

Mean-Standard Deviation Model For Minimum Cost Flow Problem

Can Gokalp¹, Stephen D. Boyles¹, and Avinash Unnikrishnan²

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

²Department of Civil and Environmental Engineering, Portland State University

November 6, 2022

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 the non-linearity 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 — bisection (BSC), Newton-Raphson (NR), and a hybrid of the two (NR-BSC) — to solve the MSDMCF problem by solving multiple MVMCF subproblems. While the methods all seek to find the specific MVMCF problem whose optimal solution coincides with the optimal solution for the given MSDMCF problem, they differ in the method used 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 NR algorithm is about twice as fast as the solver on the original problem.

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 various 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 [22, 21, 20, 15], 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 [66, 58, 61, 26, 60, 73, 71], maximizing probability of arrival or disutility associated with a pre-specified arrival time [43, 44, 18, 19, 46, 48, 13, 62, 68], percentiles [67, 68], risk aversion [49, 65, 70], and so forth. Reliability and risk related objectives have also been incorporated into traffic assignment models [12, 51, 69, 47, 50, 64, 59, 52].

There has been relatively less research on incorporating reliability objectives into other traditional minimum cost flow and max flow network problems. Boyles and Waller [11] studied a specific instance of the convex MCF problem, with independent uncertain arc costs, where the aim is to minimize a linear combination of the mean and the variance of the total flow costs, termed 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 defined arc marginal costs and used them to modify the generic cycle canceling algorithm. Their objective was non-linear, but separable by arc. This separability property was critical for their algorithms.

One drawback to the mean-variance approach is that the objective is hard to interpret because the mean and variance carry different units. As a result, it is unclear how the two objectives should be weighted. However, since the mean and standard deviation have common units, their linear combination is easier to understand. Furthermore, under an additional assumption on the arc cost distribution, a linear combination of mean and standard deviation can also be interpreted as a percentile of the flow cost. For instance, if link costs are normally distributed, minimizing the sum of mean and standard deviation is equivalent to minimizing the 84th-percentile cost. Therefore, we choose to study the mean-standard deviation formulation in this article, despite the computational challenges relative to the mean-variance formulation.

The general case of MCF problems with costs that are strictly convex, differentiable, and separable by arc is studied in [45]. They derive optimality conditions and provide a primal-dual algorithm. Another approach is to transform the problem to the traditional linear MCF problem by using piecewise linearization of the arc cost functions and use existing linear MCF solution methods [41, 29]. More recently, Vég  [63] describes a strongly polynomial 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. For simplicity, we assume that arc costs are independent; while our procedure is still *correct* even in the presence of correlations, testing whether it remains *efficient* in this case is beyond the scope of this paper.

While the mean-standard deviation objective function is still convex, it is not separable by arc. Therefore, the approaches to the convex separable version of the problem mentioned above are not applicable. This type of convex non-separable flow problem can still be solved in polynomial time, although not strongly polynomial. The best running time reported for such problems is $O(m^3L)$, where L is the total length of the input coefficients and m is the number of arcs [25]. In this paper, we adopt a different approach. 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 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 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 [2], costs and capacities [8], and network structure [7]. 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 realizations of the uncertain parameters from their pre-specified sets. The shape of the uncertainty set indicates the decision maker's risk preference and affects the tractability of the model [5]. Birge [9] and Glockner [23] applied a multi-stage stochastic programming approach to model uncertainties in network parameters in a stochastic and dynamic network flow setting. More recently, distributionally robust optimization techniques have been getting increased attention. These approaches do not require complete knowledge of the probability distributions, but rather the optimization is performed over the ambiguity set of probability distributions consistent with prior beliefs about the uncertain parameters. Chen et al.[14] applied these techniques to a maximum flow problem. A challenge in robust optimization is that planning on worst-case scenarios can lead to overly conservative solutions.

Stochastic programming, yet another approach to addressing uncertainty, requires knowledge of the probability distribution of uncertain parameters. In contrast, we only 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.

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 [35, 36, 38]. Lin et al. [37] studied the stochastic maximum flow problem, where the nodes and arcs have uncertain discrete capacities, and developed an algorithm to compute the system reliability defined as the probability that the maximum flow is greater than the given demand. Lin [39] focused on the multi-commodity variant of [37] and defined a system reliability objective as the probability that the upper bound of system capacity equals a given pattern, subject to budget constraints on flows. Along similar lines, Lin [40] adopted 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 [33] formulated two stochastic maximum flow models: maximum flow in stochastic networks, where the bandwidth or capacity has a log-concave probability distribution, and the maximum delay constrained flow problem, where an additional stochastic delay constraint is imposed on the flows. A convex formulation and polynomial-time algorithm were provided for the former problem, while the latter was shown NP-hard and solved using an approximation algorithm. The MSDMCF model presented in our 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 the 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 $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed network with $\mathcal{N}$ and $\mathcal{A}$ denoting the set of nodes and arcs, respectively, with $m = |\mathcal{A}|$ and $n = |\mathcal{N}|$. The arc costs c_{ij} are stochastic with known means $E[c_{ij}]$ and variances $Var[c_{ij}]$. Let $\bar{\lambda}$ denote a (strictly positive) weighing parameter. Nodes and arcs are assumed to have deterministic demands b_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:

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

or, more compactly,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \boldsymbol{\mu}^T \mathbf{x} + \bar{\lambda} \sqrt{\mathbf{x}^T \mathbf{V} \mathbf{x}} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{aligned} \quad (\text{MSDMCF}(\bar{\lambda}))$$

with

$$\boldsymbol{\mu} = \begin{bmatrix} E[c_1] \\ \vdots \\ E[c_m] \end{bmatrix}, \quad \mathbf{V} = \text{diag}(\mathbf{Var}) = \begin{bmatrix} Var[c_1] & 0 & 0 \\ & \ddots & \\ 0 & 0 & Var[c_m] \end{bmatrix}, \quad \bar{\lambda} \geq 0,$$

and $\mathbf{A} \mathbf{x} = \mathbf{b}$ representing the flow conservation equations, and with at least one $b_j > 0$ to exclude trivial instances. We will assume that $\mathbf{V}$ is positive definite, implying that every feasible solution has a strictly positive variance; as discussed below, our proposed approach may not apply when this is not the case. In practice, this assumption is unlikely to be limiting, as links with zero variance can be assigned a sufficiently small variance; and indeed, “risk-free” options in the real world are invariably subject to some unlikely “tail events.” 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 in the definition of convexity.

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

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

or, compactly,

$$\begin{aligned}
& \min_{\mathbf{x}} \quad \boldsymbol{\mu}^T \mathbf{x} + \lambda \mathbf{x}^T \mathbf{V} \mathbf{x} \\
& \text{s.t.} \quad \mathbf{A} \mathbf{x} = \mathbf{b} \\
& \quad \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}
\end{aligned} \tag{MVMCF(\lambda)}$$

with non-negative λ . Unlike the MSDMCF, the MVMCF problem is separable by arc. Both of the problems are convex; however, the separability structure is exploitable algorithmically [11].

2.2 Proposed approach

We adopt a parametric search method to solve the MSDMCF problem. Given any positive value of $\bar{\lambda}$, 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 use λ^* to refer to a value of this parameter that produces an optimal solution to MSDMCF($\bar{\lambda}$); we will show that this value is unique. Our approach is similar in spirit to methods that have previously been applied for the mean-standard deviation shortest path problem [30, 72]. However, since the MSDMCF is a continuous optimization problem, rather than a combinatorial optimization problem, existing results on the mean-standard deviation shortest path problem 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 the 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 $\mathcal{X}$ represents the MCF problem feasible set, μ and v are separable and differentiable functions, g is a strictly monotone, increasing and differentiable function, with the composition $g \circ v(\mathbf{x}) = g(v(\mathbf{x}))$ convex. 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 optimal solutions to the MSDMCF problem are also optimal to the MVMCF problem for a particular choice of the weight parameter. We also show that the converse of this claim is true, unlike the mean-standard deviation shortest path problem.
2. We derive a key equation characterizing the relationship between the optimal solutions of the two problems, and develop three algorithms for finding the particular weight parameter λ^* to the MVMCF problem for which the optimal solution is also optimal to the MSDMCF problem for a given $\bar{\lambda}$. We further analyze the resulting function to show that a bisection algorithm will always work, and converges to a unique root.
3. We further show that our results can be extended to a more general class of MCF problems.

This model differs from the bi-objective MCF literature [34, 54, 57, 24, 55, 16, 17, 42, 56] 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 model, the two objectives can be collapsed into a single objective using a weight parameter, and we do not directly attempt to find the set of non-dominated solutions. Nevertheless, we show in the appendix that all non-dominated solutions are solutions to MSDNFP (or MVNFP) for an appropriate choice of weighting parameter, and therefore can be found using our methods.

We expect this method to be efficient because the independence assumption means that MVMCF is a network flow problem with a separable, convex, quadratic objective, for which there is a strongly polynomial, $O(m^4 \log m)$ time algorithm [63]. For the case of general correlation, there is $O(m^5)$ time algorithm [25]. Solving MSDMCF directly is likely to be slower, since the objective is neither separable nor quadratic, and we are unaware of a strongly polynomial-time algorithm for this problem.

2.3 Relevance to the MVMCF

In this section, we will show that for any 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 $\ell(\mathbf{x}) = \mathbf{b} - \mathbf{A}\mathbf{x}$ and $\mathbf{h}(\mathbf{x}) = \mathbf{x} - \mathbf{u}$. 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:

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

where η_{ij} and p_i are the dual variables for the capacity and flow balance constraints, respec-

tively. Next, the stationary conditions are:

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

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

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 and the constraints are all linear, these necessary conditions are also sufficient [6]. Our main result now follows.

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

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

Proof. As $\mathbf{x}(\lambda)$ satisfies the KKT necessary conditions for MVMCF(λ), there exist $\boldsymbol{\eta}(\lambda)$ and $\mathbf{p}(\lambda)$ such that

$$-\boldsymbol{\mu} - \sum_{(i,j) \in \mathcal{A}} \eta_{ij}(\lambda) \nabla_{\mathbf{x}} h_{ij}(\mathbf{x}(\lambda)) - \sum_{i \in \mathcal{N}} p_i(\lambda) \nabla_{\mathbf{x}} \ell_i(\mathbf{x}(\lambda)) = \lambda 2 \mathbf{V} \mathbf{x}(\lambda). \quad (5)$$

and the complementary conditions (1) are satisfied. Since the constraint systems for MVMCF and MSDMCF are the same (regardless of weight parameters), $\mathbf{x}(\lambda)$ also satisfies the complementary conditions for MSDMCF($\bar{\lambda}$) with the same $\boldsymbol{\eta}(\lambda)$ and $\mathbf{p}(\lambda)$.

Now, if $\bar{\lambda} = 2\lambda \sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}$, then equation (5) is equivalent to

$$-\boldsymbol{\mu} - \sum_{(i,j) \in \mathcal{A}} \eta_{ij}(\lambda) \nabla_{\mathbf{x}} h_{ij}(\mathbf{x}(\lambda)) - \sum_{i \in \mathcal{N}} p_i(\lambda) \nabla_{\mathbf{x}} \ell_i(\mathbf{x}(\lambda)) = \frac{\bar{\lambda} \mathbf{V} \mathbf{x}(\lambda)}{\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}. \quad (6)$$

But this is exactly the stationary condition for MSDMCF($\bar{\lambda}$), so all of its KKT necessary conditions are also satisfied. Since the objective function for MSDMCF($\bar{\lambda}$) is convex, these conditions are also sufficient for optimality.

□

The same argument shows that an optimal solution $\mathbf{x}(\bar{\lambda})$ to $\text{MSDMCF}(\bar{\lambda})$ is also optimal for $\text{MVMCF}(\lambda)$ if the same relation holds between λ and $\bar{\lambda}$, and that therefore the sets of optimal solutions to the mean-variance and mean-standard deviation problems coincide. In fact, the sets of optimal solutions to these problems are the full sets of Pareto-optimal solutions, as shown in the appendix. This distinguishes our problem from the mean-standard deviation shortest path problem, where the analogous statement fails [30] (there are mean-variance shortest paths which are not mean-standard deviation shortest paths for any weighting). Therefore, the MVMCF and the MSDMCF have a closer relationship than the corresponding shortest path problems.

3 ALGORITHMS

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

$$f(\lambda) \equiv 2\lambda\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)} - \bar{\lambda} = 0, \quad (7)$$

solving $\text{MVMCF}(\lambda)$ will solve the original $\text{MSDMCF}(\bar{\lambda})$. Therefore, solving $\text{MSDMCF}(\bar{\lambda})$ reduces to finding a root of $f(\lambda)$. As shown below, such a root always exists and is unique, and f is strictly increasing. This section first addresses these issues of existence and uniqueness, and then the issue of selecting an initial guess or interval containing the root — these issues are common to all three of the root-finding algorithms we present. We then present BSC, NR, and NR-BSC in turn.

We first show the existence of 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 $\lambda \in [0, \infty)$. In what follows, we use $M(\lambda) \equiv \boldsymbol{\mu}^T \mathbf{x}(\lambda)$ to refer to the mean cost of the optimal solution to $\text{MVMCF}(\lambda)$, and $V(\lambda) \equiv \mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)$. It is trivial to see that f takes a negative value as λ approaches 0, since $\bar{\lambda}$ is positive and $V(\lambda)$ is finite for any value of λ . Let $V(\infty)$ represent the minimum variance of any feasible flow (this notation reflects that this value can be computed by solving an instance of MVMCF with only the variance term). Then we know $V(\lambda) \geq V(\infty)$, which is

positive by assumption. We thus conclude that $f(\lambda)$ takes a positive value as λ approaches ∞ .

Finally, since the objective function of $\text{MVMCF}(\lambda)$ is continuous in both $\mathbf{x}$ and λ and strictly convex in $\mathbf{x}$, the minimizer $\mathbf{x}(\lambda)$ is well-defined and continuous in λ by the Maximum Theorem [3]. Therefore $f(\lambda)$ is also continuous, and at least one root exists in $[0, \infty)$.

Furthermore, this root is unique, as shown in the following results.

Lemma 1. *The mean cost $M(\lambda)$ of the optimal solution is nondecreasing in λ , while $V(\lambda)$ is nonincreasing.*

Proof. Let λ_1 and λ_2 be distinct weighting parameters. Without loss of generality we can assume $0 \leq \lambda_1 < \lambda_2$. Since $\mathbf{x}(\lambda_1)$ minimizes $\boldsymbol{\mu}^T \mathbf{x} + \lambda_1 \mathbf{x}^T \mathbf{V} \mathbf{x}$, we have

$$M(\lambda_1) + \lambda_1 V(\lambda_1) \leq M(\lambda_2) + \lambda_1 V(\lambda_2). \quad (8)$$

Similarly,

$$M(\lambda_2) + \lambda_2 V(\lambda_2) \leq M(\lambda_1) + \lambda_2 V(\lambda_1). \quad (9)$$

Multiplying inequality (8) by λ_2 , inequality (9) by λ_1 , and subtracting gives

$$(\lambda_2 - \lambda_1)M(\lambda_1) \leq (\lambda_2 - \lambda_1)M(\lambda_2), \quad (10)$$

whence it follows that $M(\lambda_1) \leq M(\lambda_2)$, that is, M is nondecreasing.

Furthermore, since $M(\lambda_1) \leq M(\lambda_2)$, inequality (9) can only be satisfied if $V(\lambda_1) \geq V(\lambda_2)$, showing that V is nonincreasing and completing the lemma. $\square$

Theorem 2. *The function $f(\lambda) = 2\lambda\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)} - \bar{\lambda}$ has exactly one root.*

Proof. The above discussion establishes the existence of a root; we now show that this root is unique.

By contradiction, assume that $f(\lambda_1) = f(\lambda_2) = 0$ for some $\lambda_1 \neq \lambda_2$. Theorem 1 ensures that $\mathbf{x}_1 \equiv \mathbf{x}(\lambda_1)$ and $\mathbf{x}_2 \equiv \mathbf{x}(\lambda_2)$ are both optimal to $\text{MSDMCF}(\bar{\lambda})$. Since this problem is convex, the set of optimal solutions is convex, and $(1 - \alpha)\mathbf{x}_1 + \alpha\mathbf{x}_2$ is optimal as well for any $\alpha \in [0, 1]$.

Since all these solutions are optimal, they all have equal objective function values, so there is a constant k such that

$$k = \boldsymbol{\mu}^T((1 - \alpha)\mathbf{x}_1 + \alpha\mathbf{x}_2) + \bar{\lambda}\sqrt{((1 - \alpha)\mathbf{x}_1 + \alpha\mathbf{x}_2)^T \mathbf{V}((1 - \alpha)\mathbf{x}_1 + \alpha\mathbf{x}_2)} \quad (11)$$

for $\alpha \in [0, 1]$, and in particular $k - \boldsymbol{\mu}^T \mathbf{x}_1 = \bar{\lambda}\sqrt{\mathbf{x}_1^T \mathbf{V} \mathbf{x}_1}$ and $k - \boldsymbol{\mu}^T \mathbf{x}_2 = \bar{\lambda}\sqrt{\mathbf{x}_2^T \mathbf{V} \mathbf{x}_2}$.

Furthermore, for $\alpha = 1/2$, we have

$$\bar{\lambda}\sqrt{\left(\frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2\right)^T \mathbf{V} \left(\frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2\right)} = (k - \boldsymbol{\mu}^T \mathbf{x}_1) - \frac{1}{2}\boldsymbol{\mu}^T(\mathbf{x}_2 - \mathbf{x}_1) \quad (12)$$

or, after simplifying,

$$\frac{1}{2}\bar{\lambda}\sqrt{(\mathbf{x}_1 + \mathbf{x}_2)^T \mathbf{V} (\mathbf{x}_1 + \mathbf{x}_2)} = \frac{1}{2}\bar{\lambda}\sqrt{\mathbf{x}_1^T \mathbf{V} \mathbf{x}_1} + \frac{1}{2}\bar{\lambda}\sqrt{\mathbf{x}_2^T \mathbf{V} \mathbf{x}_2}. \quad (13)$$

Since $\bar{\lambda}$ is strictly positive, we have

$$\sqrt{(\mathbf{x}_1 + \mathbf{x}_2)^T \mathbf{V} (\mathbf{x}_1 + \mathbf{x}_2)} = \sqrt{\mathbf{x}_1^T \mathbf{V} \mathbf{x}_1} + \sqrt{\mathbf{x}_2^T \mathbf{V} \mathbf{x}_2}. \quad (14)$$

Squaring both sides and simplifying gives

$$\mathbf{x}_1^T \mathbf{V} \mathbf{x}_2 = \sqrt{(\mathbf{x}_1^T \mathbf{V} \mathbf{x}_1)(\mathbf{x}_2^T \mathbf{V} \mathbf{x}_2)}. \quad (15)$$

But $\mathbf{V}$ is also positive definite, so $\mathbf{x}^T \mathbf{V} \mathbf{y}$ forms an inner product space. The Cauchy-Schwarz inequality therefore asserts that (15) holds only if $\mathbf{x}_1$ and $\mathbf{x}_2$ are linearly dependent, that is, if $\mathbf{x}_1 = \beta \mathbf{x}_2$ for some $\beta \neq 0$. The only choice that satisfies the flow conservation constraints for both $\mathbf{x}_1$ and $\mathbf{x}_2$ is $\beta = 1$; therefore $\mathbf{x}_1 = \mathbf{x}_2$, and the solutions corresponding to λ_1 and λ_2 are in fact identical.

As a result, $V(\lambda_1) = V(\lambda_2)$. But $f(\lambda) = 2\lambda\sqrt{V(\lambda)} - \bar{\lambda}$, so $f(\lambda_1) = f(\lambda_2) = 0$ would imply $\lambda_1 = \lambda_2$, a contradiction. $\square$

With this result in hand, we can show that f is strictly increasing and therefore takes unique values for all λ .

Corollary 1. *The function $f(\lambda)$ is strictly increasing.*

Proof. We will show that $f : [0, \infty) \rightarrow [-\bar{\lambda}, \infty)$ is a bijection, using the fact that Theorems 1 and 2 are valid for any $\bar{\lambda} > 0$. The arguments above established that f is continuous, $f(0) = -\bar{\lambda}$ and $f(\lambda) \rightarrow \infty$ as $\lambda \rightarrow \infty$. Taken together, these prove the result.

The surjectivity of f follows immediately from continuity and its limiting values. To show injectivity, assume $f(\lambda_1) = f(\lambda_2) = z$. If $z = -\bar{\lambda}$, then we immediately have $\lambda_1 = \lambda_2 = 0$ since all solutions have positive variance. Otherwise, $z > -\bar{\lambda}$, so an instance of MSDMCF with weight parameter $\bar{\lambda} + z$ is valid. Then

$$2\lambda_1\sqrt{V(\lambda_1)} - (\bar{\lambda} + z) = 2\lambda_2\sqrt{V(\lambda_2)} - (\bar{\lambda} + z) = 0,$$

so $\mathbf{x}(\lambda_1)$ and $\mathbf{x}(\lambda_2)$ are optimal to MSDMCF($\bar{\lambda} + z$), and by Theorem 2, $\lambda_1 = \lambda_2$. $\square$

The assumption that $\mathbf{V}$ is positive definite (and not merely positive semidefinite) is necessary to guarantee the existence of a root. As a counterexample, consider a network with two nodes, connected by two parallel links: link 1 has mean cost 0 and standard deviation 1, whereas link 2 has a mean cost of 1 and a standard deviation of 0; the total demand is 1, and $\bar{\lambda} = 2$. The optimal solution to MSDMCF(2) is to assign all flow to link 2, and zero to link 1. However, this solution is not optimal to any mean-variance problem; for any $\lambda \geq 0$, the optimal solution to MVMCF(λ) places strictly positive flow on link 1.

3.1 Finding an initial λ

A simple way to find an interval containing this root is as follows: determine a finite upper endpoint by doubling an initial guess λ until $f(\lambda) \geq 0$, and $f(\lambda)$ is negative for $\lambda = 0$, so set the lower endpoint of the interval to 0.

However, more refined initial bounds can substantially improve algorithm performance. From Lemma 1, we know that $V(0) \geq V(\lambda)$ for all non-negative λ , and can be found efficiently by solving a linear minimum cost flow problem with the mean costs. Therefore, any λ with

$$\lambda \leq \frac{\bar{\lambda}}{2\sqrt{V(0)}}$$

also satisfies

$$\lambda \leq \frac{\bar{\lambda}}{2\sqrt{V(\bar{\lambda})}}.$$

Hence, we can set the lower bound for the interval that includes the root to $\lambda_{low} = \bar{\lambda}/(2\sqrt{V(0)})$.

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 $V(\lambda) > 1$ then $f(\lambda) > 0$. By changing units one can always satisfy the condition $V(\lambda) > 1$, 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 performs better. Specifically, we calculate $V(\infty)$ by solving a quadratic minimum cost flow problem with the variance term alone. Then, any λ with

$$\lambda \geq \frac{\bar{\lambda}}{2\sqrt{V(\infty)}}$$

also satisfies

$$\lambda \geq \frac{\bar{\lambda}}{2\sqrt{V(\bar{\lambda})}}.$$

Hence, we can set the upper bound for the interval that includes the root to $\lambda_{high} = \bar{\lambda}/(2\sqrt{V(\infty)})$

In Section 5, we show empirically that this procedure dramatically reduces the running time of the algorithm.

3.2 Bisection

A straightforward method to find the root is bisection, where a given initial interval is successively halved until the root is found to the desired precision. Pseudocode for the method is given in Figure 1. Since f has a unique root, and the above procedure guarantees an initial interval containing the root, convergence to the correct solution is assured.

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

$\tilde{\mathbf{x}}_h \leftarrow MVMCF(\lambda = 0), \tilde{\mathbf{x}}_l \leftarrow MVMCF(\lambda = \infty);$

$V(0) \leftarrow \tilde{\mathbf{x}}_h^T \mathbf{V} \tilde{\mathbf{x}}_h, V(\infty) \leftarrow \tilde{\mathbf{x}}_l^T \mathbf{V} \tilde{\mathbf{x}}_l;$

$\lambda_{low} \leftarrow \bar{\lambda}/2\sqrt{V(0)}, \quad \lambda_{high} \leftarrow \bar{\lambda}/2\sqrt{V(\infty)};$

$Found \leftarrow False;$

while not $Found$ **do**

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

$\mathbf{x}(\lambda) \leftarrow \arg \min (MVMCF(\lambda)) ;$

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

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

$Found \leftarrow True;$

else

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

$\lambda_{high} \leftarrow \lambda;$

else

$\lambda_{low} \leftarrow \lambda;$

3.3 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 [6]. 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:

$$\lambda_{n+1} = \left[\lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)} \right]^+, \quad (16)$$

where $[\cdot]^+ = \max\{0, \cdot\}$ is the positive component of its argument. 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

$$f'(\lambda) = \frac{2\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda) + 2\lambda \mathbf{x}(\lambda)^T \mathbf{V} \boldsymbol{\xi}}{\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}}. \quad (17)$$

In this section, we identify the derivatives $\boldsymbol{\xi}$ using sensitivity analysis, using similar techniques as in Boyles [10] and Jafari & Boyles [27]. 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 .

Let $C_{ij}(x_{ij}) = E[c_{ij}]x_{ij} + \lambda \text{Var}[c_{ij}]x_{ij}^2$ represent the cost of arc (i, j) , and $C'_{ij}(x_{ij}) = E[c_{ij}] + 2\lambda \text{Var}[c_{ij}]x_{ij}$ its derivative with respect to x_{ij} . Then, using $\mathbf{p}$ to represent the dual variables for the flow conservation constraints, the Karush-Kuhn-Tucker conditions for

MVMCF(λ) require that

$$\begin{aligned}
C'_{ij}(x_{ij}) + p_i - p_j &\geq 0 & \forall (i, j) : x_{ij} = 0 \\
C'_{ij}(x_{ij}) + p_i - p_j &= 0 & \forall (i, j) : 0 < x_{ij} < u_{ij} \\
C'_{ij}(x_{ij}) + p_i - p_j &\leq 0 & \forall (i, j) : x_{ij} = u_{ij} \\
\sum_{(j,k) \in \mathcal{A}} x_{jk} - \sum_{(i,j) \in \mathcal{A}} x_{ij} &= b_j & \forall j \in \mathcal{N} \\
0 \leq x_{ij} &\leq u_{ij} & \forall (i, j) \in \mathcal{A}
\end{aligned}$$

hold at optimality. Let $\mathcal{J}^*$ represent the set of arcs with $C'_{ij}(x_{ij}) + p_i - p_j = 0$, and further partition $\mathcal{J}^*$ into sets $\mathcal{J}_+^*$, $\mathcal{J}_0^*$, and $\mathcal{J}_-^*$ according to whether $x_{ij} = 0$, $0 < x_{ij} < u_{ij}$, or $x_{ij} = u_{ij}$ at optimality, respectively. (The sets $\mathcal{J}_+^*$ and $\mathcal{J}_-^*$ are empty unless the optimal solution is degenerate.)

Let φ represent the marginal change in $\mathbf{p}$ when the weight parameter λ is perturbed, $\varphi = d\mathbf{p}/d\lambda$. Differentiating the KKT conditions with respect to λ , we have

$$\begin{aligned}
2Var[c_{ij}](x_{ij} + \lambda\xi_{ij}) + \varphi_i - \varphi_j &= 0 & \forall (i, j) \in \mathcal{J}^* \\
\sum_{(j,k) \in \mathcal{J}^*} \xi_{jk} - \sum_{(i,j) \in \mathcal{J}^*} \xi_{ij} &= 0 & \forall j \in \mathcal{N} \\
\xi_{ij} &\geq 0 & \forall (i, j) \in \mathcal{J}_+^* \\
\xi_{ij} &\text{ free} & \forall (i, j) \in \mathcal{J}_0^* \\
\xi_{ij} &\leq 0 & \forall (i, j) \in \mathcal{J}_-^* \\
\xi_{ij} &= 0 & \forall (i, j) \in \mathcal{A} \setminus \mathcal{J}^*
\end{aligned}$$

which show how the optimal $\mathbf{x}$ and $\mathbf{p}$ change with λ .

A solution $\boldsymbol{\xi}$ to this problem could be obtained by solving this set of linear equations and inequalities; indeed, in the typical case where the optimal solution is nondegenerate it is simply a linear system of equations that can be solved using standard techniques. Regardless of degeneracy, we can recognize this system as the optimality conditions of the following quadratic program:

$$\begin{aligned}
\min_{\boldsymbol{\xi}} \quad & 2 \sum_{(i,j) \in \mathcal{J}^*} \text{Var}[c_{ij}] x_{ij} \xi_{ij} + \lambda \sum_{(i,j) \in \mathcal{J}^*} \text{Var}[c_{ij}] \xi_{ij}^2 \\
\text{s.t.} \quad & \sum_{(j,k) \in \mathcal{J}^*} \xi_{jk} - \sum_{(i,j) \in \mathcal{J}^*} \xi_{ij} = 0 \quad \forall j \in \mathcal{N} \\
& \xi_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{J}_+^* \quad (\Delta(\lambda, \mathbf{x})) \\
& \xi_{ij} \text{ free} \quad \forall (i,j) \in \mathcal{J}_0^* \\
& \xi_{ij} \leq 0 \quad \forall (i,j) \in \mathcal{J}_-^* \\
& \xi_{ij} = 0 \quad \forall (i,j) \in \mathcal{A} \setminus \mathcal{J}^*
\end{aligned}$$

Note that $\mathbf{x}$ is a parameter in this formulation, the optimal solution of the $\text{MVMCF}(\lambda)$, and that $\boldsymbol{\xi}$ is the only decision variable. Furthermore, this optimization problem is very nearly an instance of MVMCF , restricted to the links in the set $\mathcal{J}^*$, without capacities, with different sign constraints, and with mean link costs replaced with $2\text{Var}[c_{ij}]x_{ij}$.

There are advantages to obtaining $\boldsymbol{\xi}$ by solving this MVMCF variant, rather than solving the linear system directly. Using existing algorithms for MVMCF exploits problem structure. Our experiments showed that such algorithms were faster and more numerically stable, and furthermore provide a natural way to identify solutions at a customizable level of precision (at early iterations, high-precision solutions for these derivatives are likely not necessary).

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, the behavior of this implicit function is difficult to analyze to determine whether quadratic convergence can be guaranteed. In the next subsection, we provide a fail-safe to alleviate the lack of convergence guarantee for the pure Newton algorithm. We also note that the method converged for all the test instances in our experiments, despite the lack of a convergence proof.

3.4 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. [53]. Specifically,

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

```

 $\tilde{\mathbf{x}}_h \leftarrow MVMCF(0);$ 
 $V(0) \leftarrow \tilde{\mathbf{x}}_h^T \mathbf{V} \tilde{\mathbf{x}}_h;$ 
 $\lambda \leftarrow \bar{\lambda}/2\sqrt{V(0)};$ 
while not Found do
     $\mathbf{x}(\lambda) \leftarrow \arg \min (MVMCF(\lambda));$ 
     $f(\lambda) \leftarrow 2\lambda\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)} - \bar{\lambda};$ 
    if  $|f(\lambda)| \leq TOL$  then
         $Found \leftarrow True;$ 
    else
         $\boldsymbol{\xi} \leftarrow \arg \min \Delta(\lambda, \mathbf{x});$ 
         $f'(\lambda) \leftarrow \frac{2\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda) + 2\lambda\mathbf{x}(\lambda)^T \mathbf{V} \boldsymbol{\xi}}{\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}};$ 
         $\lambda \leftarrow \left[ \lambda - \frac{f(\lambda)}{f'(\lambda)} \right]^+;$ 

```

we switch to a bisection step whenever the current Newton-Raphson step suggests a solution out of the bracket, or whenever the bracket size is not reducing rapidly enough.

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 is needed for “rapidly enough.” In our implementation, we check whether $|f(\lambda)|$ is smaller than the $|f(\lambda)|$ in the previous iteration, if not, we apply bisection. This approach prevents possible divergent behaviors in the pure NR algorithm. The pseudocode of the algorithm is provided in Figure 3.

4 GENERALIZATION TO NON-SEPARABLE PARAMETRIC CONVEX COST PROBLEMS

This section discusses the applicability of the proposed methods to certain generalizations. The mathematical technique of solving MSDMCF by solving a sequence of MVMCF prob-

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

```

 $\tilde{\mathbf{x}}_h \leftarrow \text{MVMCF}(0);$ 
 $V(0) \leftarrow \tilde{\mathbf{x}}_h^T \mathbf{V} \tilde{\mathbf{x}}_h;$ 
 $\lambda_{low} \leftarrow \bar{\lambda}/2\sqrt{V(0)};$ 
while not Found do
     $\mathbf{x}(\lambda) \leftarrow \arg \min (\text{MVMCF}(\lambda));$ 
     $f(\lambda) \leftarrow 2\lambda\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)} - \bar{\lambda};$ 
    if  $|f(\lambda)| \leq TOL$  then
         $\text{Found} \leftarrow \text{True}$ 
    else
        if  $|f(\lambda)| \leq |f(\lambda_{prev})|$  then
             $\boldsymbol{\xi} \leftarrow \arg \min \Delta(\lambda, \mathbf{x});$ 
             $f'(\lambda) \leftarrow \frac{2\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda) + 2\lambda\mathbf{x}(\lambda)^T \mathbf{V} \boldsymbol{\xi}}{\sqrt{\mathbf{x}(\lambda)^T \mathbf{V} \mathbf{x}(\lambda)}};$ 
             $f(\lambda_{prev}) \leftarrow f(\lambda);$ 
             $\lambda_{prev} \leftarrow \lambda;$ 
             $\lambda \leftarrow \lambda - \frac{f(\lambda)}{f'(\lambda)};$ 
            if  $\lambda_{low} < \lambda < \lambda_{high}$  then
                if  $f(\lambda) > 0$  then
                     $\lambda_{high} \leftarrow \lambda;$ 
                else
                     $\lambda_{low} \leftarrow \lambda$ 
            else
                 $\lambda_{prev}$  and perform Bisection step
        else
             $\lambda_{prev}$  and perform Bisection step

```

lems can be applied to a broader class of optimization problems. The question is whether the resulting algorithms are efficient, which depends critically on whether the transformed problems are substantially simpler to solve than the original one and, in the case of the Newton methods, whether derivatives are easily available. Exploring this question thoroughly is beyond the scope of the current paper, but we provide some initial discussion here.

The BSC method is applicable not only for MCF problems, but for the general case of non-separable convex cost problems

$$\min_{\mathbf{x} \in \mathcal{X}} \mu(\mathbf{x}) + \bar{\lambda}g(v(\mathbf{x})) \quad (18)$$

where $\mathcal{X}$ is a bounded linear constraint system, μ and v are separable and differentiable functions of $\mathbf{x}$ with $v(\mathbf{x}) \geq 0$, g is a strictly monotone, increasing and differentiable function, and lastly the composition $g \circ v$ is convex.

We can define an associated convex separable problem:

$$\min_{\mathbf{x} \in \mathcal{X}} \mu(\mathbf{x}) + \lambda v(\mathbf{x}) \quad (19)$$

By following the same procedure as in the proof of Theorem 1, we can find a relation between the two problems.

$$\lambda = \bar{\lambda} \left. \frac{\partial g}{\partial v} \right|_{v=v(\mathbf{x}(\lambda))} . \quad (20)$$

Note that the differentiability of both g and v is necessary for Theorem 1 to follow. Moreover, as $g \circ v$ is convex by assumption, the KKT necessary conditions are also sufficient for optimality of (18) and therefore the optimal solution for (19) with λ satisfying (20) is also optimal for (18). By similar arguments made earlier, one can show that the function $f(\lambda)$ is continuous. Moreover, it takes values of opposite signs when evaluated at the endpoints of the domain. By assumption, g is monotonically increasing, and therefore the derivative is positive for any λ in the domain. Then, $f(\lambda)$ takes a negative 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. If, in addition, the

function g is twice differentiable and we have an MCF problem, one can conduct sensitivity analysis and apply Newton-Raphson.

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 into the class of functions for g that satisfies conditions needed by this framework. We refer the reader to an extensive survey of benchmark functions [28] 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, a standard MCF problem with the addition of a non-linearly-valued budget constraint, such as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mu(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \\ & g(v(\mathbf{x})) \leq B, \end{aligned} \tag{21}$$

can be cast into the form

$$\begin{aligned} \min_{\mathbf{x}} \max_{\bar{\lambda} \geq 0} \quad & \mu(\mathbf{x}) + \bar{\lambda}(g(v(\mathbf{x})) - B) \\ \text{s.t.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{aligned} \tag{22}$$

by Langrangianizing the budget constraints. For a given $\bar{\lambda}$, we then have an outer minimization problem which is of the form we consider in this section. Note that Langrangianizing the budget constraints will lead to a max min problem. Strong duality holds if there exists a feasible solution $\mathbf{x}$ for which $g(v(\mathbf{x})) < B$. As such point would be an interior point and so Slater's condition will then hold. In this case, one can swap the max min problem into a min max problem and end up with the optimization problem in (22).

5 COMPUTATIONAL EXPERIMENTS

In this section, we assess the performance of the proposed algorithms, using CPLEX to solve the original MSDMCF problem directly, and to solve the MVMCF subproblems in

our algorithms. We chose to use CPLEX for both cases to facilitate comparison, and to demonstrate that the MVMCF subproblems are indeed easier to solve without introducing confounding factors of different implementations, compiler and architecture optimizations, and so forth. 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. The computational experiments are performed on a quad-core 2.8 GHz computer with 16 GB RAM. The code used for the computational experiments and analysis is provided at <https://github.com/cangokalp/mean-std>.

5.1 Benchmark networks

The performance of the methods are evaluated on the networks generated with the well-known random generator NETGEN [32]. We use the benchmark suite created in [31], which was designed to compare linear MCF solution methods.

In the NETGEN problem families, the arc costs and capacities are uniformly drawn from $[1, 10^4]$ and $[1, 10^3]$, respectively. There are approximately $\sqrt{n}$ supply and demand nodes, and the average supply per supply node is set to 10^3 .

There are four problem families created with the 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.

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

TABLE 1: Initialization procedure benefits - on a network with 4096 nodes and average degree of 64.

We use the arc costs in the instances as the mean arc costs $E[c_{ij}]$. We sample a coefficient of variation COV_{ij} for each link, drawn uniformly from $[0.15, 0.3]$, and thus set the variance as $Var_{ij} = (COV_{ij}E[c_{ij}])^2$. This interval for COV_{ij} represents typical variation in transportation networks [4].

5.2 Benefits of customized initialization

Table 1 illustrates the benefits of using custom bounds found with the procedure described in Subsection 3.1. It compares the number of iterations needed, 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 helps the algorithms to start very close to λ^* , and therefore iteration numbers and running times are much lower.

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 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 criteria of $TOL = 10^{-8}$. The MVMCF subproblems are solved using the CPLEX solver. All comparisons 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 0.01% “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.

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 degree per node in each row. NR method outperforms the other methods in 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 by 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 families 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, the 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, the 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 8 presents the convergence behavior of the algorithms on the NETGEN-LO-SR family on a representative problem instance with 2^{14} nodes. The BSC 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 3.1. 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.

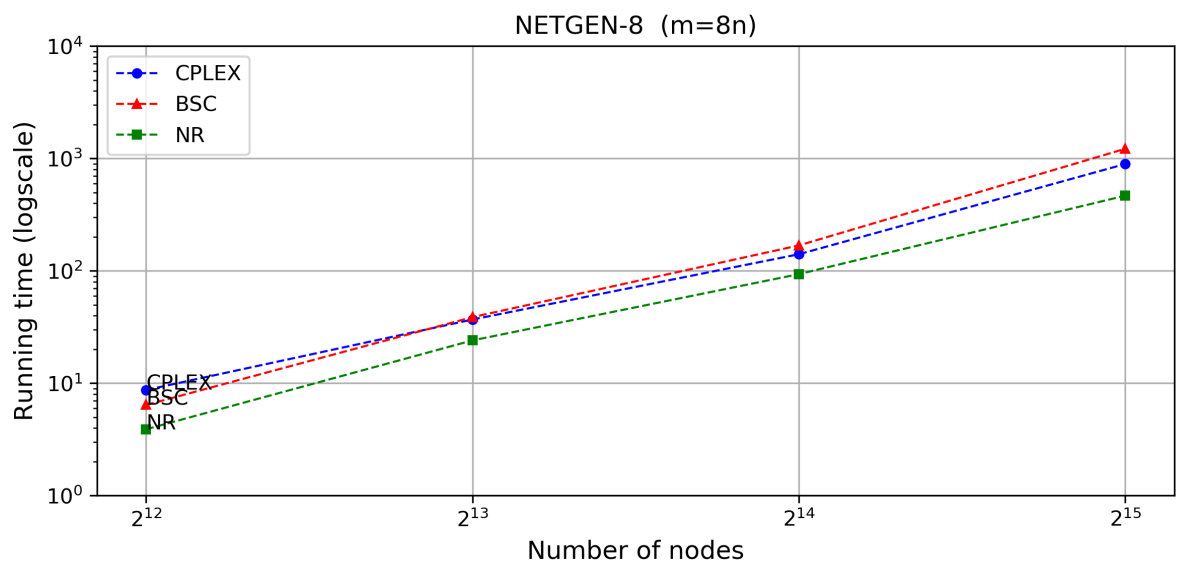


FIGURE 4: Comparison of the algorithms on NETGEN-8 families (logarithmic scale).

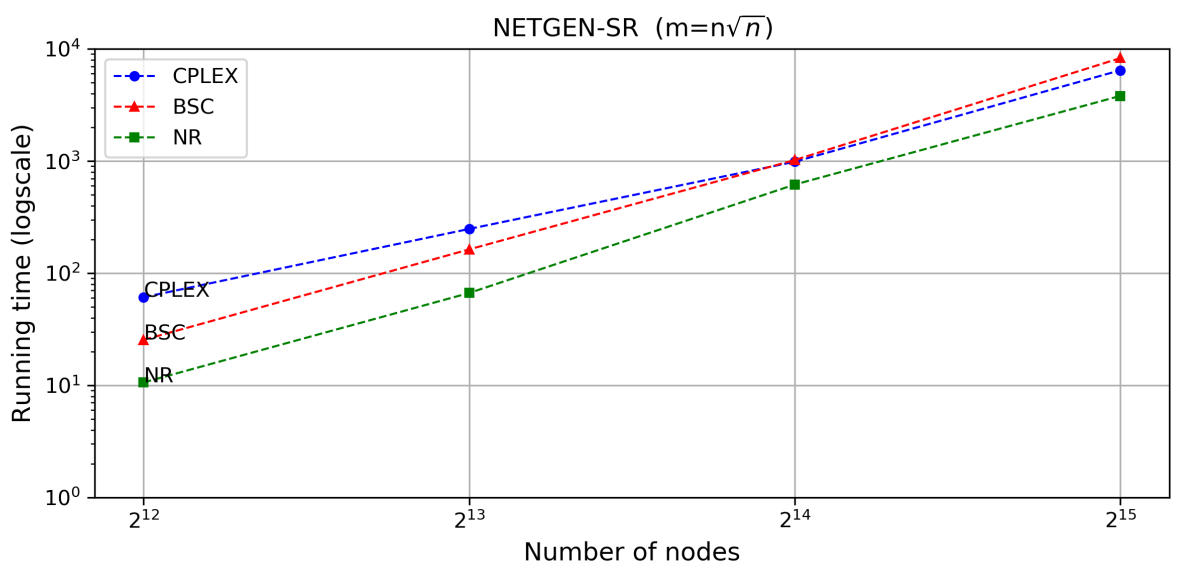


FIGURE 5: Comparison of the algorithms on NETGEN-SR families (logarithmic scale).

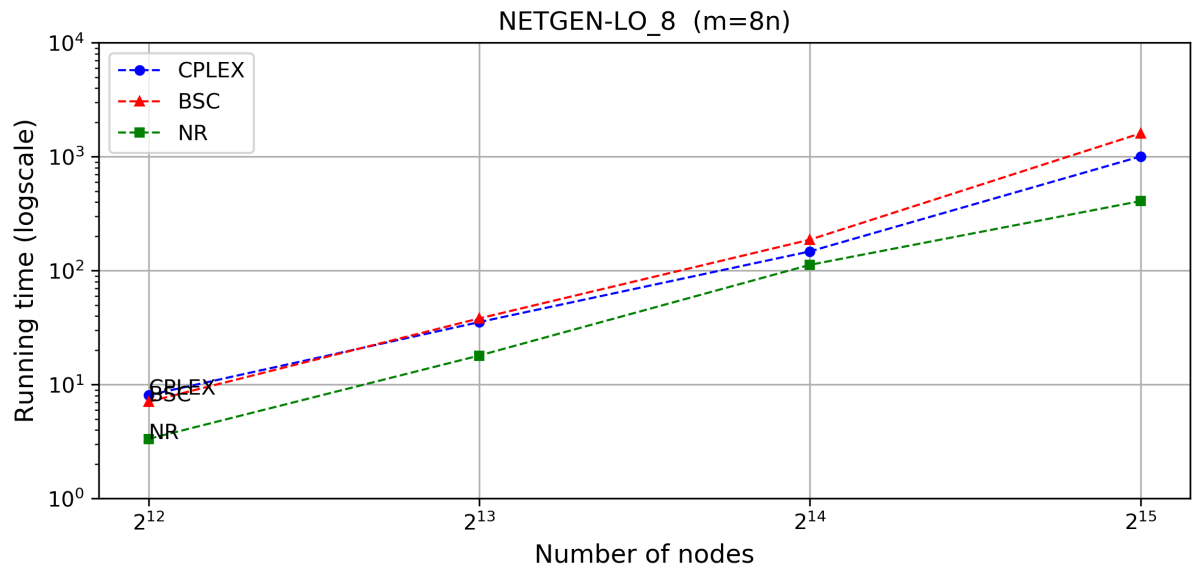


FIGURE 6: Comparison of the algorithms on NETGEN-LO-8 families (logarithmic scale).

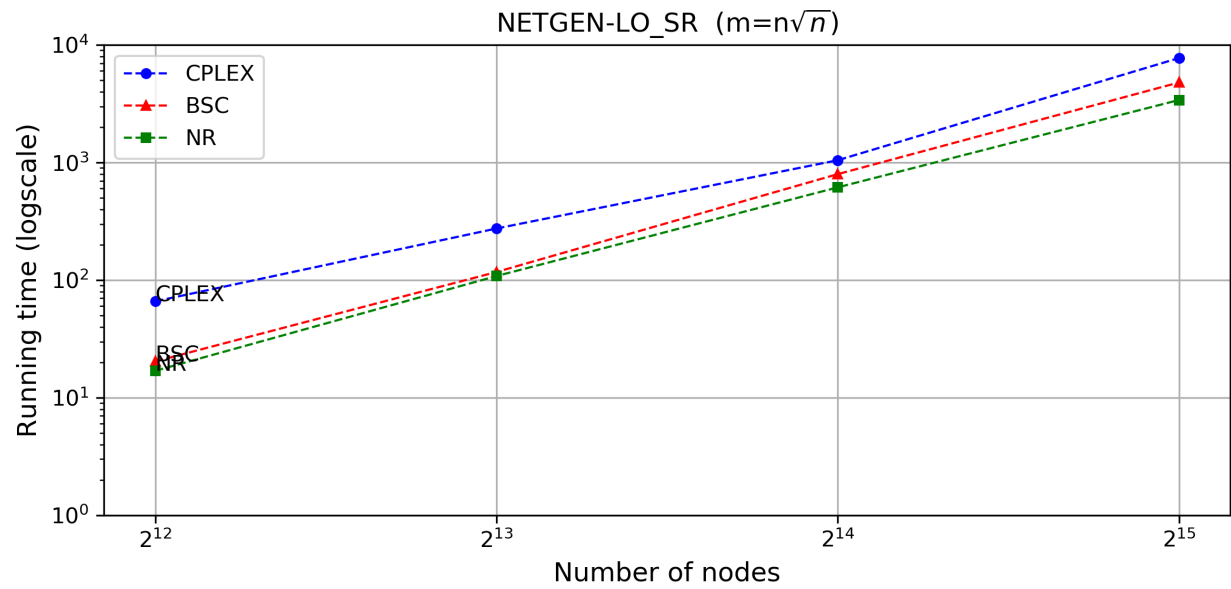


FIGURE 7: Comparison of the algorithms on NETGEN-LO-SR families (logarithmic scale).

Size		Time (s)			Avg. Iteration #	
n	deg	CPLEX	BSC	NR	NR	BSC
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.

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

TABLE 3: Comparison on NETGEN-SR instances.

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

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

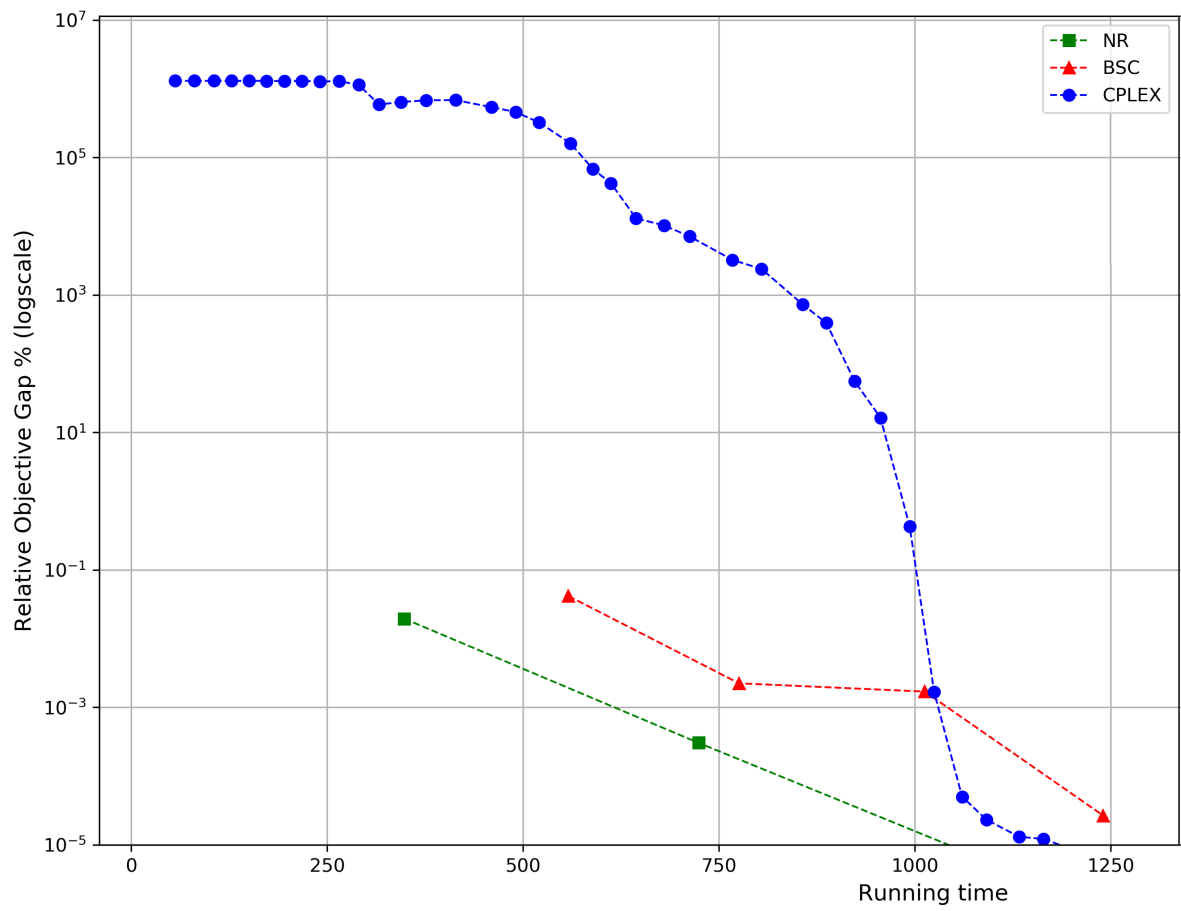


FIGURE 8: Convergence behavior.

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

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

Gap	Method	NETGEN-8				NETGEN-SR				NETGEN-LO-8				NETGEN-LO-SR			
		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.

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 tuning was done).

5.4 Sensitivity to reliability

This subsection emphasizes the need for modeling reliability by showing the difference in solutions between our model and a deterministic model. Additionally, we also investigate how the performance of the algorithms changes with respect to the changes in the reliability parameter $\bar{\lambda}$. In our experiments, standard deviations were generated uniformly from $[0.15E[c_{ij}], 0.3E[c_{ij}]]$, based on typical variation in transportation networks [4]; however in

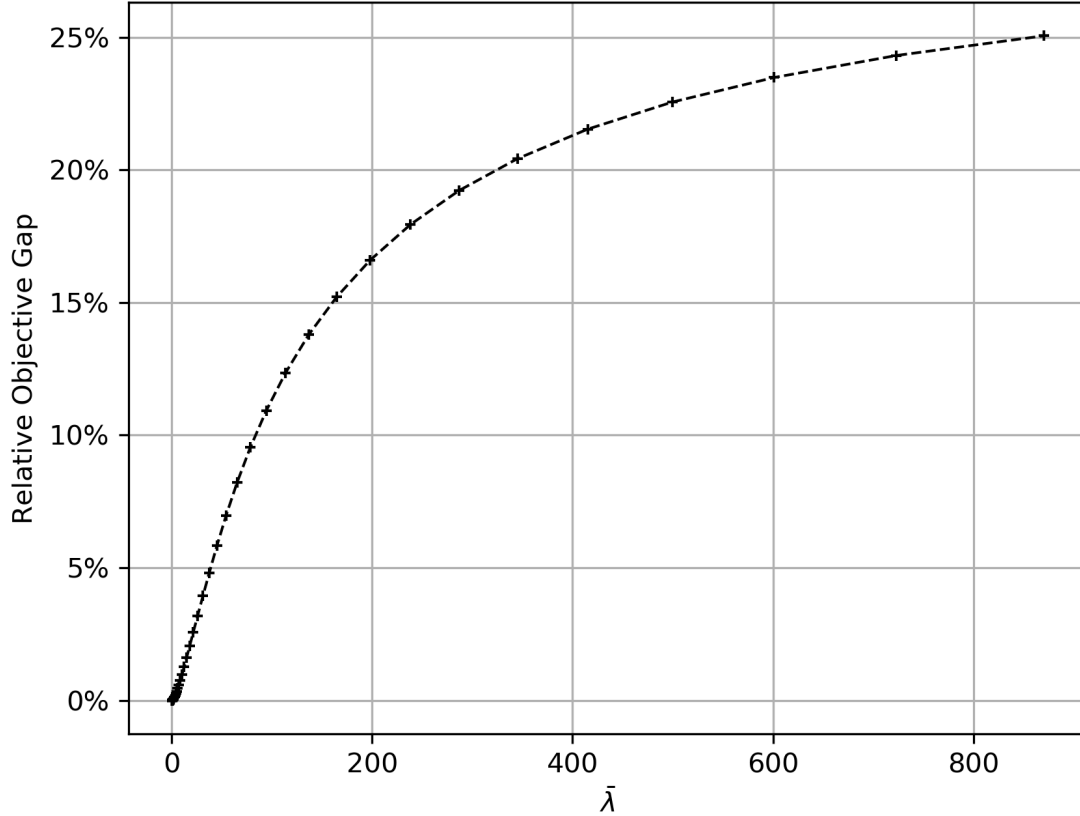


FIGURE 9: NETGEN-8 with $n = 2^{10}$.

other types of networks this problem parameter might be very different. To capture the possible effects of higher or lower variation for the arc costs, in the set of experiments we perform in this subsection we allow $\bar{\lambda}$ to range from 0.1 to 1000 and investigate the sensitivity of the problem metrics.

In terms of modeling, Figure 9 plots the percentage relative gap between the objective value of a deterministic solution obtained by minimizing 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 reliability becomes more and more important to the decision-maker, the performance of the deterministic model deteriorates. In such situations, where reliability is important, using a mean-standard deviation model may outweigh the

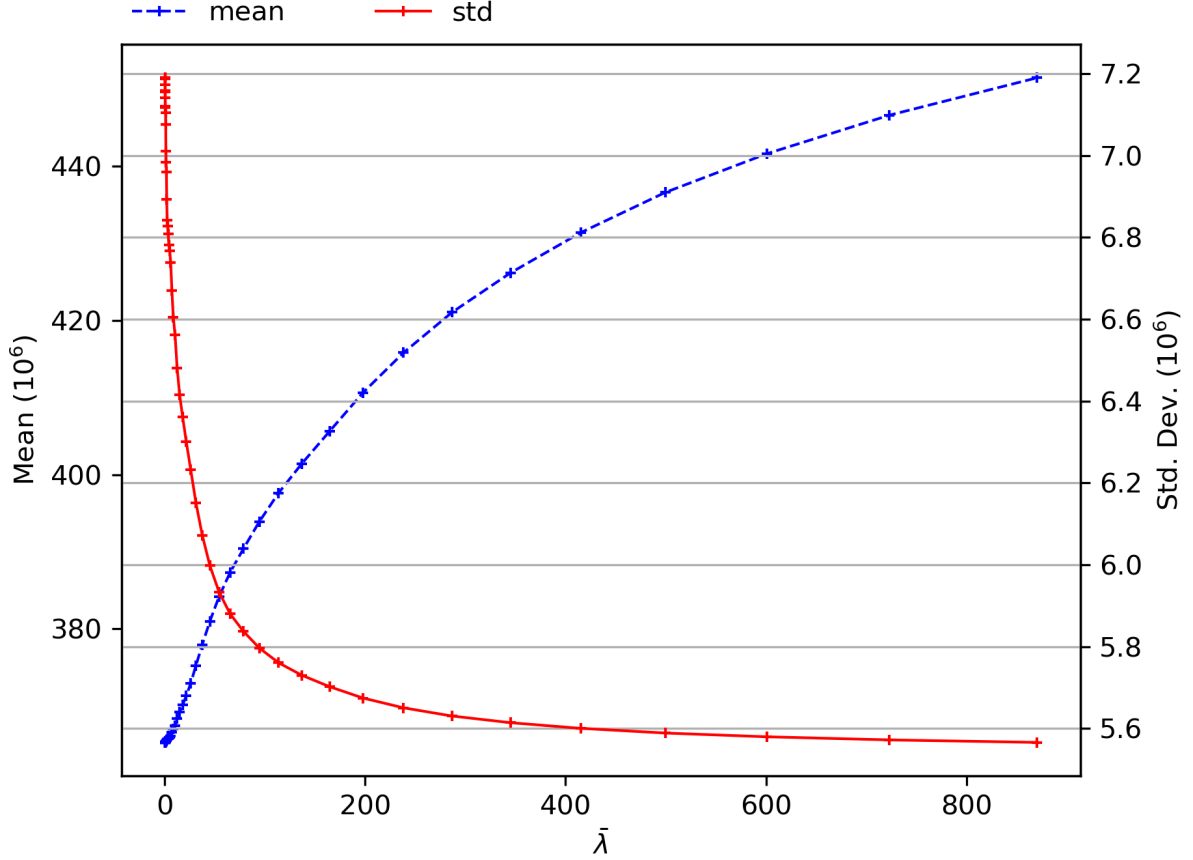


FIGURE 10: Criteria trade-off.

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 plot the performance of the algorithms with respect to different reliability parameters for each of the graph families in the benchmark suite. Among all the methods, the performance of the BSC method is the most robust against the variation in the $\bar{\lambda}$ parameter. On the other hand, the performance of the NR method is more sensitive to the changes in $\bar{\lambda}$. This is mainly due to the fact that a single iteration in NR takes longer than a single iteration in BSC. When $\bar{\lambda}$ increases, both of the methods require an additional iteration, although

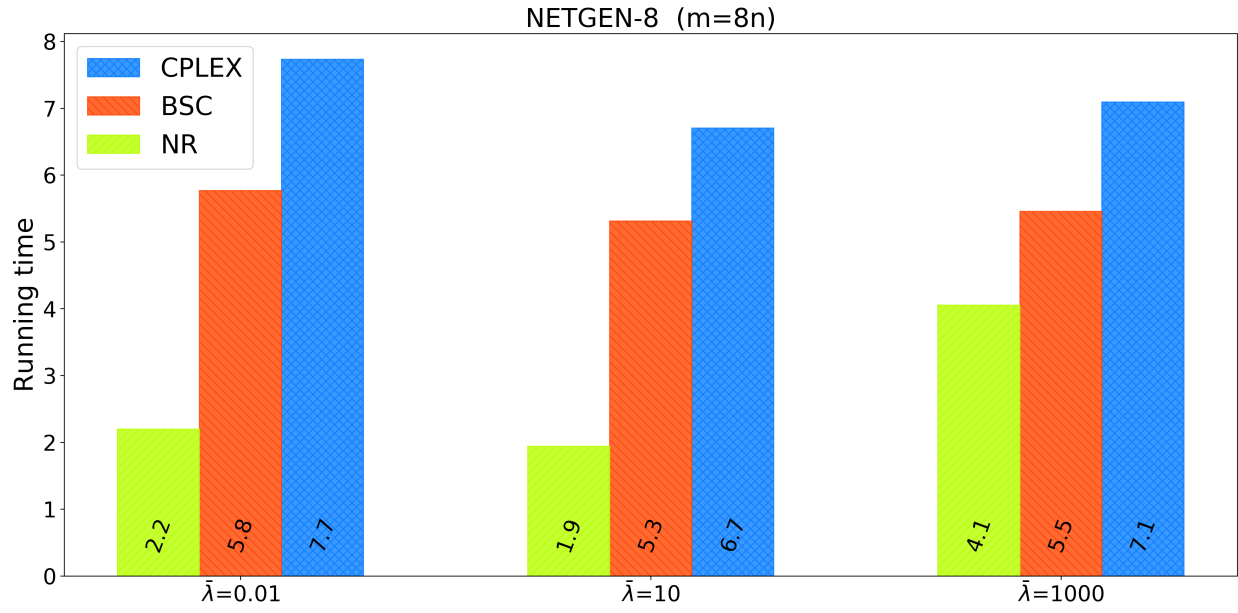


FIGURE 11: Sensitivity to $\bar{\lambda}$ on NETGEN-8 instances.

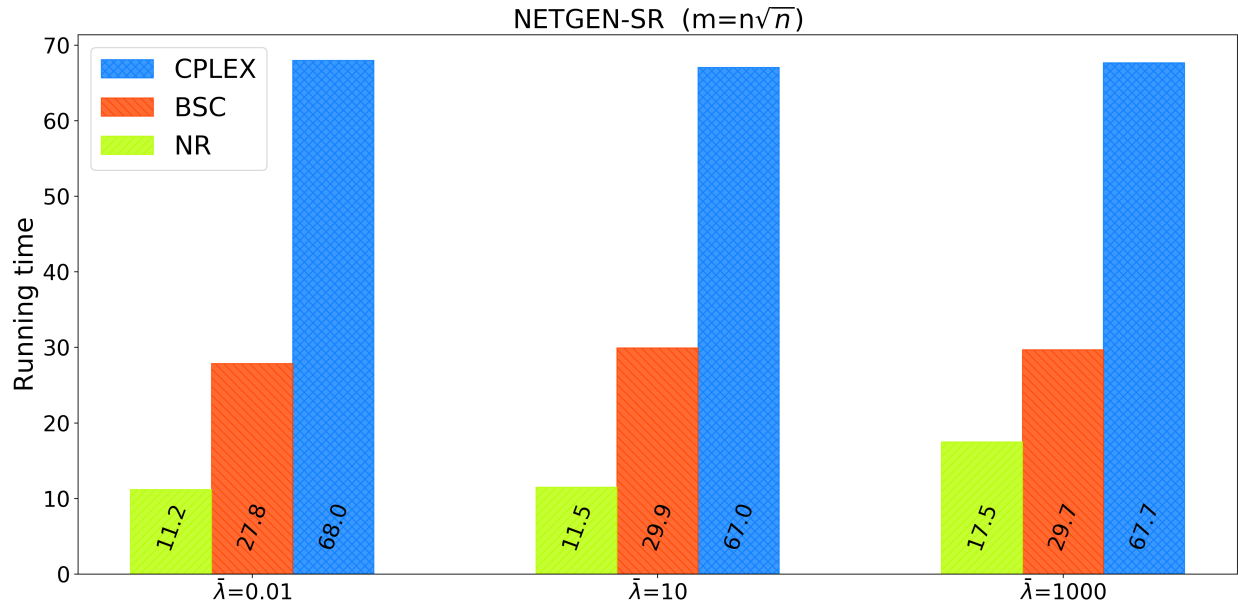


FIGURE 12: Sensitivity to $\bar{\lambda}$ on NETGEN-SR instances.

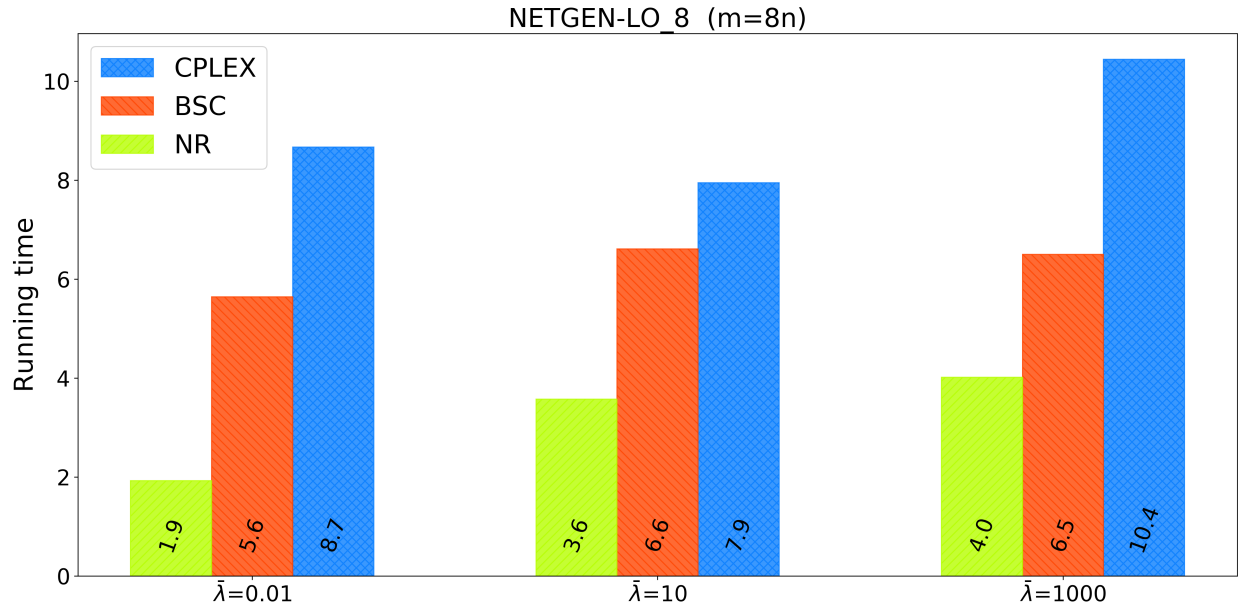


FIGURE 13: Sensitivity to $\bar{\lambda}$ on NETGEN-LO-8 instances.

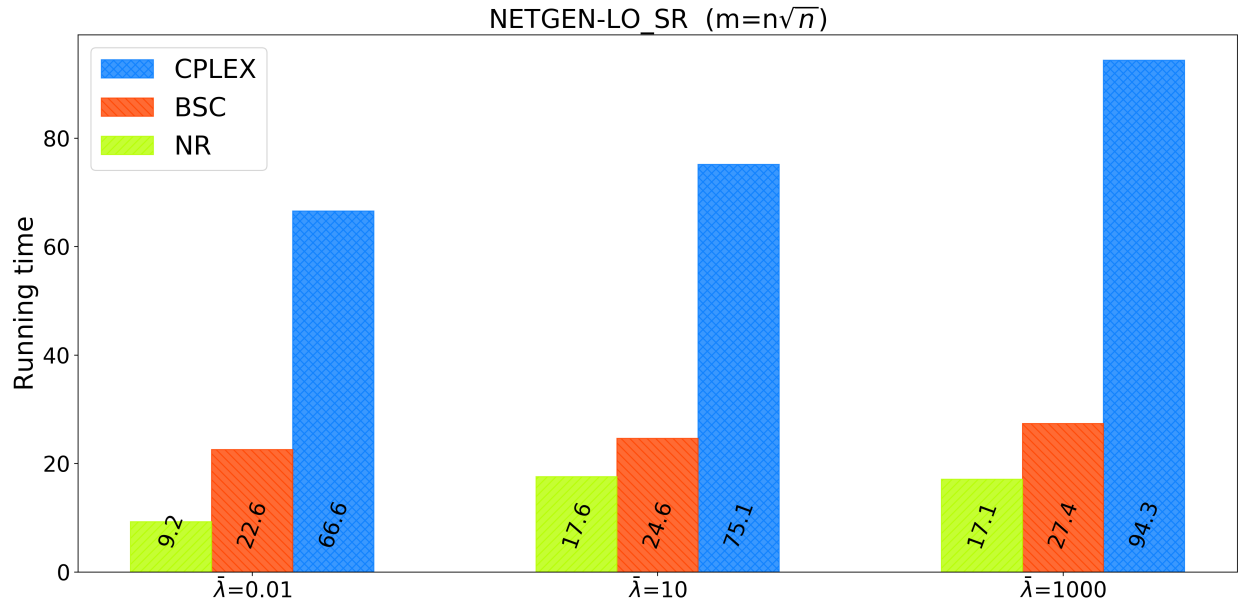


FIGURE 14: Sensitivity to $\bar{\lambda}$ on NETGEN-LO-SR instances.

the increase in solution time for NR is larger than the increase in solution time for BSC for an additional iteration. Furthermore, as $\bar{\lambda}$ increases, the non-separable part of the objective gets more weight. As a result, the difficulty for solving the problem directly increases, which is observed on harder problem instances created by NETGEN-LO-SR family. Lastly, we note that even for $\bar{\lambda} = 1000$ – where even the choice of $\bar{\lambda} > 10$ would be probably unlikely as a weight parameter – the performance of NR is still better compared to others.

6 CONCLUSION

This paper described three solution algorithms for the mean-standard deviation minimum cost flow problem, based on solving a sequence of easier MVMCF problems. The algorithms differ in the method they use to identify the weight parameter in the MVMCF subproblems. We also provide a procedure to find tighter upper and lower bounds for the root-finding methods, which improves the performance significantly. Among 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 compared the running times of the algorithms to achieve a gap level of 0.01%. The NR method outperformed CPLEX and BSC on every problem instance. In contrast, BSC outperformed CPLEX for small instances of dense network families, while performing competitively or worse for larger instances. The NR and BSC methods achieved very good solutions quickly. 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. In future work, it would be interesting to see if the tolerance used in finding a root of f can be directly linked

to the optimality gap in the resulting MSDMCF solution.

The run time 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 [45, 11, 41, 29] specialized for separable convex MCF problems could reduce run time 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 an analysis on early stopping for early iterations in the proposed methods. The framework can be 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 to explore whether the application of this method to the cases of correlated link costs, or to other optimization problems of the form in Section 4, are efficient.

7 ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under grants CMMI-1826320/1826337, CMMI-1562109/1562291, and CMMI-1254921, and the Data-Supported Transportation Operations and Planning University Transportation Center.

APPENDIX

It turns out that for both MVMCF and MSDMCF, the Pareto frontier coincides with its lower convex envelope, and that therefore the methods described in this paper can find any non-dominated solution with respect to mean and standard deviation (or variance). The key idea is that the Pareto frontier is a convex function; given any two solutions $\mathbf{x}_1$ and $\mathbf{x}_2$, and any $\alpha \in [0, 1]$, the mean cost of $\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$ is a linear function of α , whereas the variance of $\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$ is a convex function of α . Therefore, the type of situation shown in Figure 15 (where there are Pareto-optimal solutions not on the lower convex envelope) cannot occur; the dashed line connecting points $\mathbf{x}_1$ and $\mathbf{x}_2$ traces out the mean and variance of $\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$, and so a point like $\mathbf{y}$ cannot be Pareto optimal.

A more rigorous justification is as follows. First consider MVMCF. For convenience, let

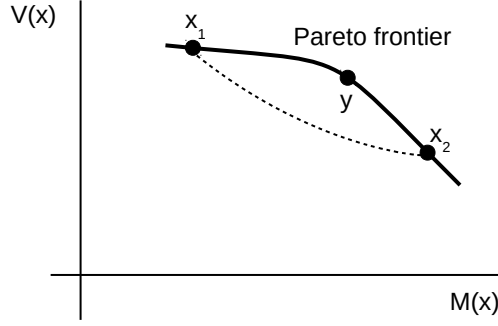


FIGURE 15: The Pareto frontier only contains points on its lower convex envelope.

$M(\mathbf{x})$ and $V(\mathbf{x})$ represent the mean and variance of a feasible solution $\mathbf{x}$, $M(\mathbf{x}) = \boldsymbol{\mu}^T \mathbf{x}$ and $V(\mathbf{x}) = \mathbf{x}^T \mathbf{V} \mathbf{x}$. The function M is clearly linear in $\mathbf{x}$, and as $\mathbf{V}$ is positive definite, V is a convex function.¹ Let $\underline{M}$ and $\overline{M}$ respectively denote the minimum and maximum values of $M(\mathbf{x})$ across all feasible $\mathbf{x}$; these values exist since M is continuous and the feasible set $\mathcal{X}$ is compact. Finally, for $m \in [\underline{M}, \overline{M}]$, let $V^*(m)$ denote the least variance among all solutions with mean cost m , that is, $V^*(m) = \min\{V(\mathbf{x}) : M(\mathbf{x}) = m, \mathbf{x} \in \mathcal{X}\}$. This function is well-defined since it is the minimum of a continuous function over a compact set, and it characterizes the set of solutions along the Pareto frontier.

Now, $V^*(m)$ is a convex function. To show this, choose $m_1, m_2 \in [\underline{M}, \overline{M}]$, let $\mathbf{x}_1$ and $\mathbf{x}_2$ be feasible solutions with $V(\mathbf{x}_1) = V^*(m_1)$ and $V(\mathbf{x}_2) = V^*(m_2)$, and consider $\alpha \in [0, 1]$. The solution $\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2$ is also feasible and has mean $\alpha m_1 + (1 - \alpha) m_2$, and therefore

$$\begin{aligned}
 V^*(\alpha m_1 + (1 - \alpha) m_2) &\leq V(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2) && \text{(by optimality)} \\
 &\leq \alpha V(\mathbf{x}_1) + (1 - \alpha) V(\mathbf{x}_2) && \text{(by convexity of } V(\mathbf{x}) \text{)} \\
 &= \alpha V^*(m_1) + (1 - \alpha) V^*(m_2).
 \end{aligned}$$

It follows that the Pareto frontier V^* is piecewise differentiable, with a subdifferential $\partial V^*(m)$ at each $m \in [\underline{M}, \overline{M}]$.

¹This is a slight abuse of notation, since Section 3.2 uses $M(\lambda)$ and $V(\lambda)$ to refer to the mean and variance of the optimal solution to $\text{MVNFP}(\lambda)$, but we feel the mnemonic advantage of using M and V to refer to means and variances outweighs any possible confusion.

Finally, since every line through $V^*(m)$ with a slope in $\partial V^*(m)$ is a supporting tangent, a solution $\mathbf{x}$ is optimal to $MVNFP(\lambda)$ if and only if $V(\mathbf{x}) = V^*(M(\mathbf{x}))$ (that is, $\mathbf{x}$ is Pareto-efficient) and $-1/\lambda \in \partial V^*(M(\mathbf{x}))$. For if $V(\mathbf{x}) = V^*(M(\mathbf{x}))$ and $-1/\lambda \in \partial V^*(M(\mathbf{x}))$, given any other feasible solution $\mathbf{y}$ we have

$$M(\mathbf{y}) + \lambda V(\mathbf{y}) \geq M(\mathbf{y}) + \lambda V^*(M(\mathbf{y})).$$

By convexity,

$$V^*(M(\mathbf{y})) \geq V^*(M(\mathbf{x})) - \frac{1}{\lambda}(M(\mathbf{y}) - M(\mathbf{x})). \quad (23)$$

Combining these gives

$$M(\mathbf{y}) + \lambda V(\mathbf{y}) \geq M(\mathbf{x}) + \lambda V(\mathbf{x}), \quad (24)$$

establishing the optimality of $\mathbf{x}$. In the other direction, if $\mathbf{x}$ is optimal to $MVNFP(\lambda)$, then clearly $V(\mathbf{x}) = V^*(M(\mathbf{x}))$. Inequality (24) is also valid for all feasible $\mathbf{y}$, and in particular those $\mathbf{y}$ for which $V(\mathbf{y}) = V^*(M(\mathbf{y}))$. Therefore (23) holds, and $-1/\lambda \in \partial V^*(M(\mathbf{x}))$.

The same arguments hold for MSDMCF since the standard deviation is also convex in $\mathbf{x}$ (as established in Section 2).

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] Alper Atamtürk and Muhong Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.
- [3] Lawrence M Ausubel and Raymond J Deneckere. A generalized theorem of the maximum. *Economic Theory*, 3(1), 1993.
- [4] John Bates. Challenges and accomplishments of modeling impacts of policy initiatives. *In Association for European Transport and Contributors*, 2008.
- [5] Aharon Ben-Tal, Laurent E Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28. Princeton University Press, 2009.
- [6] Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.
- [7] Dimitris Bertsimas, Ebrahim Nasrabadi, and Sebastian Stiller. Robust and adaptive network flows. *Operations Research*, 61(5):1218–1242, 2013.
- [8] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical programming*, 98(1–3):49–71, 2003.
- [9] John R Birge and James K Ho. Optimal flows in stochastic dynamic networks with congestion. *Operations Research*, 41(1):203–216, 1993.
- [10] Stephen D Boyles. Bush-based sensitivity analysis for approximating subnetwork diversion. *Transportation Research Part B*, 46:139–155, 2012.
- [11] 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.

- [12] 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.
- [13] Bi Y Chen, William H K Lam, Agachai Sumalee, Qingquan Li, and Mei L Tam. Reliable shortest path problems in stochastic time-dependent networks. *Journal of Intelligent Transportation Systems*, 18(2):177–189, 2014.
- [14] Louis Chen, Will Ma, James Orlin, and David Simchi-levi. *Distributionally Robust Max Flows*, pages 81–90. 01 2020.
- [15] Peng Chen and Yu M Nie. Bicriterion shortest path problem with a general nonadditive cost. *Transportation Research Part B: Methodological*, 57:419–435, 2013.
- [16] Augusto Eusébio and José R Figueira. Finding non-dominated solutions in bi-objective integer network flow problems. *Computers & Operations Research*, 36(9):2554–2564, 2009.
- [17] Augusto Eusébio, José R Figueira, and Matthias Ehrgott. On finding representative non-dominated points for bi-objective integer network flow problems. *Computers & Operations Research*, 48:1–10, 2014.
- [18] Yueyue Fan, Robert E Kalaba, and James E Moore. Arriving on time. *Journal of Optimization Theory and Applications*, 127(3):497–513, 2005.
- [19] Yueyue Fan and Yu M Nie. Optimal routing for maximizing the travel time reliability. *Networks and Spatial Economics*, 6(3-4):333–344, 2006.
- [20] Mogens Fosgerau and Leonid Engelson. The value of travel time variance. *Transportation Research Part B: Methodological*, 45(1):1–8, 2011.
- [21] Mogens Fosgerau and Anders Karlström. The value of reliability. *Transportation Research Part B: Methodological*, 44(1):38–49, 2010.

- [22] Steven A Gabriel and David Bernstein. The traffic equilibrium problem with nonadditive path costs. *Transportation Science*, 31(4):337–348, 1997.
- [23] 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.
- [24] 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.
- [25] Dorit Hochbaum. Complexity and algorithms for nonlinear optimization problems. *Annals of Operations Research*, 153(1):257–296, September 2007.
- [26] Kevin R Hutson and Douglas R Shier. Extended dominance and a stochastic shortest path problem. *Computers & Operations Research*, 36(2):584–596, 2009.
- [27] Ehsan Jafari and Stephen D Boyles. Improved bush-based methods for network contraction. *Transportation Research Part B*, 83:298–313, 2016.
- [28] Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimization problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 08 2013.
- [29] P V Kamesam and R R Meyer. *Multipoint methods for separable nonlinear networks*, pages 185–205. Springer Berlin Heidelberg, Berlin, Heidelberg, 1984.
- [30] 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.
- [31] Zoltán Király and Péter Kovács. Efficient implementations of minimum-cost flow algorithms. *Acta Universitatis Sapientiae, Informatica*, 4, 2012.

- [32] 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.
- [33] 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.
- [34] Haijune Lee and P Simin Pulat. Bicriteria network flow problems: Continuous case. *European Journal of Operational Research*, 51(1):119–126, 1991.
- [35] J Shung Lin, Chin C Jane, and John Yuan. On reliability evaluation of a capacitated-flow network in terms of minimal pathsets. *Networks*, 25(3):131–138, 1995.
- [36] Jsen S Lin. Reliability evaluation of capacitated-flow networks with budget constraints. *IIE Transactions*, 30(12):1175–1180, 1998.
- [37] Yi K Lin. A simple algorithm for reliability evaluation of a stochastic-flow network with node failure. *Computers & Operations Research*, 28(13):1277–1285, 2001.
- [38] Yi K 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.
- [39] Yi K Lin. On a multicommodity stochastic-flow network with unreliable nodes subject to budget constraint. *European Journal of Operational Research*, 176(1):347–360, 2007.
- [40] Yi K 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.
- [41] R R Meyer. Two-segment separable programming. *Management Science*, 25(4):385–395, 1979.

- [42] 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.
- [43] 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.
- [44] 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.
- [45] V A Nguyen and Y.-P Tan. Minimum convex cost flow problem. In *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, volume 2, pages 1248–1252, 2003.
- [46] 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.
- [47] Yu M Nie. Multi-class percentile user equilibrium with flow-dependent stochasticity. *Transportation Research Part B: Methodological*, 45(10):1641–1659, 2011.
- [48] Yu M Nie and Xing Wu. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological*, 43(6):597–613, 2009.
- [49] Yu M 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.
- [50] 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.

- [51] Fernando Ordóñez and Nicolás E Stier-Moses. Wardrop equilibria with risk-averse users. *Transportation Science*, 44(1):63–86, 2010.
- [52] 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.
- [53] 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.
- [54] P Simin Pulat, Fenghueih Huarng, and Haijune Lee. Efficient solutions for the bicriteria network flow problem. *Computers & Operations Research*, 19(7):649–655, 1992.
- [55] 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.
- [56] 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.
- [57] Antonio Sedeño-Noda and C González-Martín. The biobjective minimum cost flow problem. *European Journal of Operational Research*, 124(3):591–600, 2000.
- [58] 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.
- [59] 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.

- [60] 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.
- [61] Raj A Sivakumar and Rajan Batta. The variance-constrained shortest path problem. *Transportation Science*, 28(4):309–316, 1994.
- [62] 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.
- [63] László A Végh. A strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. *SIAM Journal on Computing*, 45(5):1729–1761, 2016.
- [64] 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.
- [65] 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.
- [66] 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.
- [67] 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.
- [68] 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.
- [69] 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.
- [70] Leilei Zhang and Tito Homem-de Mello. An optimal path model for the risk-averse traveler. *Transportation Science*, 51(2):518–535, 2016.
- [71] 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.
- [72] Yuli Zhang, Zuo J Max Shen, and Shiji Song. Parametric search for the bi-attribute concave shortest path problem. *Transportation Research Part B: Methodological*, 94:150–168, 2016.
- [73] Yuli Zhang, Zuo J 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.