	10.60	66.53	616.12	8265.37	3.34	17.99	112.51	408.43	16.91	108.20	614.13	3397.46
10^{-3}	CPLEX	8.70	36.90	143.45	920.82	63.49	261.11	1022.64	6746.57	27.74	109.89	464.80	1043.12	68.58	289.56	1077.93	8015.43
	BSC	6.40	38.74	168.51	1220.60	29.65	162.85	1155.14	8854.55	7.08	39.77	226.26	1988.61	28.07	124.54	936.20	5672.07
	NR	3.89	24.06	93.55	467.02	20.99	121.62	1002.74	5845.18	3.34	20.27	143.81	882.13	18.46	118.63	687.83	5329.62

TABLE 6: Time elapsed to achieve gap levels.

Figure 8 presents the convergence behavior of the algorithms on the NETGEN-LO-SR family on a representative problem instance with 2^{14} nodes. The BS and NR methods we propose start very close to the optimal solution, thanks to the tight interval found for the parameter using the procedure described in Subsection 5.2. Both of the methods achieve a percentage gap of 0.1% in their first iteration. Similar behavior is observed in other graph families and instances. The time needed to achieve various gap levels is shown in Table 6. For dense networks, for the early iterations CPLEX has a much higher gap value than the methods we propose. Moreover, the performance from NR and BSC methods can be further optimized by tuning the precision to which the subproblems are solved, since high-precision subproblem solutions are likely more useful in later iterations than in earlier ones (in these experiments, no such optimization was done).

5.4 Sensitivity to reliability

In this subsection, we emphasize the need for a reliable model by showing difference in solutions between a reliable model versus deterministic model where arc costs are stochastic. Additionally, we also investigate how the performance of the algorithms changes with respect to the changes in the reliability parameter. Note that the change in the nonlinear part of the objective depends on both $\bar{\lambda}$ and the standard deviation parameter of the arcs. In our experiments, standard deviation was generated uniformly from $[0.15\mu_{ij}, 0.3\mu_{ij}]$ as that represents typical variation in transportation networks [67], however in other types of networks this problem parameter might be very different. To capture the possible affects of higher or lower parameter values for the standard deviation, we allowed the coefficient of variation to range from 0.1 to 1000.

In terms of modeling, Figure 9 plots the percentage relative gap between the objective value of a deterministic model, that only considers the mean cost, and the objective value of the mean-standard deviation model versus the reliability parameter $\bar{\lambda}$ on a small network with 1024 nodes and 8192 arcs. As the reliability becomes more and more important to the decision maker, performance of the deterministic model deteriorates. In such situations, where reliability is important, using a mean-standard deviation model may outweigh the additional computation costs over optimizing expected performance only. Moreover, Figure 10 demonstrates that a significant decrease in the standard deviation cost can be traded off with a relatively small increase in the mean cost, especially when $\bar{\lambda}$ is small. It is thus possible to substantially improve reliability with a small impact to mean cost.

Figures 11–14 plots performance of the algorithms with respect to different reliability parameter for each of the graph family in the benchmark suite. Amongst all the methods, performance of the BSC method is the most robust against the variation in the λ parameter. On the other hand, the performance of the NR method and CPLEX seems to be affected when the λ varied, while this is observable in all families, the change in runtime especially notable on NETGEN-LO-SR family.

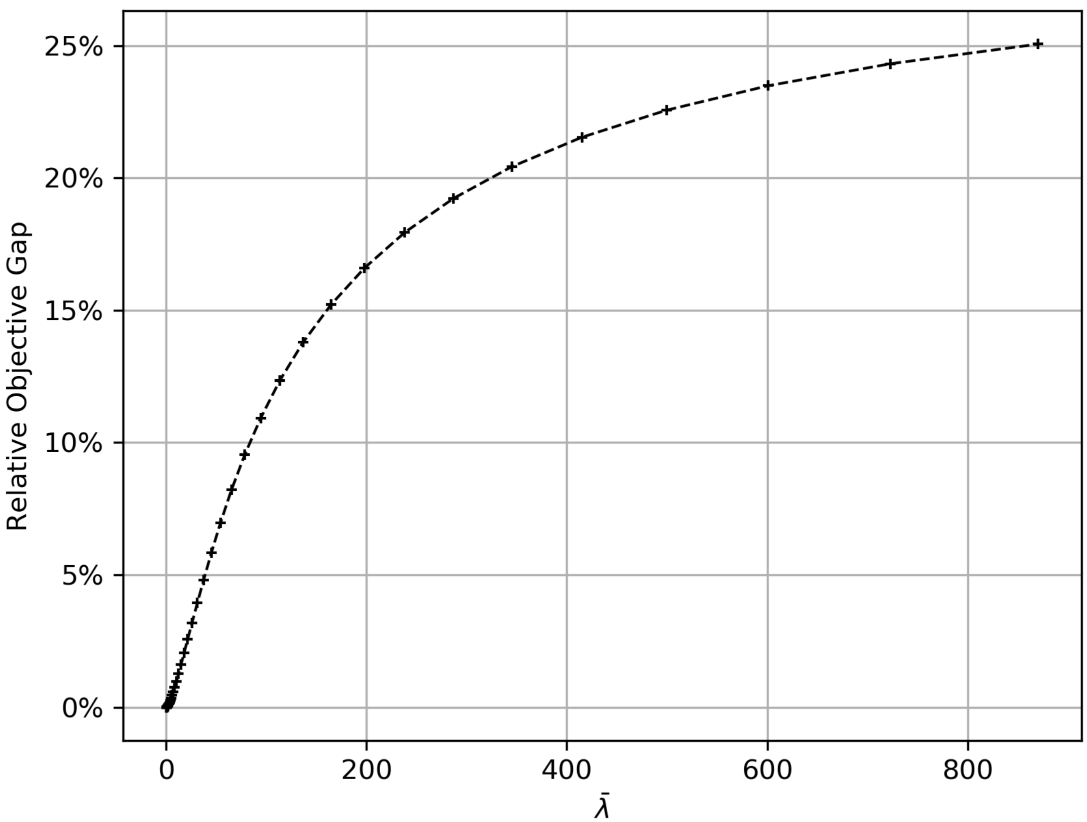

FIGURE 9: Relative Objective Gap (%).

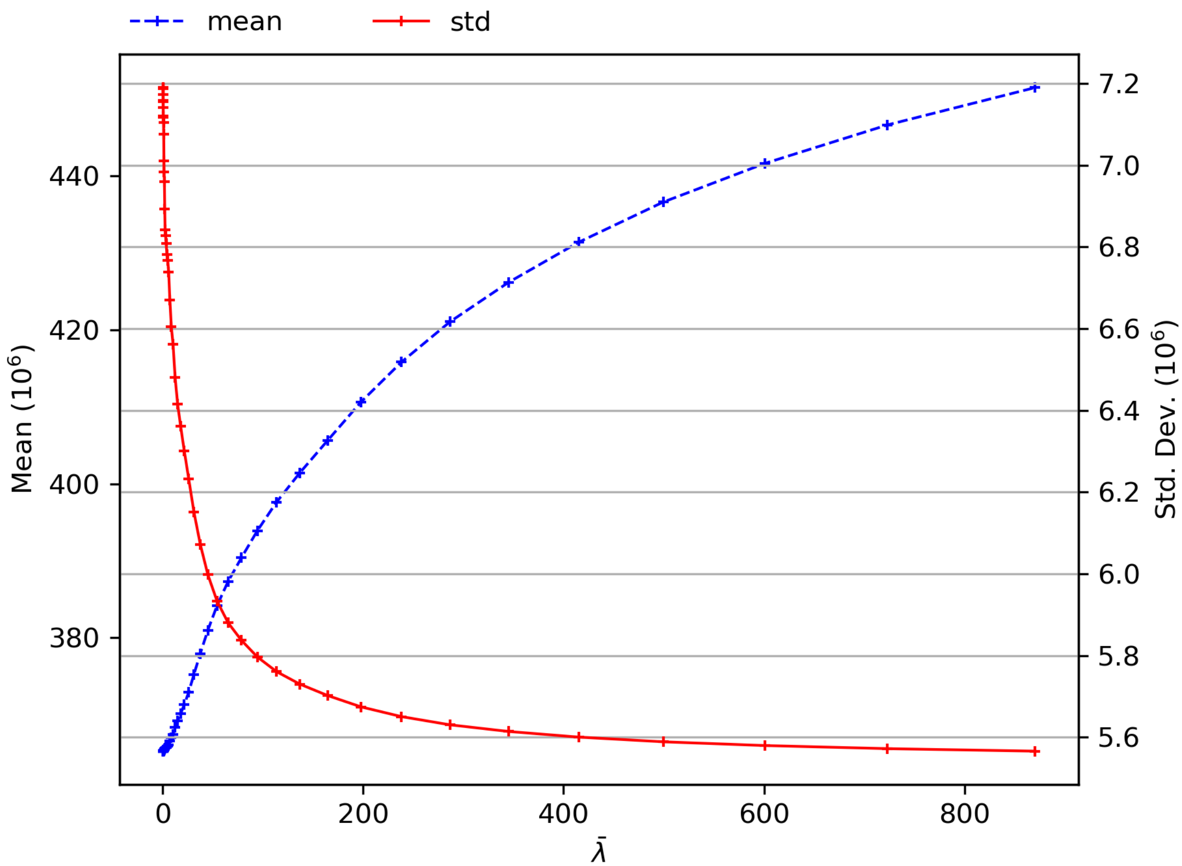

FIGURE 10: Criteria trade-off.

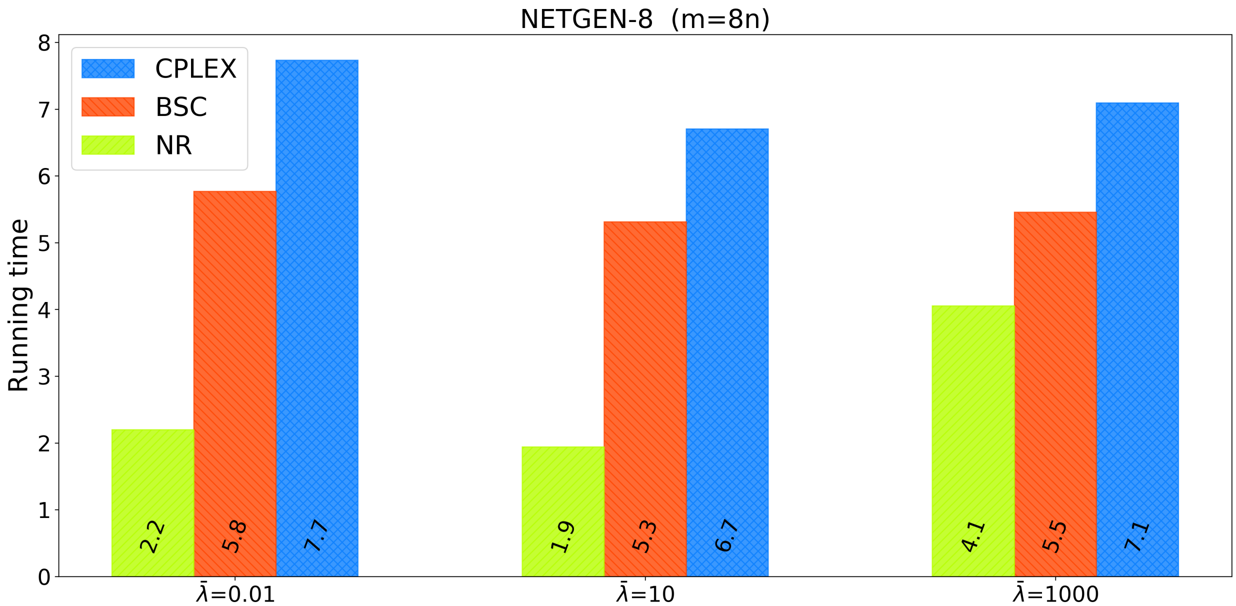

FIGURE 11: Sensivity to λ on NETGEN-8 instances.

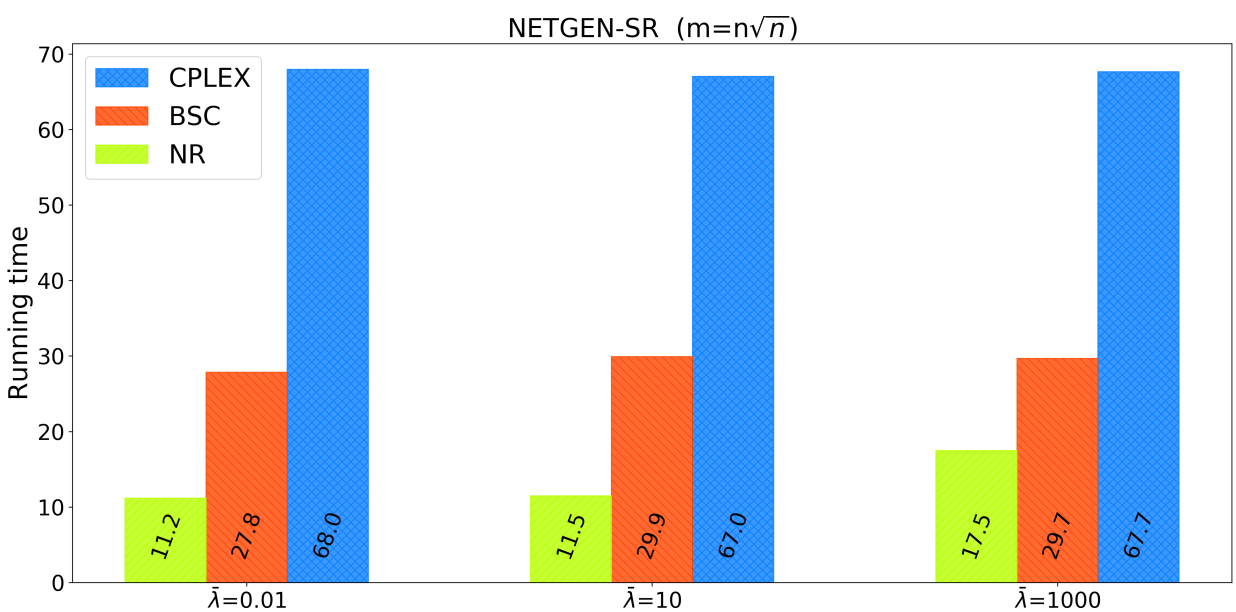

FIGURE 12: Sensivity to λ on NETGEN-SR instances.

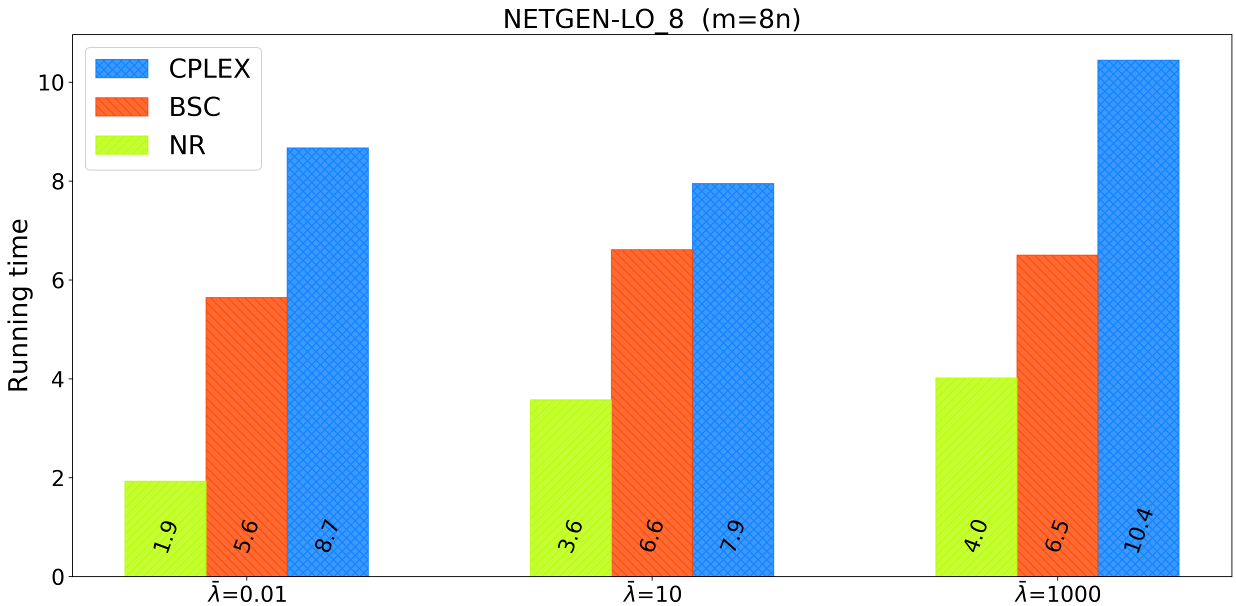

FIGURE 13: Sensivity to λ on NETGEN-LO-8 instances.

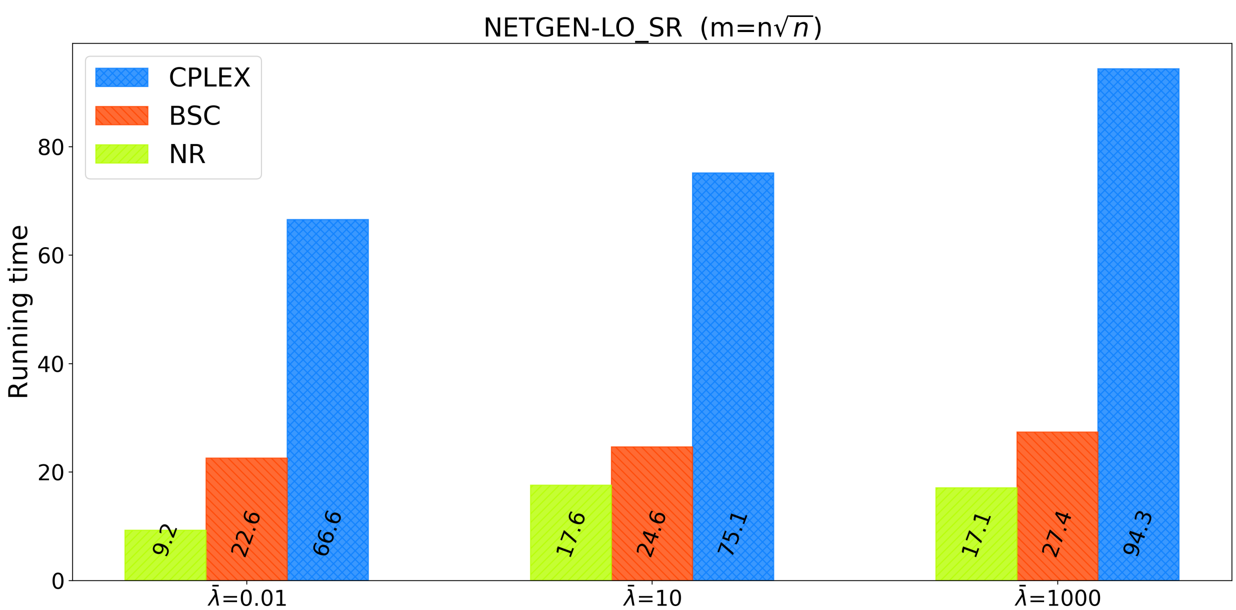

FIGURE 14: Sensivity to λ on NETGEN-LO-SR instances.

6 CONCLUSION

This paper described three solution algorithms for the MSDMCF. The proposed methods solve the MSDMCF problem by repeatedly solving easier MVMCF problems. The effectiveness of the algorithms relies on the number of easier MVMCF problems required to be solved until the particular weight parameter λ is found. The algorithms differ in the root-finding method that they use. We also provide a procedure to find tighter upper and lower bounds for the root-finding methods, which is shown to improve the performance significantly. Amongst all, the BSC method is the simplest to implement. However, it needs more iterations to converge compared to the NR method. In contrast, the NR method requires solution derivatives, which can be obtained through sensitivity analysis. In each iteration of the NR method, we thus solve two problems, one subproblem and one auxiliary problem for finding the derivatives. The starting λ for the NR method is crucial, as starting far from the root may cause divergent behavior. In order to alleviate this potentially divergent behavior of the pure Newton method, we also provide a "failsafe" Hybrid method. These algorithms can also be applied to more general GNPMCF problems.

In our experiments, we compare running times of the algorithms to achieve a gap level of 0.01%. The NR method outperforms CPLEX and BSC on every problem instance. In contrast, BSC method outperforms CPLEX for small instances of dense network families, while performing competitively or worse for larger instances. Another advantage of the NR and BSC methods is achieving very good solution, very fast. This can even be improved by changing the strategy to find the initial λ. Spending less effort for finding an initial parameter for the algorithms to start with, will result in time savings while trading off with solution quality.

The runtime of the proposed algorithms provided in this paper can be further improved in several ways. We used CPLEX solver to solve the MVMCF subproblems. However, faster solution methods that exploits the additive structure of these subproblems could reduce runtime significantly. Another approach could be finding ways to improve the root-finding procedure, possibly exploring or modifying the methods to descend even faster than the ones provided. One can also do analysis on early stopping for early iterations in the proposed methods. The framework can used for any problem with linear constraints and continuous variables, where the objective function meets the requirements. Other potential directions for future research is to investigate the case where the second criterion is concave and differentiable, and considering dependent arc costs.

7 ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation under grants CMMI-1562109/1562291 and CMMI-1254921, and the Data-Supported Transportation and Planning University Transportation Center.

REFERENCES

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1993.

[2] Steven A Gabriel and David Bernstein. The traffic equilibrium problem with nonadditive path costs. *Transportation Science*, 31(4):337–348, 1997.

[3] Mogens Fosgerau and Anders Karlström. The value of reliability. *Transportation Research Part B: Methodological*, 44(1):38–49, 2010.

[4] Mogens Fosgerau and Leonid Engelson. The value of travel time variance. *Transportation Research Part B: Methodological*, 45(1):1–8, 2011.

[5] Peng Chen and Yu Marco Nie. Bicriterion shortest path problem with a general nonadditive cost. *Transportation Research Part B: Methodological*, 57:419–435, 2013.

[6] Tao Xing and Xuesong Zhou. Finding the most reliable path with and without link travel time correlation: A lagrangian substitution based approach. *Transportation Research Part B: Methodological*, 45(10):1660–1679, 2011.

[7] Suvrajeet Sen, Rekha Pillai, Shirish Joshi, and Ajay K Rathi. A mean-variance model for route guidance in advanced traveler information systems. *Transportation Science*, 35(1):37–49, 2001.

[8] Raj A Sivakumar and Rajan Batta. The variance-constrained shortest path problem. *Transportation Science*, 28(4):309–316, 1994.

[9] Kevin R Hutson and Douglas R Shier. Extended dominance and a stochastic shortest path problem. *Computers & Operations Research*, 36(2):584–596, 2009.

[10] Mehrdad Shahabi, Avinash Unnikrishnan, and Stephen D Boyles. An outer approximation algorithm for the robust shortest path problem. *Transportation Research Part E: Logistics and Transportation Review*, 58:52–66, 2013.

[11] Yuli Zhang, Zuo-Jun Max Shen, and Shiji Song. Lagrangian relaxation for the reliable shortest path problem with correlated link travel times. *Transportation Research Part B: Methodological*, 104:501–521, 2017.

[12] Yufeng Zhang and Alireza Khani. An algorithm for reliable shortest path problem with travel time correlations. *Transportation Research Part B: Methodological*, 121:92–113, 2019.

[13] Ishwar Murthy and Sumit Sarkar. A relaxation-based pruning technique for a class of stochastic shortest path problems. *Transportation Science*, 30(3):220–236, 1996.

[14] Ishwar Murthy and Sumit Sarkar. Exact algorithms for the stochastic shortest path problem with a decreasing deadline utility function. *European Journal of Operational Research*, 103(1):209–229, 1997.

[15] Yueyue Fan, Robert E Kalaba, and James E Moore. Arriving on time. *Journal of Optimization Theory and Applications*, 127(3):497–513, 2005.

[16] Yueyue Fan and Yu Marco Nie. Optimal routing for maximizing the travel time reliability. *Networks and Spatial Economics*, 6(3-4):333–344, 2006.

[17] Yu Nie and Yueyue Fan. Arriving-on-time problem: discrete algorithm that ensures convergence. *Transportation Research Record: Journal of the Transportation Research Board*, (1964):193–200, 2006.

[18] Yu Marco Nie and Xing Wu. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological*, 43(6):597–613, 2009.

[19] Bi Yu Chen, William HK Lam, Agachai Sumalee, Qingquan Li, and Mei Lam Tam. Reliable shortest path problems in stochastic time-dependent networks. *Journal of Intelligent Transportation Systems*, 18(2):177–189, 2014.

[20] Karthik K Srinivasan, AA Prakash, and Ravi Seshadri. Finding most reliable paths on networks with correlated and shifted log-normal travel times. *Transportation Research Part B: Methodological*, 66:110–128, 2014.

[21] Lixing Yang and Xuesong Zhou. Optimizing on-time arrival probability and percentile travel time for elementary path finding in time-dependent transportation networks: Linear mixed integer programming reformulations. *Transportation Research Part B: Methodological*, 96:68–91, 2017.

[22] Tao Xing and Xuesong Zhou. Reformulation and solution algorithms for absolute and percentile robust shortest path problems. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):943–954, 2013.

[23] Yu Marco Nie, Xing Wu, and Tito Homem-de Mello. Optimal path problems with second-order stochastic dominance constraints. *Networks and Spatial Economics*, 12(4):561–587, 2012.

[24] Xing Wu. Study on mean-standard deviation shortest path problem in stochastic and time-dependent networks: A stochastic dominance based approach. *Transportation Research Part B: Methodological*, 80:275–290, 2015.

[25] Leilei Zhang and Tito Homem-de Mello. An optimal path model for the risk-averse traveler. *Transportation Science*, 51(2):518–535, 2016.

[26] Anthony Chen and Zhong Zhou. The α-reliable mean-excess traffic equilibrium model with stochastic travel times. *Transportation Research Part B: Methodological*, 44(4):493–513, 2010.

[27] Fernando Ordóñez and Nicolás E Stier-Moses. Wardrop equilibria with risk-averse users. *Transportation Science*, 44(1):63–86, 2010.

[28] Chao Zhang, Xiaojun Chen, and Agachai Sumalee. Robust wardrop's user equilibrium assignment under stochastic demand and supply: expected residual minimization approach. *Transportation Research Part B: Methodological*, 45(3):534–552, 2011.

[29] Yu Marco Nie. Multi-class percentile user equilibrium with flow-dependent stochasticity. *Transportation Research Part B: Methodological*, 45(10):1641–1659, 2011.

[30] Evdokia Nikolova and Nicolás E Stier-Moses. A mean-risk model for the traffic assignment problem with stochastic travel times. *Operations Research*, 62(2):366–382, 2014.

[31] Judith YT Wang, Matthias Ehrgott, and Anthony Chen. A bi-objective user equilibrium model of travel time reliability in a road network. *Transportation Research Part B: Methodological*, 66:4–15, 2014.

[32] Ravi Seshadri and Karthik K Srinivasan. Robust traffic assignment model: formulation, solution algorithms and empirical application. *Journal of Intelligent Transportation Systems*, 21(6):507–524, 2017.

[33] A Arun Prakash, Ravi Seshadri, and Karthik K Srinivasan. A consistent reliability-based user-equilibrium problem with risk-averse users and endogenous travel time correlations: formulation and solution algorithm. *Transportation Research Part B: Methodological*, 114:171–198, 2018.

[34] Stephen D Boyles and S Travis Waller. A mean-variance model for the minimum cost flow problem with stochastic arc costs. *Networks*, 56(3):215–227, 2010.

[35] Alper Atamtürk and Muhong Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

[36] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical programming*, 98(1–3):49–71, 2003.

[37] Dimitris Bertsimas, Ebrahim Nasrabadi, and Sebastian Stiller. Robust and adaptive network flows. *Operations Research*, 61(5):1218–1242, 2013.

[38] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28. Princeton University Press, 2009.

[39] John R Birge and James K Ho. Optimal flows in stochastic dynamic networks with congestion. *Operations Research*, 41(1):203–216, 1993.

[40] Gregory D Glockner and George L Nemhauser. A dynamic network flow problem with uncertain arc capacities: formulation and problem structure. *Operations Research*, 48(2):233–242, 2000.

[41] Jsen-Shung Lin, Chin-Chia Jane, and John Yuan. On reliability evaluation of a capacitated-flow network in terms of minimal pathsets. *Networks*, 25(3):131–138, 1995.

[42] Jsen-shung Lin. Reliability evaluation of capacitated-flow networks with budget constraints. *IIE Transactions*, 30(12):1175–1180, 1998.

[43] Yi-Kuei Lin. Using minimal cuts to evaluate the system reliability of a stochastic-flow network with failures at nodes and arcs. *Reliability Engineering & System Safety*, 75(1):41–46, 2002.

[44] Yi-Kuei Lin. A simple algorithm for reliability evaluation of a stochastic-flow network with node failure. *Computers & Operations Research*, 28(13):1277–1285, 2001.

[45] Yi-Kuei Lin. On a multicommodity stochastic-flow network with unreliable nodes subject to budget constraint. *European Journal of Operational Research*, 176(1):347–360, 2007.

[46] Yi-Kuei Lin. 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, 2007.

[47] Fernando A Kuipers, Song Yang, Stojan Trajanovski, and Ariel Orda. Constrained maximum flow in stochastic networks. In *2014 IEEE 22nd International Conference on Network Protocols*, pages 397–408. IEEE, 2014.

[48] Alireza Khani and Stephen D Boyles. An exact algorithm for the mean-standard deviation shortest path problem. *Transportation Research Part B: Methodological*, 81:252–266, 2014.

[49] Yuli Zhang, Zuo-Jun M Shen, and Shiji Song. Parametric search for the bi-attribute concave shortest path problem. *Transportation Research Part B: Methodological*, 94:150–168, 2016.

[50] Haijune Lee and P Simin Pulat. Bicriteria network flow problems: Continuous case. *European Journal of Operational Research*, 51(1):119–126, 1991.

[51] P Simin Pulat, Fenghueih Huarng, and Haijune Lee. Efficient solutions for the bicriteria network flow problem. *Computers & Operations Research*, 19(7):649–655, 1992.

[52] Antonio Sedeño-Noda and C González-Martın. The biobjective minimum cost flow problem. *European Journal of Operational Research*, 124(3):591–600, 2000.

[53] Horst W Hamacher, Christian Roed Pedersen, and Stefan Ruzika. Multiple objective minimum cost flow problems: A review. *European Journal of Operational Research*, 176(3):1404–1422, 2007.

[54] Andrea Raith and Matthias Ehrgott. A two-phase algorithm for the biobjective integer minimum cost flow problem. *Computers & Operations Research*, 36(6):1945–1954, 2009.

[55] Augusto Eusébio and José Rui Figueira. Finding non-dominated solutions in bi-objective integer network flow problems. *Computers & Operations Research*, 36(9):2554–2564, 2009.

[56] Augusto Eusébio, José Rui Figueira, and Matthias Ehrgott. On finding representative non-dominated points for bi-objective integer network flow problems. *Computers & Operations Research*, 48:1–10, 2014.

[57] Siamak Moradi, Andrea Raith, and Matthias Ehrgott. A bi-objective column generation algorithm for the multi-commodity minimum cost flow problem. *European Journal of Operational Research*, 244(2):369–378, 2015.

[58] Andrea Raith and Antonio Sedeño-Noda. Finding extreme supported solutions of biobjective network flow problems: An enhanced parametric programming approach. *Computers & Operations Research*, 82:153–166, 2017.

[59] Dimitri P Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

[60] Lawrence M Ausubel and Raymond J Deneckere. A generalized theorem of the maximimum. *Economic Theory*, 3(1), 1993.

[61] Stephen D Boyles. Bush-based sensitivity analysis for approximating subnetwork diversion. *Transportation Research Part B*, 46:139–155, 2012.

[62] Ehsan Jafari and Stephen D Boyles. Improved bush-based methods for network contraction. *Transportation Research Part B*, 83:298–313, 2016.

[63] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.).* Cambridge University Press., 1992.

[64] Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimization problems. *Int. Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194.

[65] D. Klingman, A. Napier, and J. Stutz. Netgen: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems. *Management Science*, 20(5):814–821, 1974.

[66] Zoltán Király and Péter Kovács. Efficient implementations of minimum-cost flow algorithms. *Acta Universitatis Sapientiae, Informatica*, 4, 2012.

[67] John Bates. Challenges and accomplishments of modeling impacts of policy initiatives. *In Association for European Transport and contributors*, 2008.