A Stochastic Benders Decomposition Scheme for Large-Scale Stochastic Network Design

Dimitris Bertsimas
Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, US,
ORCID: 0000-0002-1985-1003
dbertsim@mit.edu

Ryan Cory-Wright
Department of Analytics, Marketing and Operations, Imperial College Business School, London, UK
ORCID: 0000-0002-4485-0619
r.cory-wright@imperial.ac.uk

Jean Pauphilet
Management Science and Operations, London Business School, London, UK
ORCID: 0000-0001-6352-0984
jpauphilet@london.edu

Periklis Petridis
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA
ORCID: 0000-0002-1019-0763
periklis@mit.edu

Network design problems involve constructing edges in a transportation or supply chain network to minimize construction and daily operational costs. We study a stochastic version where operational costs are uncertain due to fluctuating demand and estimated as a sample average from historical data. This problem is computationally challenging, and instances with as few as 100 nodes often cannot be solved to optimality using current decomposition techniques. We propose a stochastic variant of Benders decomposition that mitigates the high computational cost of generating each cut by sampling a subset of the data at each iteration and nonetheless generates deterministically valid cuts, rather than the probabilistically valid cuts frequently proposed in the stochastic optimization literature, via a dual averaging technique. We implement both single-cut and multi-cut variants of this Benders decomposition, as well as a variant that uses clustering of the historical scenarios. To our knowledge, this is the first single-tree implementation of Benders decomposition that facilitates sampling. On instances with 100–200 nodes and relatively complete recourse, our algorithm achieves 5-7% optimality gaps, compared with 16-27% for deterministic Benders schemes, and scales to instances with 700 nodes and 50 commodities within hours. Beyond network design, our strategy could be adapted to generic two-stage stochastic mixed-integer optimization problems where second-stage costs are estimated via a sample average.

Key words: Generalized Benders Decomposition; Network Design; Stochastic Integer Optimization

1. Introduction

Network design is one of the most famous and frequently studied problems in the Operations Research literature, with widespread applications in logistics, air transportation (Barnhart et al. 2003), supply chains (Santoso et al. 2005, Pishvaee et al. 2014), telecommunications (Balakrishnan et al. 1991), and energy markets (Binato et al. 2001) among other domains. These problems are large-scale and involve uncertain parameters which reflect deviations between the forecast and realized utilization of a network, e.g., uncertain consumer demand in an air traffic control problem or uncertain renewable generation output in a capacity expansion problem. Moreover, we often have data on past realizations of the uncertain parameters. Unfortunately, despite the rapid advances in the scalability of branch-and-bound solvers over the past 25 years, stochastic network design problems with as few as 100 nodes are, to our knowledge, currently regarded as intractable and instead are solved via domain-specific approximation algorithms or heuristics (Crainic et al. 2021a).

To scale to network design problems with up to 50 nodes, the mixed-integer optimization (MIO) community has developed a suite of algorithms for mixed-integer nonlinear problems over the past 25 years, originating with the works of Ceria and Soares (1999), Stubbs and Mehrotra (1999) and refined by Günlük and Linderoth (2009), Crainic et al. (2016) among others. These methods tackle mixed-integer problems with logical constraints and a partially separable objective function, and enforce logical constraints implicitly via perspective functions, thus tightening the Boolean relaxation. Indeed, mixed-integer decomposition schemes that exploit perspective reformulations often solve problems to optimality at sizes an order of magnitude larger than was previously possible; see Fischetti et al. (2017), Bertsimas et al. (2021) for related decomposition schemes.

In a different direction, the machine-learning community has enjoyed considerable success over the past 25 years in improving the scalability of unconstrained stochastic optimization. A common meta-approach is to modify a classical optimization algorithm to sample from an observed dataset at each iteration of the algorithm, and not consider the entire dataset as part of each iterate. Remarkably, each sample often conveys the same essential information as the entire dataset but can be processed multiple orders of magnitude faster. This sampling approach routinely produces a multiple-order-of-magnitude scalability improvement on classical optimization algorithms. Stochastic variants of first-order methods such as Stochastic Gradient Descent (SGD, Davis et al. 2020), the Stochastic Average Gradient method (Schmidt et al. 2017), or Adam (Kingma and Ba 2014) are currently considered to be state-of-the-art for unconstrained problems.

In this paper, we propose to embed a sampling technique within a Benders decomposition scheme (Geoffrion 1972) run on the perspective reformulation (Günlük and Linderoth 2009) of a network design problem. To our knowledge, this is the first single-tree implementation of Benders decomposition that facilitates sampling scenarios while maintaining deterministic optimality guarantees. We demonstrate that this approach obtains bound gaps of 5–7% on instances with 100–200 nodes, three times smaller than the bound gaps obtained by deterministic Benders decomposition schemes in a comparable amount of time. Moreover, our approach successfully scales to obtain bound gaps of 10–40% on instances with 700 nodes and 50 commodities. At this scale, deterministic Benders schemes obtain optimality gaps of 25–55%. Our numerical success can be explained by the fact that sampling allows us to generate significantly more Benders cuts within a given time budget than is possible via a deterministic Benders approach, while conveying most of the essential information stored in each deterministic cut. Although developed for the special case of stochastic multi-commodity capacitated fixed-charge network design problems, we believe our approach could be applied to two-stage stochastic optimization problems where the first-stage variables are discrete, and the second-stage cost is evaluated via a sample average approximation.

1.1. Problem Formulation and Main Contributions

Problem Formulation: We propose a new approach for solving stochastic Multi-commodity Capacitated Fixed-charge Network Design (MCFND) problems to certifiable optimality, which we formally define in the next paragraph. Similar models appear in Magnanti and Wong (1984), Costa (2005), Crainic et al. (2016), Rahmaniani et al. (2018), Ramírez-Pico et al. (2023) among other works.

In MCFND problems, there is an index set of commodities $\mathcal{K}$ to be shipped over a capacitated directed network $(\mathcal{N},\mathcal{E})$, where $\mathcal{N}$ denotes a set of nodes and $\mathcal{E}$ denotes a set of edges. Our overall objective is to perform this transshipment in a manner that minimizes the construction plus flow transportation cost. Let $\boldsymbol{A}$ denote this network's corresponding flow conservation matrix. The capacity of arc $(i,j)\in\mathcal{E}$ is given by $u_{i,j}$ and each node $n\in\mathcal{N}$ supplies or demands an amount $d_n^{k,r}$ of each commodity $k\in\mathcal{K}$ in each scenario $r\in\mathcal{R}$. There is a fixed cost c_{ij} of activating each edge $(i,j)\in\mathcal{E}$, and given this problem data, we introduce binary design variables $z_{i,j}\in\{0,1\}$ to denote whether the (i,j)th edge is activated. In addition to taking activation costs into account in the objective, some applications can also involve a fixed limit on the number of edges to be activated, c_0. The flow variable $x_{ij}^{k,r}$ then denotes the quantity of commodity k routed on edge (i,j) in scenario r, and f_{ij}^k denotes the marginal transportation cost, i.e., the per unit cost of transporting the kth commodity through edge (i,j). Moreover, we follow the standard Sample Average Approximation (SAA) paradigm (see Shapiro et al. 2021, for a general theory) in placing equal weight on each observation of historical data r in our objective.

The complete optimization formulation for MCFND can then be written as:

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in\mathcal{E}} c_{i,j} z_{i,j} + \frac{1}{|\mathcal{R}|} \sum_{r\in\mathcal{R}} \sum_{(i,j)\in\mathcal{E}} \left(\sum_{k\in\mathcal{K}} f^k_{i,j} x^{k,r}_{i,j} + \frac{1}{2\gamma} \left(\sum_{k\in\mathcal{K}} x^{k,r}_{i,j} \right)^2 \right) \\
\text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x}^{k,r} = \boldsymbol{d}^{k,r}, \quad \forall k \in \mathcal{K}, r \in \mathcal{R}, \\
& \sum_{k\in\mathcal{K}} x^{k,r}_{i,j} \le u_{i,j}, \quad \forall (i,j) \in \mathcal{E}, r \in \mathcal{R}, \\
& \boldsymbol{x}^{k,r} \ge 0, \ x^{k,r}_{i,j} = 0 \text{ if } z_{i,j} = 0, \quad \forall (i,j) \in \mathcal{E}, \\
& \sum_{(i,j)\in\mathcal{E}} z_{i,j} \le c_0, \ z_{i,j} \in \{0,1\} \quad \forall (i,j) \in \mathcal{E},
\end{aligned} \tag{1}$$

where $\gamma > 0$ controls a strongly quadratic regularization term in the objective, which can be seen as a penalization of the hard constraint on each edge's capacity, $\sum_{k\in\mathcal{K}} x^{k,r}_{i,j} \le u_{i,j}$ (see also Atamtürk and Günlük (2018) for a discussion of capacity constraints in network design problems). We refer to this term as a "regularization" term throughout the paper, and justify its use from both a theoretical and a practical perspective in Online Supplement OS.1.

Observe that in Problem (1), we link the discrete and continuous decisions in (1) with a logical 'if' statement. In the network design literature, these logical constraints are typically replaced with big-M constraints of the form $\sum_{k\in\mathcal{K}} x^{k,r}_{i,j} \le u_{i,j} z_{i,j}$ by default (Glover 1975). However, there are alternative ways to convexify logical constraints, which sometimes lead to tighter formulations, e.g., by leveraging the presence of the strongly quadratic term in the objective —leading to the so-called perspective formulation, with second-order cone constraints (Ceria and Soares 1999, Günlük and Linderoth 2009)— or by leveraging both the quadratic term and the capacity constraints (as we do in this paper). Accordingly, we formulate network design with logical constraints to facilitate tighter convexifications and stronger Benders cuts than are achievable via the big-M technique alone; see also Wei et al. (2022) for a detailed study of conic formulations that give tighter relaxations of logically constrained problems than big-M relaxations in other contexts.

Main Contributions: In this paper, we provide two main contributions.

First, we propose a new decomposition method that combines sampling-based methods from the stochastic optimization and machine learning literature with a Generalized Benders Decomposition approach in the spirit of Geoffrion (1972). Our approach can tackle large-scale mixed-integer problems by leveraging weak duality to obtain valid dual variables for scenarios we do not explicitly sample. To our knowledge, this is the first single-tree implementation of a Generalized Benders Decomposition scheme that facilitates sampling.

Second, we implement and benchmark our approach across a wide variety of large-scale network design instances, and explore the performance benefits of various design and implementation choices. Our approach allows us to solve network design problems with 200 nodes and relatively complete recourse to within 7% of optimality in hours, and obtain high-quality feasible solutions on instances with relatively complete recourse and up to 700 nodes.

1.2. Background and Literature Review

Our work is built on two intertwined literatures. First, decomposition schemes for large-scale deterministic problems with logical constraints developed by the MIO community. Second, sampling algorithms for problems with exogenous uncertainty developed by the stochastic optimization community. We further remark that, owing to Problem (1)'s significant computational difficulty, a wide variety of approximation algorithms (Agrawal et al. 1991, Goemans and Bertsimas 1993, Bertsimas and Teo 1998) and heuristic methods have also been proposed for solving Problem (1); see Rodríguez-Martín and Salazar-González (2010), Gendron et al. (2018) for reviews.

Cutting-Plane Schemes for Mixed-Integer Optimization: Problem (1) is a computationally challenging mixed-integer problem that encompasses hard combinatorial problems such as Steiner tree optimization (Garey and Johnson 1977) and possesses extremely poor Boolean relaxations (Gendron et al. 1999). Indeed, generic branch-and-bound solvers cannot currently solve network design (ND) problems at even moderate problem sizes with tens of nodes (see Crainic et al. 2021b, Section 6.1 for an investigation of CPLEX version 12.8's performance on synthetic ND instances with ten nodes). Accordingly, and due to its cardinal importance in practice, ND has emerged as one of the most frequently studied problems in the MIO literature over the past 50 years.

Throughout the first 30 years of the field of Operations Research, there was a spirited debate regarding the most efficient technique for solving ND problems, with many proposals, including branch-and-bound (Boyce et al. 1973), Lagrangian methods (Cornuejols et al. 1980), and dynamic programming (Erickson et al. 1987). The idea of solving ND problems via Generalized Benders decomposition (Geoffrion 1972) was moved front-and-center by Magnanti and Wong (1981, 1984). Building upon several influential prior works, including Geoffrion and Graves (1974), Florian et al. (1976), Richardson (1976), they found that an accelerated Benders decomposition was a viable and often more scalable alternative for ND problems than several other optimization approaches, including the three aforementioned ones. Ever since, Benders decomposition has been widely recognized as one of the most competitive methods for solving ND problems; we refer to Fischetti et al. (2017), Crainic et al. (2021b) for modern reviews of Benders decomposition for ND problems.

In a related direction, a significant line of work has developed a suite of cutting planes that iteratively strengthen Problem (1)'s Boolean relaxation upon their imposition; see, e.g., Van Roy and Wolsey (1985), Magnanti et al. (1993, 1995), Bienstock et al. (1998), Günlük (1999), Atamtürk and Günlük (2021) and references therein. Remarkably, these approaches are so numerically successful and easy to implement that they are usually incorporated within commercial branch-and-cut solvers within several years of their proposal (Bixby 2012). As a result, some of the decomposition schemes reviewed above may even be considered "sleeping beauties" in the sense of Ke et al. (2015), i.e., were not originally considered numerically successful but would be if proposed today, implicitly in conjunction with these valid inequalities.

Decomposition Schemes for Large-Scale Optimization Under Uncertainty: Cotemporally, a considerable amount of attention has been devoted by the stochastic optimization community to solving large-scale convex optimization problems with uncertain parameters for which we have access to either a joint probability distribution or observations from historical data. Initiated by the independent works of Dantzig (1955), Beale (1955), and subsequently refined by Wets (1966), Van Slyke and Wets (1969), contemporary optimizers for large-scale stochastic problems typically invoke the Minkowski-Weyl theorem (c.f. Bertsimas and Tsitsiklis 1997, Chapter 4) to solve their deterministic equivalents via Benders decomposition (which was termed the L-shaped method by Van Slyke and Wets 1969). Alternatively, works like Zakeri et al. (2000), Fábián (2000), Rei et al. (2009), Guigues (2020) propose generating Benders cuts without solving each subproblem to optimality.

The two main variants of Benders decomposition invoked for two-stage stochastic integer optimization problems such as Problem (1) are called single-cut and multi-cut Benders. Single-cut schemes maintain a single epigraph variable that upper bounds the expected transshipment cost and generates a single cut at each iteration of Benders decomposition. Multi-cut schemes associate a separate epigraph variable with the cost incurred in each scenario and generate a separate cut for each epigraph variable in each iteration (Birge and Louveaux 1988). Therefore, single-cut schemes typically require more iterations to converge but require less time to perform each iteration (see Birge and Louveaux 2011, de Camargo et al. 2008, You and Grossmann 2013, for comparisons). Problems with fewer scenarios are typically solved faster via multi-cut approaches. However, the relative performance of each variant is highly problem-dependent.

More recently, considerable attention has been devoted to designing variants of Benders decomposition that avoid solving a subproblem for each scenario at each iteration by sampling. Higle and Sen (1991), Pereira and Pinto (1991), Dantzig and Infanger (1993), Infanger (1992) initiated this line of inquiry by proposing stochastic cutting-plane schemes that converge almost surely (see also Bertsimas and Li 2022). Determining convergence of these schemes is technically challenging. Various statistical tests exist (see, e.g., Higle and Sen 1996, Morton 1998, Mak et al. 1999) that provide confidence intervals on the duality gap. Yet, to avoid multiple-testing problems, practitioners typically run stochastic cutting-plane methods for a prespecified number of iterations and then perform a statistical test on termination (De Matos et al. 2015).

1.3. Structure

We propose a stochastic Benders decomposition scheme that combines the perspective reformulation technique from the MIO literature with sampling ideas from the stochastic optimization literature to, for the first time, successfully solve data-driven capacitated network design problems with hundreds of nodes to certifiable (near) optimality. The rest of this paper is laid out as follows:

- In Section 2, we propose stochastic variants of the single- and multi-cut versions of Benders decomposition to solve a perspective reformulation of (1). Our algorithms randomly sample a subset of scenarios $\mathcal{R}_t \subseteq \mathcal{R}$ at each iteration and use a dual averaging technique to generate cuts that are deterministically valid for all $r \in \mathcal{R}$, while previous stochastic approaches generate cuts that are only valid on average or with high probability. We prove high probability bounds on the approximation error stemming from our dual averaging technique.
- In Section 3, we propose rigorous convergence criteria to terminate our stochastic decomposition schemes at a certifiable optimal solution. Since our master optimization problem is an MIO problem, we also discuss the specific termination challenges arising when Benders Decomposition is implemented via branch-and-cut (or lazy constraints). We also review techniques for accelerating the convergence of our methods, by warm-starting their upper and lower bounds.
- In Section 4, we apply our decomposition schemes to a collection of network design instances that are synthetically generated or obtained from the literature (Crainic et al. 2016, 2021a). On the synthetic instances (which exhibit relatively complete recourse), our best stochastic cutting-plane strategy achieves 7–11% (resp. 20–30%) optimality gaps within two hours for instances with 70–300 nodes (resp. 500-700 nodes) compared with 12–26% (resp. 50–55%) for its deterministic counterpart. On the **R** instances introduced by Crainic et al. (2000) and frequently benchmarked against in the literature, we find that our approach provides a noticeable reduction in optimality gap (by 5–10 percentage points) as the number of scenarios increases. The code for our experiments is available at Bertsimas et al. (2024).

Notation

We let non-boldface characters such as b denote scalars, lowercase bold-faced characters ($\boldsymbol{x}$) denote vectors, uppercase bold-faced characters ($\boldsymbol{A}$) denote matrices, and calligraphic uppercase characters ($\mathcal{Z}$) denote sets. We let $[n]$ denote the running set of indices $\{1, \dots, n\}$. We let $\boldsymbol{e}$ denote the vector of ones, and $\mathbf{0}$ denote the vector of all zeros. Finally, we let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product between two vectors of the same size, i.e., $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \sum_{i=1}^{n} x_i y_i$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

2. Deterministic and Stochastic Cutting-Plane Methods

This section proposes an efficient numerical strategy for solving Problem (1) to certifiable optimality. The backbone of our approach is a Generalized Benders Decomposition scheme run on a perspective reformulation of Problem (1), which uses sampling techniques to avoid explicitly solving each scenario at each iteration of the method. Instead, we use dual-optimal solutions from the sampled subproblems to construct dual-feasible solutions to the remaining subproblems and thereby construct valid cuts. We further discuss the convergence properties of our method.

2.1. A Two-Stage Reformulation

We observe that the flow minimization problem with respect to each $\boldsymbol{x}^{\cdot,r}$ in (1) is decomposable across scenarios $r \in \mathcal{R}$. Therefore, consider a set of demand vectors $\boldsymbol{d}^k \in \mathbb{R}^{\mathcal{N}}$ for $k \in \mathcal{K}$ and define

$$f(\boldsymbol{z};\boldsymbol{d}) := \min_{\boldsymbol{x}^k \in \mathbb{R}_+^{\mathcal{E}}, k\in\mathcal{K}} \sum_{k\in\mathcal{K}} \langle \boldsymbol{f}^k, \boldsymbol{x}^k \rangle + \frac{1}{2\gamma} \sum_{(i,j)\in\mathcal{E}} \left(\sum_{k\in\mathcal{K}} x_{i,j}^k \right)^2 \text{ s.t. } \boldsymbol{A}\boldsymbol{x}^k = \boldsymbol{d}^k, \forall k \in \mathcal{K}, \tag{2}$$

$$\sum_{k\in\mathcal{K}} x_{i,j}^k \leq u_{i,j}, \forall (i,j) \in \mathcal{E},$$

$$x_{i,j}^k = 0 \textit{ if } z_{i,j} = 0, \forall (i,j) \in \mathcal{E},$$

to be the operational cost of serving demand $\boldsymbol{d}$ on network $(\mathcal{N}, \mathcal{E})$ with design variables $\boldsymbol{z}$. Observe that the minimization problem defining $f(\boldsymbol{z};\boldsymbol{d})$ is not decomposable across commodities because of shared capacity constraints. With this notation, Problem (1) is equivalent to

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \langle \boldsymbol{c}, \boldsymbol{z} \rangle + \frac{1}{|\mathcal{R}|} \sum_{r\in\mathcal{R}} f(\boldsymbol{z}; \boldsymbol{d}^{\cdot,r}), \tag{3}$$

where $\boldsymbol{d}^{\cdot,r}$ denotes the collection of demand vectors $\{\boldsymbol{d}^{k,r}, k \in \mathcal{K}\}$ and $\mathcal{Z} = \{\boldsymbol{z} \in \{0,1\}^{\mathcal{E}} : \sum_{(i,j)} z_{i,j} \leq c_0\}$ denotes the set of feasible edges. The network design formulation (3) separates the discrete design variables $\boldsymbol{z}$ from the continuous second-stage routing variables $\boldsymbol{x}^k$, thus giving a pure integer optimization formulation that is readily amenable to outer-approximation techniques.

2.2. A Linear Lower Approximation of the Second-Stage Cost Function

In this section, we derive a family of Benders cuts that successfully outer-approximate a perspective reformulation of (2).

Since the objective function in (3) involves the average of the function $f(\boldsymbol{z}, \boldsymbol{d})$ over $|\mathcal{R}|$ realizations of $\boldsymbol{d}$, we start by analyzing properties of the function $f(\boldsymbol{z}, \boldsymbol{d})$ in isolation, with a view to establish that $f(\boldsymbol{z}, \boldsymbol{d})$ is convex in $\boldsymbol{z}$ and a valid subgradient can be obtained by solving a dual problem, as has already been done in the literature, e.g., in Bertsimas et al. (2021).

Proposition 1. *For any $\boldsymbol{z} \in \{0,1\}^{\mathcal{E}}$ and demand vectors $\boldsymbol{d}^k$, $k \in \mathcal{K}$ such that Problem (2) admits a feasible solution, we have:*

$$f(\boldsymbol{z};\boldsymbol{d}) = \max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^{\mathcal{E}}, \boldsymbol{\beta}\in\mathbb{R}_+^{\mathcal{E}} \\ \boldsymbol{p}^k\in\mathbb{R}^{\mathcal{N}}, k\in\mathcal{K}}} \sum_{k\in\mathcal{K}} \left\langle \boldsymbol{p}^k, \boldsymbol{d}^k \right\rangle - \sum_{(i,j)\in\mathcal{E}} z_{i,j} \left[u_{i,j}\beta_{i,j} + \frac{\gamma}{2}(\alpha_{i,j} + \beta_{i,j})^2 \right] \; s.t. \; \boldsymbol{A}^\top \boldsymbol{p}^k \leq \boldsymbol{f}^k - \boldsymbol{\alpha}. \tag{4}$$

The proof of Proposition 1 follows analogously to Bertsimas et al. (2021, Theorem 2.5) and relies on deriving the dual of the minimization problem defining $f(\boldsymbol{z};\boldsymbol{d})$ by using a variable decomposition *à la Fenchel*; for completeness, we provide a formal proof in Online Supplement OS.2.1. Observe that the optimization problem (4) remains well defined if there are no hard constraints on edge capacity (i.e., if $u_{i,j}=+\infty$, we set $\beta_{i,j}=0$) or if there is no quadratic term in the objective (i.e., if $\gamma=+\infty$ we set $\alpha_{i,j}+\beta_{i,j}=0$).

Proposition 1 calls for a few observations. First, according to the dual reformulation, $f(\boldsymbol{z};\boldsymbol{d})$ can be expressed as the point-wise maximum of affine functions in $\boldsymbol{z}$, hence $f(\boldsymbol{z};\boldsymbol{d})$ is convex in $\boldsymbol{z}$. Second, any feasible dual solution $\boldsymbol{\alpha}\in\mathbb{R}^{\mathcal{E}}$, $\boldsymbol{\beta}\in\mathbb{R}^{\mathcal{E}}_{+}$, $\boldsymbol{p}^k\in\mathbb{R}^{\mathcal{N}}$ such that $\boldsymbol{A}^\top\boldsymbol{p}^k\leq\boldsymbol{f}^k-\boldsymbol{\alpha}$ provides a valid linear lower approximation of $f(\boldsymbol{z};\boldsymbol{d})$. Namely, for any $\boldsymbol{z}$,

$$f(\boldsymbol{z};\boldsymbol{d})\geq\sum_{k\in\mathcal{K}}\langle\boldsymbol{p}^k,\boldsymbol{d}^k\rangle-\sum_{(i,j)\in\mathcal{E}}z_{i,j}\left[u_{i,j}\beta_{i,j}+\frac{\gamma}{2}(\alpha_{i,j}+\beta_{i,j})^2\right].$$

When the dual variables are optimal for a particular vector $\boldsymbol{z}^0$, the resulting offset and slope in the above linear approximation are exactly the value of $f(\boldsymbol{z}^0;\boldsymbol{d})$ and a subgradient of f at $\boldsymbol{z}^0$, i.e.,

$$f(\boldsymbol{z};\boldsymbol{d})\geq f(\boldsymbol{z}^0;\boldsymbol{d})+\langle\nabla f(\boldsymbol{z}^0;\boldsymbol{d}),\boldsymbol{z}-\boldsymbol{z}^0\rangle.$$

Third, Proposition 1 applies if Problem (2) is feasible for the current design vector $\boldsymbol{z}=\boldsymbol{z}^0$. On the other hand, if (2) is not feasible, then the following feasibility problem does not admit a solution:

$$\exists\,\boldsymbol{x}\in\mathbb{R}^{\mathcal{E}\times\mathcal{K}}_{+}:\ \boldsymbol{A}\boldsymbol{x}^k=\boldsymbol{d}^k\ \forall k\in\mathcal{K},\ \sum_{k\in\mathcal{K}}x^k_{i,j}\leq u_{i,j}z^0_{i,j},\ \forall(i,j)\in\mathcal{E}.$$

Hence, by Farkas's lemma (see, e.g., Bertsimas and Tsitsiklis 1997, Theorem 4.6), we can find a certificate of infeasibility, i.e., we can find $\boldsymbol{\beta}\in\mathbb{R}^{\mathcal{E}}_{+},\boldsymbol{p}^k\in\mathbb{R}^{\mathcal{N}},k\in\mathcal{K}$ such that $\boldsymbol{A}^\top\boldsymbol{p}^k-\boldsymbol{\beta}\leq\boldsymbol{0}$ and $\sum_{k\in\mathcal{K}}\langle\boldsymbol{p}^k,\boldsymbol{d}^k\rangle-\sum_{(i,j)\in\mathcal{E}}z^0_{i,j}u_{i,j}\beta_{i,j}>0$. In particular, the existence of such vectors $\boldsymbol{\beta}$, $\{\boldsymbol{p}^k\}_{k\in\mathcal{K}}$ implies that Problem (4) is unbounded. Therefore, we can separate the infeasible incumbent solution $\boldsymbol{z}^0$ by imposing the feasibility cut

$$\sum_{k\in\mathcal{K}}\langle\boldsymbol{p}^k,\boldsymbol{d}^k\rangle-\sum_{(i,j)\in\mathcal{E}}z_{i,j}u_{i,j}\beta_{i,j}\leq 0 \tag{5}$$

on the first-stage variable $\boldsymbol{z}$.

Finally, as has already been observed in the literature (Xie and Deng 2020, Bertsimas et al. 2021), our reformulation can alternatively be achieved by performing a perspective reformulation on (2) to rewrite it as a mixed-integer second-order cone problem (c.f. Günlük and Linderoth 2009) and taking the dual of this perspective reformulation with respect to the continuous variables.

2.3. Epigraph Formulations: Modeling Choice and Algorithmic Implications

In this section, we exploit our previously developed characterization of $f(\boldsymbol{z}, \boldsymbol{d})$ as the pointwise maximum of functions linear in $\boldsymbol{z}$ to revisit three deterministic outer-approximation methods that solve Problem (1) to certifiable optimality. For simplicity, we focus our description on optimality cuts in this section; feasibility cuts follow in much the same way.

Outer-approximation methods such as generalized Benders decomposition solve (3) by constructing a lower approximation of the second-stage operational cost $\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r})$ and refining this approximation at each step. However, since the second-stage cost is the average operational cost over $|\mathcal{R}|$ scenarios, one can either approximate each term $f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r})$ separately or their sum, which we refer to as multi-cut and single-cut approaches respectively.

In a multi-cut approach, we consider the following epigraph formulation of Problem (3), as originally proposed by Birge and Louveaux (1988) for two-stage stochastic linear optimization:

$$\min_{\substack{\boldsymbol{z}\in\mathcal{Z}\\ \eta_r\in\mathbb{R},\forall r\in\mathcal{R}}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \eta_r \text{ s.t. } \eta_r \geq f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r}),\ \forall r\in\mathcal{R},$$

and iteratively refine a piecewise linear lower approximation of $f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r})$ for each epigraph constraints until convergence. Specifically, at each iteration T, the multi-cut cutting-plane algorithm solves the MIO problem

$$\min_{\substack{\boldsymbol{z}\in\mathcal{Z}\\ \eta_r\in\mathbb{R},\forall r\in\mathcal{R}}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \eta_r \text{ s.t. } \eta_r \geq f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r}) + \langle \nabla f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r}), \boldsymbol{z}-\boldsymbol{z}^t\rangle,\ \forall t\in[T],\ \forall r\in\mathcal{R}. \tag{6}$$

Observe that, in this implementation, each of the $|\mathcal{R}|$ functions $f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r})$ is linearized at T points $\boldsymbol{z}^t$, so (6) comprises $|\mathcal{R}|\times T$ linear constraints. The solution of (6), $\boldsymbol{z}^{T+1}$, then serves as a linearization point to further improve the approximations of the functions $f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r})$ at the next iteration.

Alternatively, the single-cut approach, as originally proposed for two-stage stochastic linear optimization by Van Slyke and Wets (1969), considers a more compact epigraph formulation:

$$\min_{\substack{\boldsymbol{z}\in\mathcal{Z}\\ \eta\in\mathbb{R}}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \frac{1}{|\mathcal{R}|}\eta \text{ s.t. } \eta \geq \sum_{r\in\mathcal{R}} f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r}),$$

and constructs a piece-wise linear lower-approximation of $\sum_{r\in\mathcal{R}} f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r})$ directly. In a single-cut cutting-plane algorithm, at a given iteration T, the epigraph constraint is replaced by linear constraints of the form

$$\eta \geq \sum_{r\in\mathcal{R}} f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r}) + \left\langle \sum_{r\in\mathcal{R}} \nabla f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r}), \boldsymbol{z}-\boldsymbol{z}^t \right\rangle. \tag{7}$$

The single-cut approach involves only one epigraph variable η (compared with $|\mathcal{R}|$ in the multi-cut implementation) and adds one linear constraint at each iteration (vs. $|\mathcal{R}|$). As a result, the MIO problems involved in the single-cut approach are smaller and usually more tractable than those solved by the multi-cut approach. Yet, multi-cut methods approximate the second-stage cost function more accurately and might require fewer iterations to converge. Various studies, including Birge and Louveaux (2011), de Camargo et al. (2008), You and Grossmann (2013) have reported mixed results on the relative merits of single and multi-cut methods, and which method works best appears to depend on the underlying problem and the number of scenarios.

Regarding feasibility, if there exists a scenario $\boldsymbol{d}^{\cdot,r}$ for which the incumbent solution $\boldsymbol{z}^T$ is not feasible (i.e., $f(\boldsymbol{z}^T,\boldsymbol{d}^{\cdot,r})=+\infty$), then a feasibility cut of the form (5) is imposed. In the single-cut approach, the feasibility cut is imposed instead of an optimality cut (7). However, in the multi-cut approach, optimality cuts on the other epigraph variables $\eta_{r'}, r'\neq r$, which correspond to feasible scenarios, can still be added.

We remark that all these methods converge in a finite but possibly exponential number of iterations by the finiteness of $\{0,1\}^{\mathcal{E}}$ and since no method visits a binary vector $\boldsymbol{z}$ twice (see also Geoffrion 1972, Theorem 2.4). A common thread between these approaches is that evaluating values of functions of the form $f(\boldsymbol{z},\boldsymbol{d})$ (and their subgradients) —an operation referred to as calling a separation oracle— is the main computational bottleneck, and the number of function evaluations is the same, $|\mathcal{R}|$, which can be prohibitive, especially when the number of past scenarios $|\mathcal{R}|$ increases. Accordingly, we propose stochastic versions of these approaches with improved per-iteration complexity in the next section.

Remark 1. To successfully combine the best aspects of single and multi-cut approaches, Trukhanov et al. (2010), Contreras et al. (2011) proposed to partition the scenarios into subsets of similar scenarios and introduce one epigraph variable per cluster. For conciseness, we discuss this approach (and propose a stochastic variant), which we refer to as a τ-cut approach, in Online Supplement OS.3.

2.4. A Stochastic Cutting-Plane Algorithm

In this section, we propose stochastic variants of the cutting-plane methods proposed in the previous section, which obtain high-quality deterministically valid lower bounds without explicitly solving an optimization problem in each scenario and each commodity at each iteration of the method. We also discuss the convergence of these methods. As these methods do not provide deterministically valid upper bounds from a single sample, we defer a detailed discussion of their upper bounds, the corresponding termination criteria, and their single-tree implementation to Section 3, and assume for ease of exposition that all cutting-plane methods are multi-tree throughout the section.

First, a stochastic variant of the multi-cut algorithm can be developed in a straightforward manner. Indeed, in its deterministic implementation, at each iteration t of the multi-cut cutting-plane algorithm, we add one linear constraint for each epigraph variable η_r, for $r\in\mathcal{R}$. Instead, we can sample a subset $\mathcal{R}_t \subseteq \mathcal{R}$ of scenarios and only add linear constraints for these scenarios. Formally, at iteration T of the algorithm, we solve

$$\min_{\substack{\boldsymbol{z}\in\mathcal{Z} \\ \eta_r\in\mathbb{R}, r\in\mathcal{R}}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \eta_r \text{ s.t. } \eta_r \geq f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}) + \langle \nabla f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}), \boldsymbol{z} - \boldsymbol{z}^t\rangle, \forall t \in [T], \forall r \in \mathcal{R}_t,$$

instead of (6), as sketched in the multi-tree case in Algorithm 1 (we defer a detailed discussion of its single-tree implementation and termination criteria to Section 3). Consequently, each iteration only requires solving $|\mathcal{R}_{T+1}|$ optimization problems that define $f(\boldsymbol{z}^{T+1}; \boldsymbol{d}^{\cdot,r})$, which can be significantly faster. Moreover, it is not too hard to see that this algorithm converges almost surely under any reasonable sampling scheme (e.g., sampling subsets of $\mathcal{R}$ of fixed cardinality uniformly) since we almost surely sample each subset $\mathcal{R}_t$ infinitely often and there are finitely many binaries. Note that, in the pseudo-code, Algorithm 1 is initialized with a set of valid constraints generated from scenarios $r \in \mathcal{R}_0$. However, in practice, these constraints do not have to be generated at $\boldsymbol{z}_0$, nor be binding. We can initialize the algorithm with any set of valid (linear) constraints on $(\boldsymbol{z}, \boldsymbol{\eta})$.

Algorithm 1 A Multi-Cut Sample-Based Cutting Plane Method

```
initialize $\boldsymbol{z}_0$; $f(\boldsymbol{z}_0; \boldsymbol{d}^{\cdot,r}), \nabla f(\boldsymbol{z}_0; \boldsymbol{d}^{\cdot,r}), \forall r \in \mathcal{R}_0$.
set $T \leftarrow 0$
repeat
    compute $\boldsymbol{z}^{T+1}, \boldsymbol{\eta}^{T+1} \leftarrow \arg\min_{\boldsymbol{z},\boldsymbol{\eta_r}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \eta_r$
        s.t. $\eta_r \geqslant f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}) + \langle \nabla f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}), \boldsymbol{z} - \boldsymbol{z}^t\rangle, \forall t \in [T], \forall r \in \mathcal{R}_t$,
    sample $\mathcal{R}_{T+1} \subseteq \mathcal{R}$
    calculate $f(\boldsymbol{z}^{T+1}; \boldsymbol{d}^{\cdot,r}), \nabla f(\boldsymbol{z}^{T+1}; \boldsymbol{d}^{\cdot,r})$ for $r \in \mathcal{R}_{T+1}$
    set $T \leftarrow T + 1$
until Termination Criterion Met
```

On the other hand, developing a stochastic version of the single-cut method is technically challenging because constraint (7) aggregates information across scenarios. To address this issue, Infanger (1992) propose generating probabilistic cuts by sampling a subset of scenarios $\mathcal{R}_t \subseteq \mathcal{R}$ at each iteration and imposing the constraint

$$\frac{|\mathcal{R}_t|}{|\mathcal{R}|} \times \eta \geq \sum_{r\in\mathcal{R}_t} f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}) + \left\langle \sum_{r\in\mathcal{R}_t} \nabla f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}), \boldsymbol{z} - \boldsymbol{z}^t \right\rangle, \tag{8}$$

instead of (7), where the quantities $\frac{|\mathcal{R}|}{|\mathcal{R}_t|}\sum_{r\in\mathcal{R}_t} f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r})$ and $\frac{|\mathcal{R}|}{|\mathcal{R}_t|}\sum_{r\in\mathcal{R}_t} \nabla f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r})$ are unbiased estimates of the original offset and slope terms, $\sum_{r\in\mathcal{R}} f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r})$ and $\sum_{r\in\mathcal{R}} \nabla f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r})$ respectively, so that (8) is a reasonable approximation of the original constraint (7). This intuition is similar to that of SGD in unconstrained continuous optimization. Unfortunately, these cuts are only valid probabilistically and may cut off part of the feasible region when combined. Moreover, while the sampled cuts are unbiased estimates of the slope, optimizing these estimates via Benders decomposition yields solutions that suffer from the so-called optimizer's curse (Smith and Winkler 2006). SGD shares the same drawbacks but mitigates them by performing only one gradient step at each iteration, hence 'forgetting' estimation errors between iterations. Conversely, in a cutting-plane algorithm, cuts added at one iteration are imposed in subsequent iterations, until termination.

We reconcile the computational benefits of sampling with the aforementioned drawbacks of the stochastic single-cut approach by leveraging the dual formulation of $f(\boldsymbol{z};\boldsymbol{d})$ in Proposition 1 to derive deterministically valid lower-approximations for scenarios r that are not sampled. Further, we argue that provided the sampled scenarios are sufficiently representative of the remaining scenarios, this approximation is sufficiently accurate that we eventually obtain a near-optimal solution with high probability; see also Zakeri et al. (2000) for an "inexact" Benders decomposition method.

Specifically, recall that any feasible dual solution $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{p})$ provides a valid lower bound:

$$f(\boldsymbol{z};\boldsymbol{d}^{\cdot,r}) \geq q(\boldsymbol{z}^t,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{p};\boldsymbol{d}^{\cdot,r}) + \langle \nabla_{\boldsymbol{z}} q(\boldsymbol{z}^t,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{p};\boldsymbol{d}^{\cdot,r}), \boldsymbol{z}-\boldsymbol{z}^t\rangle\,,$$

with $q(\boldsymbol{z},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{p};\boldsymbol{d}) := \sum_{k\in\mathcal{K}}\langle \boldsymbol{p}^k,\boldsymbol{d}^k\rangle - \sum_{(i,j)\in\mathcal{E}} z_{i,j}\left[u_{i,j}\beta_{i,j} + \frac{\gamma}{2}(\alpha_{i,j}+\beta_{i,j})^2\right]$. Hence, we replace (7) by a constraint of the form

$$\eta \geq \sum_{r\in\mathcal{R}} q(\boldsymbol{z}^t,\boldsymbol{\alpha}^r,\boldsymbol{\beta}^r,\boldsymbol{p}^r;\boldsymbol{d}^{\cdot,r}) + \sum_{r\in\mathcal{R}} \langle \nabla_{\boldsymbol{z}} q(\boldsymbol{z}^t,\boldsymbol{\alpha}^r,\boldsymbol{\beta}^r,\boldsymbol{p}^r;\boldsymbol{d}^{\cdot,r}), \boldsymbol{z}-\boldsymbol{z}^t\rangle\,, \tag{9}$$

for some feasible dual solutions $(\boldsymbol{\alpha}^r,\boldsymbol{\beta}^r,\boldsymbol{p}^r)$. Observe that, unlike (8), the constraint (9) is a deterministically valid (although not necessarily tight) lower bound on the true operational cost.

Collecting these observations yields our overall stochastic single-cut approach: First, to reduce the computational burden of solving an optimization problem for each scenario, at each iteration, we only solve a random subset of scenarios $r \in \mathcal{R}_t \subseteq \mathcal{R}$ –hence effectively computing $f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r})$ and $\nabla f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r})$. Second, for the remaining scenarios $r \notin \mathcal{R}_t$, we refrain from solving (4) and instead use the cheap to compute and feasible dual average solution $(\boldsymbol{\alpha}^r,\boldsymbol{\beta}^r,\boldsymbol{p}^r) = (\bar{\boldsymbol{\alpha}}^{\mathcal{R}_t},\bar{\boldsymbol{\beta}}^{\mathcal{R}_t},\bar{\boldsymbol{p}}^{\mathcal{R}_t}) := \frac{1}{|\mathcal{R}_t|}\sum_{r'\in\mathcal{R}_t}(\boldsymbol{\alpha}^{r'},\boldsymbol{\beta}^{r'},\boldsymbol{p}^{r'})$ instead. This gives a stochastic cutting-plane method with a sequence of deterministically valid non-decreasing lower bounds, which we formalize in Algorithm 2 (we defer a detailed discussion of its single-tree implementation and termination criterion to Section 3).

However, whether this method converges towards an optimal solution (e.g., in a limit) or generates a never-ending sequence of deterministically valid but not tight cuts is not obvious. We now provide some reassurance in this direction, by showing that for the incumbent solution $\boldsymbol{z}^t$, the approximation error of cuts obtained via dual averaging can be decomposed, with high probability, as the sum of two terms: one term that depends on the variance of the optimal dual variables and that captures the heterogeneity in the demand scenarios, and one estimation error term that vanishes as $|\mathcal{R}_t|$ grows (proof deferred to Online Supplement OS.2.2):

Proposition 2. *Fix $\boldsymbol{z}^t$. For any $r \in \mathcal{R}$, denote $(\boldsymbol{\alpha}^r, \boldsymbol{\beta}^r, \boldsymbol{p}^r)$ the optimal dual solutions of* (4) *for $\boldsymbol{z} = \boldsymbol{z}^t$ and $\boldsymbol{d} = \boldsymbol{d}^{\cdot,r}$. Denote ν^2 the variance in optimal dual variables, defined as*

$$\nu^2 = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \left\| (\boldsymbol{\alpha}^r, \boldsymbol{\beta}^r, \boldsymbol{p}^r) - (\bar{\boldsymbol{\alpha}}^{\mathcal{R}}, \bar{\boldsymbol{\beta}}^{\mathcal{R}}, \bar{\boldsymbol{p}}^{\mathcal{R}}) \right\|^2 \quad \textit{with } (\bar{\boldsymbol{\alpha}}^{\mathcal{R}}, \bar{\boldsymbol{\beta}}^{\mathcal{R}}, \bar{\boldsymbol{p}}^{\mathcal{R}}) = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} (\boldsymbol{\alpha}^r, \boldsymbol{\beta}^r, \boldsymbol{p}^r).$$

Then, there exist universal constants $L, M > 0$ such that, for any $\delta \in (0, e^{-1})$, when $\mathcal{R}_t$ is sampled without replacement from $\mathcal{R}$ with a fixed size $|\mathcal{R}_t|$, we have with probability $1 - 3\delta$:

$$\sum_{r \notin \mathcal{R}_t} \left| q(\boldsymbol{z}^t, \bar{\boldsymbol{\alpha}}^{\mathcal{R}_t}, \bar{\boldsymbol{\beta}}^{\mathcal{R}_t}, \bar{\boldsymbol{p}}^{\mathcal{R}_t}; \boldsymbol{d}^r) - f(\boldsymbol{z}^t; \boldsymbol{d}^r) \right| \le L\sqrt{|\mathcal{R} \setminus \mathcal{R}_t|}\nu + D\sqrt{|\mathcal{R} \setminus \mathcal{R}_t| \log(1/\delta)}, \tag{10}$$

with

$$D := L\, M \sqrt{2|\mathcal{E}| + |\mathcal{N}| \times |\mathcal{K}|} \left[\sqrt{|\mathcal{R}|} \left(\frac{1}{|\mathcal{R}_t|} - \frac{1}{|\mathcal{R}|} \right)^{1/2} + \left(\frac{1}{|\mathcal{R} \setminus \mathcal{R}_t|} - \frac{1}{|\mathcal{R}|} \right)^{1/4} \right].$$

Algorithm 2 A Single-Cut Sample-Based Cutting Plane Method

```
initialize $\boldsymbol{z}_1$; $f(\boldsymbol{z}_0; \boldsymbol{d}^{\cdot,r}), \nabla f(\boldsymbol{z}_0; \boldsymbol{d}^{\cdot,r}), \forall r \in \mathcal{R}_0$.
set $T \leftarrow 1$
repeat
    compute $\boldsymbol{z}^{T+1}, \eta^{T+1} \leftarrow \arg\min_{\boldsymbol{z},\eta} \langle \boldsymbol{c}, \boldsymbol{z} \rangle + \frac{1}{|\mathcal{R}|}\eta$
        s.t. $\eta \geqslant \sum_{r \in \mathcal{R}} q(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}) + \langle \nabla q(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}), \boldsymbol{z} - \boldsymbol{z}^t \rangle, \forall t \in [T]$,
    sample $\mathcal{R}_{T+1} \subseteq \mathcal{R}$
    calculate $f(\boldsymbol{z}^{T+1}; \boldsymbol{d}^{\cdot,r}), \nabla f(\boldsymbol{z}^{T+1}; \boldsymbol{d}^{\cdot,r})$ for $r \in \mathcal{R}_{T+1}$
    set $T \leftarrow T + 1$
until Termination Criterion Met
```

Proposition 2 provides a probabilistic guarantee on the quality of each cut in terms of the sample size $|\mathcal{R}_t|$. Observe that the approximation error is proportional to $\sqrt{|\mathcal{R} \setminus \mathcal{R}_t|}$, which means that the approximation error is zero in the limit where $\mathcal{R}_t \to \mathcal{R}$ (as expected) but which also means that the approximation error grows sub-linearly in the number of scenarios to approximate $|\mathcal{R} \setminus \mathcal{R}_t|$.

Moreover, by the probabilistic method (see, e.g., Grimmett and Stirzaker 2020), Proposition 2 reveals that, for any $\boldsymbol{z}^t$ and sufficiently small δ, there exists some $\mathcal{R}_t$ such that this guarantee holds deterministically. Indeed, setting $\delta < (1 - \binom{|\mathcal{R}|}{|\mathcal{R}_t|}^{-1})/3$ reveals that, with the notations of Proposition 2, repeatedly sampling $\mathcal{R}_t$ for a given $\boldsymbol{z}^t$ eventually gives a cut which is an underestimator of $f(\boldsymbol{z}^t)$ by at most ρ, where

$$\rho := L\sqrt{|\mathcal{R} \setminus \mathcal{R}_t|}\nu + D\sqrt{|\mathcal{R} \setminus \mathcal{R}_t| \log(1/\delta)}. \tag{11}$$

The above observation implies that running Algorithm 2 without termination and selecting a $\boldsymbol{z}^t$, which minimizes our underestimator in the limit, almost surely returns a ρ-optimal solution to Problem (1), where ρ is defined by Equation (11). Therefore, in practice, when Algorithm 2's lower bound stabilizes, we can either increase the number of scenarios sampled (and thus reduce ρ), or terminate with confidence if, according to a statistical test, the gap between our stochastic upper bound (see Section 3) and our deterministic lower bound is sufficiently small. As we observe in our numerical results (see Section 4), the optimality gap from single-cut at termination with a sample rate of around 10% is usually quite small in practice.

We conclude this section with two remarks that contrast our approach with the recent work of Ramírez-Pico et al. (2023) and incorporate dual averaging within our multi-cut method respectively:

Remark 2. Recently, (Ramírez-Pico et al. 2023) proposed an adaptive scenario aggregation scheme that applies to stochastic network design problems. Their scheme clusters scenarios into groups, and generates a lower bound on the average cost within each group by the cost associated with the average scenario for that group (via Jensen's inequality). Their approach shares some commonalities with this work, chiefly applying a separation oracle to a subset of scenarios in a stochastic network design problem, rather than all scenarios. However, it differs from our approach in two important aspects: First, Ramírez-Pico et al. (2023) aggregate demand vectors $\boldsymbol{d}^r$ within each group, while we aggregate optimal dual variables. Second, their clustering of scenarios into groups (hence, the scenarios passed to the separation oracle) is fixed throughout Benders algorithm and only refined after termination, while we sample a new subset for each incumbent explored through the algorithm.

Remark 3. Although the dual averaging technique is not needed to develop a stochastic multi-cut cutting-plane algorithm, it can be used to improve its convergence. In Algorithm 1, instead of only imposing a new cut for the epigraph variables η_r with $r \in \mathcal{R}_t$, we can also use dual averaging to impose one additional constraint on the variables $\eta_r, r \notin \mathcal{R}_t$. Formally,

$$\sum_{r \notin \mathcal{R}_t} \eta_r \geq \sum_{r \notin \mathcal{R}_t} q(\boldsymbol{z}^t, \bar{\boldsymbol{\alpha}}^{\mathcal{R}_t}, \bar{\boldsymbol{\alpha}}^{\mathcal{R}_t}, \bar{\boldsymbol{p}}^{\mathcal{R}_t}; \boldsymbol{d}^{\cdot,r}) + \sum_{r \notin \mathcal{R}_t} \left\langle \nabla_{\boldsymbol{z}} q(\boldsymbol{z}^t, \bar{\boldsymbol{\alpha}}^{\mathcal{R}_t}, \bar{\boldsymbol{\beta}}^{\mathcal{R}_t}, \bar{\boldsymbol{p}}^{\mathcal{R}_t}; \boldsymbol{d}^{\cdot,r}), \boldsymbol{z} - \boldsymbol{z}^t \right\rangle, \qquad (12)$$

In our experiments, we refer to this implementation as the *accelerated multi-cut* approach.

3. Upper Bounds in Stochastic Cutting Planes with Binary Variables

In this section, we analyze the upper bounds obtained at each iteration of our cutting-plane methods and design convergence criteria that allow us to terminate our methods with confidence.

The primary motivation for this section is that while the lower bounds for the three stochastic cutting plane methods introduced in Section 2 are deterministic, their per iteration estimates of the cost associated with each incumbent solution $\boldsymbol{z}^t$,

$$\langle \boldsymbol{c}, \boldsymbol{z}^t \rangle + \frac{1}{|\mathcal{R}_t|} \sum_{r \in \mathcal{R}_t} f(\boldsymbol{z}^t; \boldsymbol{d}^{\cdot,r}) \qquad (13)$$

are stochastic estimates that depend on the sample $\mathcal{R}_t$. Accordingly, we cannot simply use these stochastic estimates in the same way as in a deterministic method and terminate when the deterministic lower bound, say $\langle \boldsymbol{c}, \boldsymbol{z}^t \rangle + \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \eta_r$ in the multi-cut case, is within ϵ of our stochastic upper bound, or we may terminate because $\boldsymbol{z}^t$ is a high variance solution and we picked an optimistic sample set $\mathcal{R}_t$, rather than because $\boldsymbol{z}^t$ is an optimal solution; see also Smith and Winkler (2006).

In addition, another salient characteristic of our problem is that the decision variables $\boldsymbol{z}$ are binary. Hence, as described in pseudo-code in Algorithm 1 and 2, a MIO problem needs to be solved at each iteration by constructing a branch-and-bound tree (multi-tree implementation). Nowadays, efficient implementations of these schemes exist that simultaneously construct the branch-and-bound tree and generate cutting planes (single-tree implementation). We also discuss the extent to which the stochastic cutting-plane algorithms we developed in the previous section can be implemented with a single-tree instead of multi-tree approach.

3.1. Convergence Criteria

In this section, we define a convergence criterion by using an asymptotically normal estimator of the upper bound and using a related upper confidence bound. Suppose that one of our stochastic cutting-plane methods finds a solution $\boldsymbol{z}$, and that we would like to evaluate its quality. Then, we can use a sample $\mathcal{W}$ to estimate the true cost of this solution

$$\bar{c} = \langle \boldsymbol{c}, \boldsymbol{z} \rangle + \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} f(\boldsymbol{z}; \boldsymbol{d}^{\cdot,r})$$

by its estimate on $\mathcal{W}$:

$$\hat{c}^{\mathcal{W}} = \langle \boldsymbol{c}, \boldsymbol{z} \rangle + \frac{1}{|\mathcal{W}|} \sum_{r \in \mathcal{W}} f(\boldsymbol{z}; \boldsymbol{d}^{\cdot,r}).$$

In this section, for simplicity, we omit the dependency of $\hat{c}^{\mathcal{W}}$, $\bar{c}$, and the following quantities, on the solution $\boldsymbol{z}$. We also denote $\mathcal{W}$ the random sample used for termination since it could be a new independent draw from the sample $\mathcal{R}_t$ used in the algorithm (and should be, for our estimation procedure to be unbiased).

As noted by Morton (1998), Mak et al. (1999), under some mild assumptions on the distribution of $\boldsymbol{d}^k$ (e.g., finite variance), for an infinite number of scenarios $|\mathcal{R}|$, this estimator obeys a central limit theorem:

$$\sqrt{|\mathcal{W}|}\, [\hat{c}^{\mathcal{W}} - \bar{c}] \xrightarrow{d} \mathcal{N}(0, \sigma_c^2) \text{ as } |\mathcal{W}| \to \infty,$$

where $\sigma_c^2 = \mathrm{Var}(f(\boldsymbol{z}, d^{\cdot r}))$ can be estimated via the sample variance estimator

$$\hat{\sigma}_c^2 := \frac{1}{|\mathcal{W}| - 1} \sum_{r \in \mathcal{W}} \left(f(\boldsymbol{z}, \boldsymbol{d}^{\cdot r}) - \tfrac{1}{|\mathcal{W}|} \sum_{s \in \mathcal{W}} f(\boldsymbol{z}; \boldsymbol{d}^{\cdot,s}) \right)^2 .$$

In reality, however, we only have finitely many observations $\mathcal{R}$. Yet, provided $|\mathcal{R}|$ is large relative to $|\mathcal{W}|$, we can still apply the CLT to estimate the cost of $\boldsymbol{z}$. Consequently, letting q_α be such that $\mathbb{P}(\mathcal{N}(0,1) \leq q_\alpha) = 1 - \alpha$, we can construct an asymptotically valid confidence interval for this estimator at level α of the form

$$\left[\hat{c}^{\mathcal{W}} - \frac{q_{\alpha/2}}{\sqrt{|\mathcal{W}|}}\hat{\sigma}_c, \hat{c}^{\mathcal{W}} + \frac{q_{\alpha/2}}{\sqrt{|\mathcal{W}|}}\hat{\sigma}_c\right].$$

We terminate our method using a modified version of the convergence criteria proposed by Morton (1998). Namely, letting

$$\bar{c}_{\alpha,t} := \hat{c}^{\mathcal{W}} + \frac{q_{\alpha/2}}{\sqrt{|\mathcal{W}|}}\hat{\sigma}$$

denote an upper confidence bound at level α on the cost of $\boldsymbol{z}^t$, the solution generated at the tth iterate of one of our cutting-plane methods, we terminate as soon the conservative bound gap falls below a predefined threshold ϵ, i.e., for the multi-cut method

$$\frac{\bar{c}_{\alpha,t} - \left(\langle \boldsymbol{c}, \boldsymbol{z}^t\rangle + \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \eta_{r,t}\right)}{\bar{c}_{\alpha,t}} \leq \epsilon \tag{14}$$

and the termination criteria for the two remaining methods are similar. Alternatively, we terminate if we exceed a time limit, as discussed in our numerical results. In the latter case, we evaluate the true cost of $\boldsymbol{z}^t$ by computing its cost across each scenario in $\mathcal{R}$.

We remark that for some adversarial instances of Problem (1), using the same sample size at each iteration in conjunction with this termination criterion could lead to unattractive results where we terminate at a highly suboptimal solution with high probability (c.f. Morton 1998, Example 1). To address this issue and provide a confidence bound on our overall solution (accounting for multiple testing problems), we can increase the sample size at each iteration of the method in accordance with Morton (1998, Theorem 2) or use another sampling rule discussed therein (see also Bayraksan and Morton 2011). However, owing to the single-tree implementation of our cutting-plane methods, as discussed in the next section, we do not test every candidate solution we generate when deciding to terminate. Therefore, as we observe in our numerical results, using the same sample size at each iteration is usually adequate. This is particularly true for the single-cut and τ-cut methods, which, as discussed previously, often generate conservative lower bounds in practice, meaning that we often terminate at a computational time limit.

Finally, in circumstances where the total number of scenarios is relatively small, we can evaluate the true upper bound directly, rather than a stochastic estimate of the bound. Accordingly, we take this approach whenever the number of scenarios is sufficiently small.

3.2. Integrating Optimality Cuts Within a Branch-and-Cut Framework

Once our cut-generation and termination criterion schemes have been designed, they need to be embedded within a branch-and-cut framework to solve Problem (1) to certifiable optimality. Indeed, in the naive implementation of our algorithms described in the pseudocode of the previous section, we need to solve a mixed-integer problem at each iteration. For further scalability benefits, we can integrate our stochastic cut generation procedure within a state-of-the-art commercial mixed-integer solver (namely, `Gurobi` version 9.1.2) using `lazy constraint callbacks`, which accelerate cutting-plane methods by constructing a single branch-and-bound tree. For example, they have been used to implement deterministic cutting-plane algorithms in a highly efficient and relatively standard way; see, e.g., Fischetti et al. (2017, Section 4).

Mixed-integer solvers assume that `lazy constraints` are binding at the point they are generated. Accordingly, they do not visit and do not generate `lazy constraints` twice at the same solution. Our stochastic cuts, however, are not binding, they provide a valid yet not necessarily tight lower bound. Therefore, when we implement our method with `lazy constraints`, the MIO solver can terminate with a highly suboptimal solution it deems optimal, because it (mistakenly) assumes the value of the cut generated at $\boldsymbol{z}^t$ and evaluated at that point is precisely the cost of $\boldsymbol{z}^t$. To avoid this issue, we take a hybrid approach between single- and multi-tree branch-and-cut, which, to our knowledge, has not yet been described in the literature.

Namely, we maintain an outer loop where, at each iteration, we run a single-tree implementation of branch-and-cut with stochastic cutting-planes. We save all the cuts generated and imposed as `lazy constraints` within a separate cut pool during the branch-and-cut algorithm. After the branch-and-cut algorithm, we randomly sample a subset of scenarios $\mathcal{W}$ and compute the termination criterion described in the previous section to determine whether the solution returned by the branch-and-cut algorithm is indeed ϵ-optimal with high probability ($\alpha = 0.90$). By computing this convergence criterion at each iteration of the outer loop only, we mitigate the issue of multiple hypothesis testing that would arise when testing the quality of a solution at each iteration of Algorithm 1–2 (inner loop). If the convergence criterion is met, we terminate the algorithm. Otherwise, we rerun the branch-and-cut algorithm and ensure the MIO solver no longer considers the previously generated `lazy constraints` as binding: We apply the constraints generated in the lazy cut pool as regular linear constraints, purge the lazy cut pool, and rerun the branch-and-cut algorithm. In addition to an optimality gap criterion, we terminate the algorithm when the total computational time exceeds a predefined `TimeLimit`. We summarize this procedure in Algorithm 3. We remark that this approach is related to the notion of restarting a single-tree decomposition in a classical deterministic Benders scheme (see, e.g., Fischetti et al. 2016, Section 4.4).

Algorithm 3 Outer Loop for Stochastic Branch-and-Cut

```
initialize $CutPool = \emptyset$, $t = 0$
repeat
    Increment $t \leftarrow t + 1$
    Initialize Algorithm 1/2 with constraints in $CutPool$.
    Run lazy-constraint implementation of Algorithm 1/2
    Save all lazy constraints generated in $CutPool$.
    Obtain candidate optimal solution $\boldsymbol{z}^t$.
    Obtain valid lower bound from the MIO branch-and-cut solver.
    Sample $\mathcal{W}$ and compute $\bar{c}_{\alpha,t}$
until (14) or TimeLimit
Return $\boldsymbol{z}^t$, stochastic upper bound, and deterministic lower bound
```

Finally, in addition to the hybrid scheme described in this section, one could also consider a pure multi-tree implementation of our stochastic cutting-plane methods, as suggested in Section 2 and the classical network design literature (Geoffrion and Graves 1974). However, in preliminary numerical experiments, we found that such an approach is significantly slower because it involves solving a different MIO to generate each cut. Accordingly, we do not consider such an approach as part of our numerical experiments.

3.3. Accelerating the Convergence of our Approach

We now describe practical enhancements to our stochastic cutting-plane approaches that improve their convergence, sometimes substantially; see also Fischetti et al. (2016, 2017), Bertsimas et al. (2021) for related discussions on accelerating the convergence of decomposition schemes. To facilitate a fair comparison, we implement these strategies for all Benders-type methods in our experiments.

Warm-Starting the Lower Bound: Cuts at the Root Node First, we can warm-start our lower bound by applying cutting planes at the root node obtained after solving a Boolean relaxation of (3) using a continuous analog of our discrete cutting-plane method. This strategy is referred to a two-phase Benders approach (McDaniel and Devine 1977) and has been successfully been applied in network design (Crainic et al. 2016) and other contexts (e.g., Fischetti et al. 2017, Bertsimas et al. 2021). Note that the continuous cutting plane algorithm can also be implemented in a multi- or single-cut fashion and in a deterministic or stochastic version. To balance the tightness of the formulation at the root node against the overall computation cost, we impose a hard constraint on the total number of root node cuts applied (typically 10 or 20).

Warm-Starting the Upper Bound: We supply the initial network (without any new construction) and the network obtained by constructing all the edges as warm-starts. However, we do not implement a more sophisticated warm-starting strategy for any of our methods, to better isolate the numerical benefit of our decomposition schemes. We remark that, in practice, the Boolean relaxation could be randomly rounded to generate provably high-quality feasible solutions (c.f. Bertsimas et al. 2021, Section 3.2), and other heuristics specific to network design, as reviewed in the introduction, could also be applied.

Warm-Starting Feasibility Constraints: Problem-specific inequalities can be added to provide more structure to the master problem (Rahmaniani et al. 2018). Based on the numerical evidence of Rahmaniani et al. (2018), we implement two types of valid inequalities (origin and destination node inequalities and network connectivity cuts). We also implement partial optimality cuts: Namely, when the incumbent solution $\boldsymbol{z}^T$ is infeasible, we not only impose feasibility cuts of the form (5) for scenarios $r \in \mathcal{R}_{\texttt{inf}}$, we also derive optimality cuts for scenarios $r \in \mathcal{R} \setminus \mathcal{R}_{\texttt{inf}}$ (or $\mathcal{R}_T \setminus \mathcal{R}_{\texttt{inf}}$ in the stochastic version). In a multi-cut implementation, we can impose these constraints for each η_r. In the single-cut implementation, we use our dual averaging technique to derive a valid linear inequality on the single epigraph variable η.

We remark that, on preliminary experiments (Table OS.5.4), we found no clear benefit from using Pareto-optimal cuts (Magnanti and Wong 1981). Accordingly, we did not consider them in our implementation. Note that this finding is consistent with prior literature on Pareto-optimal cuts, which finds that they do more harm than good (c.f. Papadakos 2008, Fischetti et al. 2017).

4. Numerical Experiments

In this section, we numerically benchmark our stochastic Benders decomposition schemes on data-driven MCFND problems. We also compare their performance with their deterministic counterparts and `Gurobi` on a perspective reformulation of the original MIO formulation (1). Our code is available on the article's GitHub repository (Bertsimas et al. 2024).

4.1. Implementation Details

All experiments were conducted on MIT's Supercloud Cluster (Reuther et al. 2018), which hosts Intel Xeon Platinum 8260 processors. All algorithms were implemented in Julia v1.7.3 (Bezanson et al. 2017) using JuMP v0.21.10 (Dunning et al. 2017) and Gurobi v9.5.1 (Gurobi Optimization, LLC 2022). The RAM allocated varies from 4GB to 176GB for the largest instances, see OS.5.2 for a detailed breakdown.

In Section 4.2 and 4.3, we consider synthetic instances generated according to a methodology from Günlük and Linderoth (2009) and Bertsimas et al. (2021). In particular, in these instances, the

Table 1 Dimensions of the MCFND problems generated, by scale (small-, medium-, and large-scale).

Scale	$\lvert\mathcal{N}\rvert$		$\lvert\mathcal{K}\rvert$		$\lvert\mathcal{R}\rvert$
Small	{10,30,50,70}	×	{5,10,25,50}	×	{10,30,50,70,100}
Medium	{100,150,200}	×	{5,10,25,50}	×	{10,30,50,70,100}
Large	{300,500,700}	×	{5,10,25,50}	×	{10,30,50,70,100}

network flow problem for each commodity corresponds to an all-to-one shortest path, and feasibility is not an issue (the pre-existing edges are sufficient to guarantee feasibility). All in all, we generate instances with varying numbers of nodes $|\mathcal{N}|$, commodities $|\mathcal{K}|$, and scenarios $|\mathcal{R}|$, as described in Table 1. We later refer to these instances as small-, medium-, and large-scale instances based on the number of nodes $|\mathcal{N}|$. In Section 4.4, we evaluate the experiments on the **R** instances from Crainic et al. (2016), with demand scenarios generated by Rahmaniani et al. (2018). Details on generating the synthetic and **R** instances are provided in Online Supplement OS.5.1.

For our algorithms, we use two termination criteria: a time limit (7,200 seconds) and an optimality gap target $\epsilon = 1\%$ (with $\alpha = 0.90$ for our stochastic algorithms). Note that the time limit applies to the full outer-loop presented in Algorithm 3 (and not on each run of the branch-and-cut algorithm only). For all stochastic methods, we use a sampling rate, $|\mathcal{R}_t|/|\mathcal{R}|$, of 10%. We also fix the regularization parameter γ to 1 —we discuss its impact on our algorithms in Online Supplement OS.4. We warm-start all methods with the original connected graph as an initial solution.

4.2. Comparison of Different Stochastic Cutting-Plane Algorithms

In this section, we benchmark the variants of the stochastic cutting-plane algorithm proposed in Section 2, namely the multi-, single-, accelerated multi-cut, in terms of their ability to obtain a certifiably near-optimal solution with high confidence. We also measure the impact of warm-starting these methods with cuts obtained from solving the perspective relaxation with a multi- or single-cut stochastic cutting-plane algorithm, and applying these cuts at the root node in our branch-and-cut scheme (which we refer to as multi-cut or single-cut root node cuts respectively). We report average computational time (capped at 7,200 seconds) for solving our small and medium-scale synthetic instances in Table 2. To augment these results, Table 3 reports the average optimality gap at termination, and Table OS.5.3 (see Online Supplement OS.5.3) reports the fraction of instances solved within the time limit. Note that the optimality gaps reported in Table 3 are computed using the true cost of the incumbent solution, using all scenarios in $\mathcal{R}$, and that the time required to calculate this true cost is not included in the computational time of any cutting-plane algorithm.

We observe that the multi-cut and single-cut warm-start strategies both effectively reduce the relative optimality gap at termination. Indeed, our root node strategies more than halve the optimality gap at termination compared to not applying cuts at the root node. For the multi- and single-cut

Table 2 **Computational time (in seconds) of the multi-, single-, and accelerated multi-cut stochastic cutting plane algorithm, with different warm-start strategies at the root node (none, multi-cut, and single-cut root node cuts). Metrics are averaged across instances with the same number of nodes $|\mathcal{N}|$.**

	Multi-Cut			Single-Cut			Accelerated Multi-Cut		
$\lvert\mathcal{N}\rvert$	None	Multi	Single	None	Multi	Single	None	Multi	Single
10	384.87	107.11	89.25	142.96	76.89	85.27	542.76	124.92	94.08
30	5760.18	5763.62	5512.78	5166.12	4018.97	4414.57	6262.14	5951.10	6409.78
50	7200.00	6588.54	7200.00	5154.68	4514.63	4426.90	6625.89	6865.11	7200.00
70	6788.22	7200.00	7200.00	5861.50	5248.96	5132.34	7200.00	7200.00	7200.00
100	7105.08	7200.00	7157.26	6117.74	5490.28	4775.17	7200.00	7200.00	7103.43
150	7200.00	7200.00	7200.00	6531.33	5819.10	5992.10	7200.00	7200.00	6865.95
200	7200.00	7200.00	7200.00	6571.25	6430.63	5625.82	7200.00	7194.36	7200.00

Table 3 **Relative optimality gap (in %) at termination for multi-, single-, and accelerated multi-cut algorithms, with different warm-start strategies at the root node. Metrics are averaged across instances with same number of nodes $|\mathcal{N}|$.**

	Multi-Cut			Single-Cut			Accelerated Multi-Cut		
$\lvert\mathcal{N}\rvert$	None	Multi	Single	None	Multi	Single	None	Multi	Single
10	0.26	0.02	0.93	0.10	0.23	0.23	0.12	0.07	0.93
30	10.36	4.21	3.85	14.25	5.27	4.30	18.48	4.47	3.88
50	15.19	2.69	2.21	10.56	4.33	3.72	26.32	2.37	2.32
70	48.23	11.78	8.17	24.78	16.58	12.12	52.00	7.28	8.46
100	50.55	8.90	4.44	29.73	18.51	11.35	49.75	6.99	4.56
150	61.71	11.35	9.19	53.03	22.68	12.80	58.06	5.51	9.93
200	63.49	12.49	10.43	47.34	20.70	12.80	60.51	7.15	12.01

approach, a single-cut strategy at the root node appears to outperform a multi-cut root node strategy in terms of the relative gap at termination. For the accelerated multi-cut approach, however, both root node strategies are comparable, with a small edge for multi-cut. All in all, applying a single-cut approach warm-started with a single-cut method at the root node performs best in terms of computational time, while the accelerated multi-cut approach warm-started with a multi-cut method at the root node achieves the lowest average gap at termination. For this reason, we only report results for these two variants in the following two sections.

Next, we investigate the number of iterations of the outer loop performed by our methods; recall that in Section 3, we propose an outer loop procedure that allows our sampling approach to be safely integrated within a branch-and-cut procedure, without requiring a new branch-and-bound tree each time we generate a cut. To this end, Figure 1 depicts the number of outer-loop iterations performed by our single-cut algorithm on the small- and medium-scale instances; recall that these instances are

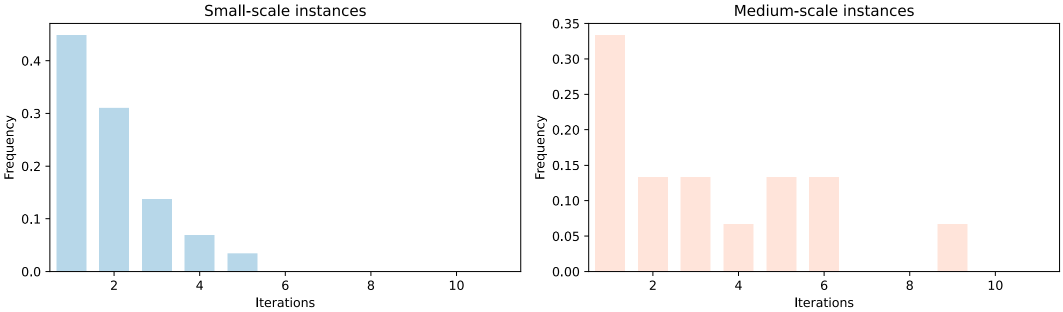


Figure 1 **Distribution of the number of outer-loop iterations required by the single-stochastic cutting-plane algorithms with single-cut root node cuts on small-scale (left panel) and medium-scale (right panel) instances; see Table 1 for definitions of small and medium-scale instances.**

defined in Table 1, and comprise instances with 10–70 and 100–200 nodes respectively. We observe that only one outer loop iteration is performed in many cases (50% for small-scale and 70% for medium-scale instances). In the remaining cases, the first iteration of branch-and-cut with stochastic cuts terminates with a solution that is not ϵ-optimal but Algorithm 3 is very efficient, requiring a limited number of additional iterations to identify an optimal solution. This verifies that a single outer loop iteration often wrongly terminates at a solution that is not optimal. On the other hand, only a small number of iterations of the outer loop are usually needed to achieve optimality. Therefore, the tractability of our approach is not compromised by the outer loop. We remind the reader that we impose a global time limit of two hours and Algorithm 3 terminates when it either converges or reaches this time limit. Accordingly, the results on small-scale instances may be less right-censored.

4.3. Benchmarking Scalability on Synthetic Instances

We now compare the performance of our stochastic cutting plane methods (single- and accelerated multi-cut) against two benchmarks: (a) solving Problem (1)'s perspective reformulation directly with `Gurobi`, (b) a deterministic single-cut method with single-cut root node cuts (we also report the performance of the deterministic method with several acceleration strategies from the literature in Table OS.5.4 in Online Supplement OS.5.4). For our stochastic approaches, we use a sampling rate of 10%. We impose a time limit of 7,200 seconds for all methods. To calibrate our approaches and verify their correctness, we use the smallest instances to verify that all methods terminate with the same optimal solution (see Table OS.5.5 in Online Supplement OS.5.4).

We report the average computational time and optimality gap of all methods, on the small-, medium-, and large-scale instances, in Table 4, with metrics averaged over instances with the same number of nodes $|\mathcal{N}|$.

Table 4 Runtime (in seconds) and final optimality gap (in %) for each algorithm, averaged over instances with the same number of nodes $|\mathcal{N}|$.

	Gurobi with (1)		Deterministic		Stochastic			
					Single		Accerated Multi	
$\lvert\mathcal{N}\rvert$	Runtime	Gap	Runtime	Gap	Runtime	Gap	Runtime	Gap
10	223.60	0.00	247.79	0.02	85.27	0.23	124.92	0.07
30	7200.00	42.68	7163.94	6.22	4414.57	4.30	5951.10	4.47
50	7200.00	67.71	7200.00	4.87	4426.90	3.72	6865.11	2.37
70	7200.00	77.56	7200.00	11.85	5132.34	12.12	7200.00	7.28
100	7200.00	85.15	7165.78	16.37	4775.17	11.35	7200.00	6.99
150	7200.00	95.97	7186.61	23.49	5992.10	12.80	7200.00	5.51
200	7196.63	92.87	6853.71	26.68	5625.82	12.80	7194.36	7.15
300	-	-	6237.87	23.11	6017.72	11.04	7200.00	11.04
500	-	-	6441.49	49.09	6321.34	26.77	7200.00	21.90
700	-	-	6499.08	53.39	6295.69	39.53	7200.00	30.72

We observe that a perspective reformulation of the original formulation (1) cannot be solved by Gurobi with 100 or more nodes within the time (2 hours) and memory ($>$ 72GB) limits. Indeed, while this approach converges within minutes for instances with ten nodes, it fails to identify an optimal solution within the two-hour time limit for instances with 20-70 nodes and terminates with large optimality gaps ($>30\%$) on average. On the other hand, a deterministic Benders decomposition scheme reaches optimality gaps that are an order of magnitude smaller on instances with 20-70 nodes, scales to instances with up to 200 nodes, but fails to recover a solution with a meaningful optimality gap within the time limit for larger problems.

Our stochastic cutting plane algorithms significantly improve upon their deterministic counterpart. On small- and medium-scale instances, the single-cut stochastic cutting-plane algorithm reduces the average computational time by 40-90% on the small instances and 20–50% on the medium ones. A comparison in terms of average computational times might be misleading, however, because of the time limit, and because many of these instances are not solved to ϵ-optimality. Accordingly, we also compare in terms of the optimality gap. We observe that our single-cut stochastic cutting-plane algorithm terminate with gaps half the size of deterministic algorithms on medium to large instances (i.e., around 5% for the instances with 10–50 nodes, 12% for the instances with 70–300 nodes, and 30–40% for the largest instances compared with 5%, 12–23%, and 50% for the deterministic approach respectively). Finally, we observe that our accelerated multi-cut stochastic cutting-plane algorithm is generally slower than its single-cut counterpart (probably due to the increased number of epigraph variables) but achieves even lower optimality gaps at termination: less than 5% for 10–50 nodes, 7–11% for 70–300 nodes, and 20–30% for 400-700 nodes. Figures OS.5.1-OS.5.2 in Online Supplement OS.5.4 display the optimality gap achieved by both our methods for each value of $|\mathcal{N}|$, $|\mathcal{R}|$, and $|\mathcal{K}|$, and shows that our methods are most sensitive to the number of commodities and nodes.

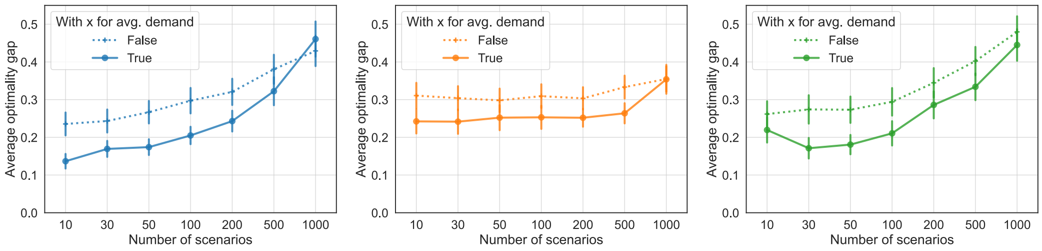
4.4. Benchmarking on the R Instances

We now benchmark our methods on network design instances from the literature, the so-called **R** instances, originally introduced by Crainic et al. (2000) for deterministic network design problems. We use a total 54 instances (see details in Online Supplement OS.5.1) with 10–20 nodes and 10–50 commodities. So, compared to our synthetic instances, the ratio $|\mathcal{K}|/|\mathcal{N}|$ is higher for these instances. The number of scenarios generated varies from 10 to 1,000. In line with our results in the previous section, we evaluate the performance of our single-cut and accelerated multi-cut stochastic cutting plane algorithm with a 10% sampling rate and a time limit of 7,200 seconds for all methods.

Compared with the synthetic instances considered in the previous sections, each commodity in these instances has one origin and one destination. In addition, we start from a network without any edge, which is challenging for decomposition schemes because they do not have access to second-stage variables $\boldsymbol{x}^k$ to ensure primal feasibility of $\boldsymbol{z}$. In this setting, the separation oracle in the (deterministic or stochastic) cutting-plane algorithm generates either a feasibility or an optimality cut, depending on the feasibility of the incumbent solution $\boldsymbol{z}^t$, as described in Section 2.3. We also implemented the strategies presented in Section 3.3.

Furthermore, a necessary condition for feasibility is to be feasible for one particular scenario. In particular, we consider adding one second-stage variable $\boldsymbol{x}$ to the master problem, to enforce feasibility with respect to the average demand. As displayed in Figure 2, we find that this simple strategy effectively reduces the optimality gap achieved by all decomposition schemes. However, the benefit shrinks as the total number of scenarios increases, which suggests that more than one second-stage variable might be needed to achieve the same gain in these instances.

Figure 3 compares the optimality gap achieved at termination (with a 2-hour time limit) for the three cutting-plane algorithms. As expected, we observe that our stochastic cutting-plane algorithm achieves smaller optimality gaps than its deterministic counterpart as the total number of scenarios increases. Compared to the results on synthetic instances, however, we observe that the deterministic cutting-plane approach outperforms the single-cut stochastic one when the number of scenarios is smaller. We believe this behavior could be explained by the fact that the **R** instances are small ($\mathcal{N} = 10$–20 nodes), a regime where the different cutting-plane algorithms were achieving comparable optimality gaps in Table 4, and the fact the deterministic implementation (computes and) imposes more constraints per iteration, which might be more valuable when feasibility constraints are needed, especially in the first iterations of the algorithm.



(a) Deterministic (b) Stochastic Single (c) Stochastic Acc. Multi

Figure 2 **Impact of implementing a cutting-plane algorithm with one second-stage variable x in the master problem (to ensure at least feasibility for the average demand) on the average optimality gap achieved on the R instances, as the number of scenarios $|\mathcal{R}|$ increases. Bars represent standard errors.**

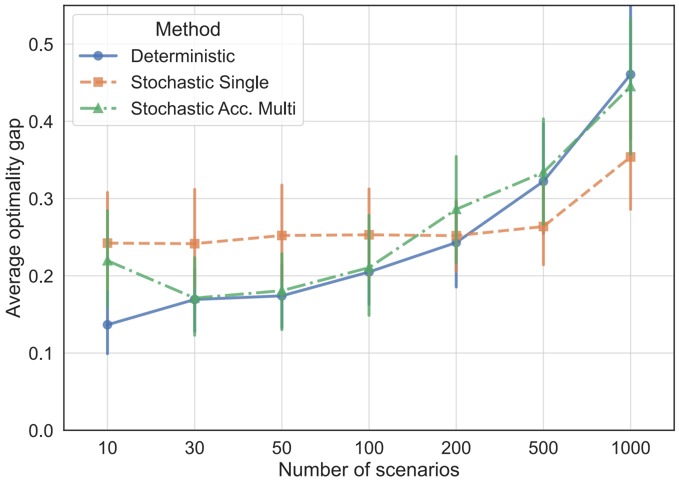


Figure 3 **Average optimality gap achieved on the R instances by the deterministic and our stochastic (single-cut and accelerated multi-cut) cutting-plane algorithms, for different number of scenarios $|\mathcal{R}|$. Bars represent standard errors.**

In Online Supplement OS.5.5, we report the average gap achieved by the naive formulation solved with Gurobi (Figure OS.5.3), as well as the distribution of optimality gaps achieved by each method (Figure OS.5.4).

5. Conclusion

We propose a stochastic Benders decomposition scheme which solves large-scale stochastic network design problems. Our approach mitigates the high computational cost of generating each cut by sampling a subset of the data at each iteration, while applying a dual-averaging technique to ensure that the cuts generated remain valid for the original problem. We also propose an outer loop technique to ensure the safe termination of our algorithm when the Benders decomposition scheme is implemented via lazy callbacks. To our knowledge, this is the first work synthesizing sampling with a single-tree approach for generating Benders cuts. We consider multi- and single-cut variants of our algorithm (and τ-cut in Online Supplement) and discuss its implementation within a branch-and-cut solver. In numerical experiments, we demonstrate that our stochastic decomposition schemes obtain optimality gaps of 5–7% on instances with 100–200 nodes, compared to 16–26% for deterministic Benders schemes. Moreover, we obtain bound gaps of around 30% on instances with up to 700 nodes and 50 commodities, i.e., problem sizes an order of magnitude larger than any instances addressed by exact methods in the literature. Beyond network design, we believe our approach could be applied to other two-stage stochastic optimization problems addressed via sample average approximations.

Acknowledgments

We thank two anonymous referees for valuable comments that improved the manuscript. Ryan Cory-Wright gratefully acknowledges the MIT-IBM Research Lab for hosting him while part of this work was conducted.

References

Agrawal A, Klein P, Ravi R (1991) When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *Proceedings of ACM Symposium on Theory of Computing*, 134–144.

Atamtürk A, Günlük O (2018) A note on capacity models for network design. *Operations Research Letters* 46(4):414–417.

Atamtürk A, Günlük O (2021) Multicommodity multifacility network design. *Network Design with Applications to Transportation and Logistics*, 141–166 (Springer).

Balakrishnan A, Magnanti TL, Shulman A, Wong RT (1991) Models for planning capacity expansion in local access telecommunication networks. *Annals of Operations Research* 33(4):237–284.

Barnhart C, Belobaba P, Odoni AR (2003) Applications of operations research in the air transport industry. *Transportation Science* 37(4):368–391.

Bayraksan G, Morton DP (2011) A sequential sampling procedure for stochastic programming. *Operations Research* 59(4):898–913.

Beale EM (1955) On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society: Series B (Methodological)* 17(2):173–184.

Bertsimas D, Cory-Wright R, Pauphilet J (2021) A unified approach to mixed-integer optimization problems with logical constraints. *SIAM Journal on Optimization* 31(3):2340–2367.

Bertsimas D, Cory-Wright R, Pauphilet J, Petridis P (2024) A Stochastic Benders Decomposition Scheme for Large-Scale Stochastic Network Design. URL `http://dx.doi.org/10.1287/ijoc.2023.0074.cd`, available for download at https://github.com/INFORMSJoC/2023.0074.

Bertsimas D, Li ML (2022) Stochastic cutting planes for data-driven optimization. *INFORMS Journal on Computing* 34(5):2400–2409.

Bertsimas D, Teo CP (1998) From valid inequalities to heuristics: A unified view of primal-dual approximation algorithms in covering problems. *Operations Research* 46(4):503–514.

Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization*, volume 6 (Athena Scientific Belmont, MA).

Bezanson J, Edelman A, Karpinski S, Shah VB (2017) Julia: A fresh approach to numerical computing. *SIAM Review* 59(1):65–98.

Bienstock D, Chopra S, Günlük O, Tsai CY (1998) Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming* 81(2):177–199.

Binato S, Pereira MVF, Granville S (2001) A new Benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems* 16(2):235–240.

Birge JR, Louveaux F (2011) *Introduction to Stochastic Programming.* Springer Series in Operations Research and Financial Engineering (New York, NY: Springer New York).

Birge JR, Louveaux FV (1988) A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research* 34(3):384–392.

Bixby RE (2012) A brief history of linear and mixed-integer programming computation. *Documenta Mathematica* 2012:107–121.

Boyce DE, Farhi A, Weischedel R (1973) Optimal network problem: a branch-and-bound algorithm. *Environment and Planning A* 5(4):519–533.

Ceria S, Soares J (1999) Convex programming for disjunctive convex optimization. *Mathematical Programming* 86(3):595–614.

Contreras I, Cordeau JF, Laporte G (2011) Benders decomposition for large-scale uncapacitated hub location. *Operations Research* 59(6):1477–1490.

Cornuejols G, Nemhauser GL, Wolsey LA (1980) A canonical representation of simple plant location problems and its applications. *SIAM Journal on Algebraic Discrete Methods* 1(3):261–272.

Costa AM (2005) A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research* 32(6):1429–1450.

Crainic TG, Gendreau M, Farvolden JM (2000) A simplex-based Tabu search method for capacitated network design. *INFORMS Journal on Computing* 12(3):223–236.

Crainic TG, Gendreau M, Gendron B, eds. (2021a) *Network Design with Applications to Transportation and Logistics* (Cham: Springer International Publishing).

Crainic TG, Hewitt M, Maggioni F, Rei W (2021b) Partial Benders decomposition: general methodology and application to stochastic network design. *Transportation Science* 55(2):414–435.

Crainic TG, Rei W, Hewitt M, Maggioni F (2016) *Partial Benders Decomposition Strategies for Two-Stage Stochastic Integer Programs*, volume 37 (CIRRELT).

Dantzig GB (1955) Linear programming under uncertainty. *Management Science* 1(3-4):197–206.

Dantzig GB, Infanger G (1993) Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research* 45(1):59–76.

Davis D, Drusvyatskiy D, Kakade S, Lee JD (2020) Stochastic subgradient method converges on tame functions. *Foundations of Computational Msathematics* 20(1):119–154.

de Camargo RS, Miranda Jr G, Luna HP (2008) Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research* 35(4):1047–1064.

De Matos VL, Philpott AB, Finardi EC (2015) Improving the performance of stochastic dual dynamic programming. *Journal of Computational and Applied Mathematics* 290:196–208.

Dunning I, Huchette J, Lubin M (2017) JuMP: A modeling language for mathematical optimization. *SIAM Review* 59(2):295–320.

Erickson RE, Monma CL, Veinott Jr AF (1987) Send-and-split method for minimum-concave-cost network flows. *Mathematics of Operations Research* 12(4):634–664.

Fábián CI (2000) Bundle-type methods for inexact data. *Central European Journal of Operations Research* 8(1):35–55.

Fischetti M, Ljubić I, Sinnl M (2016) Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research* 253(3):557–569.

Fischetti M, Ljubić I, Sinnl M (2017) Redesigning Benders Decomposition for Large-Scale Facility Location. *Management Science* 63(7):2146–2162.

Florian M, Bushell G, Ferland J, Guerin G, Nastansky L (1976) The engine scheduling problem in a railway network. *INFOR: Information Systems and Operational Research* 14(2):121–138.

Garey MR, Johnson DS (1977) The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics* 32(4):826–834.

Gendron B, Crainic TG, Frangioni A (1999) Multicommodity capacitated network design. *Telecommunications Network Planning*, 1–19 (Springer).

Gendron B, Hanafi S, Todosijević R (2018) Matheuristics based on iterative linear programming and slope scaling for multicommodity capacitated fixed charge network design. *European Journal of Operational Research* 268(1):70–81.

Geoffrion AM (1972) Generalized Benders decomposition. *Journal of Optimization Theory and Applications* 10(4):237–260.

Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by Benders decomposition. *Management Science* 20(5):822–844.

Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. *Management Science* 22(4):455–460.

Goemans MX, Bertsimas DJ (1993) Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming* 60(1):145–166.

Grimmett G, Stirzaker D (2020) *Probability and Random Processes* (Oxford university press).

Guigues V (2020) Inexact cuts in stochastic dual dynamic programming. *SIAM Journal on Optimization* 30(1):407–438.

Günlük O (1999) A branch-and-cut algorithm for capacitated network design problems. *Mathematical Programming* 86(1):17–39.

Günlük O, Linderoth J (2009) Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical Programming* 183–205.

Gurobi Optimization, LLC (2022) Gurobi Optimizer Reference Manual.

Higle JL, Sen S (1991) Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research* 16(3):650–669.

Higle JL, Sen S (1996) Duality and statistical tests of optimality for two stage stochastic programs. *Mathematical Programming* 75(2):257–275.

Infanger G (1992) Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs. *Annals of Operations Research* 39(1):69–95.

Ke Q, Ferrara E, Radicchi F, Flammini A (2015) Defining and identifying sleeping beauties in science. *Proceedings of the National Academy of Sciences* 112(24):7426–7431.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Magnanti TL, Mirchandani P, Vachani R (1993) The convex hull of two core capacitated network design problems. *Mathematical Programming* 60(1):233–250.

Magnanti TL, Mirchandani P, Vachani R (1995) Modeling and solving the two-facility capacitated network loading problem. *Operations Research* 43(1):142–157.

Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* 29(3):464–484.

Magnanti TL, Wong RT (1984) Network design and transportation planning: models and algorithms. *Transportation Science* 18(1):1–55.

Mak WK, Morton DP, Wood RK (1999) Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* 24(1-2):47–56.

McDaniel D, Devine M (1977) A modified Benders' partitioning algorithm for mixed integer programming. *Management Science* 24(3):312–319.

Morton DP (1998) Stopping rules for a class of sampling-based stochastic programming algorithms. *Operations Research* 46(5):710–718.

Papadakos N (2008) Practical enhancements to the Magnanti–Wong method. *Operations Research Letters* 36(4):444–449.

Pereira MV, Pinto LM (1991) Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming* 52(1):359–375.

Pishvaee MS, Razmi J, Torabi SA (2014) An accelerated Benders decomposition algorithm for sustainable supply chain network design under uncertainty: A case study of medical needle and syringe supply chain. *Transportation Research Part E: Logistics and Transportation Review* 67:14–38.

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2018) Accelerating the Benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization* 28(1):875–903.

Ramírez-Pico C, Ljubić I, Moreno E (2023) Benders adaptive-cuts method for two-stage stochastic programs. *Transportation Science* 57(5):1252–1275.

Rei W, Cordeau JF, Gendreau M, Soriano P (2009) Accelerating Benders decomposition by local branching. *INFORMS Journal on Computing* 21(2):333–345.

Reuther A, Kepner J, Byun C, Samsi S, Arcand W, Bestor D, Bergeron B, Gadepally V, Houle M, Hubbell M, Jones M, Klein A, Milechin L, Mullen J, Prout A, Rosa A, Yee C, Michaleas P (2018) Interactive supercomputing on 40,000 cores for machine learning and data analysis. *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 1–6 (IEEE).

Richardson R (1976) An optimization approach to routing aircraft. *Transportation Science* 10(1):52–71.

Rodríguez-Martín I, Salazar-González JJ (2010) A local branching heuristic for the capacitated fixed-charge network design problem. *Computers & Operations Research* 37(3):575–581.

Santoso T, Ahmed S, Goetschalckx M, Shapiro A (2005) A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research* 167(1):96–115.

Schmidt M, Le Roux N, Bach F (2017) Minimizing finite sums with the stochastic average gradient. *Mathematical Programming* 162(1):83–112.

Shapiro A, Dentcheva D, Ruszczynski A (2021) *Lectures on Stochastic Programming: Modeling and Theory* (SIAM).

Smith JE, Winkler RL (2006) The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science* 52(3):311–322.

Stubbs RA, Mehrotra S (1999) A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming* 86(3):515–532.

Trukhanov S, Ntaimo L, Schaefer A (2010) Adaptive multicut aggregation for two-stage stochastic linear programs with recourse. *European Journal of Operational Research* 206(2):395–406.

Van Roy TJ, Wolsey LA (1985) Valid inequalities and separation for uncapacitated fixed charge networks. *Operations Research Letters* 4(3):105–112.

Van Slyke RM, Wets R (1969) L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* 17(4):638–663.

Wei L, Gómez A, Küçükyavuz S (2022) Ideal formulations for constrained convex optimization problems with indicator variables. *Mathematical Programming* 192(1):57–88.

Wets RJB (1966) Programming under uncertainty: the equivalent convex program. *SIAM Journal on Applied Mathematics* 14(1):89–105.

Xie W, Deng X (2020) Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization* 30(4):3359–3386.

You F, Grossmann IE (2013) Multicut Benders decomposition algorithm for process supply chain planning under uncertainty. *Annals of Operations Research* 210(1):191–211.

Zakeri G, Philpott AB, Ryan DM (2000) Inexact cuts in Benders decomposition. *SIAM Journal on Optimization* 10(3):643–657.

Online Supplement: A Stochastic Benders Decomposition Scheme for Large-Scale Stochastic Network Design

Dimitris Bertsimas, Ryan Cory-Wright, Jean Pauphilet, Periklis Petridis

OS.1. Justification for the Strongly Quadratic Penalty Term in Problem (1)

In this section, we justify using the quadratic regularization term in Problem (1), from both a practical perspective and a theoretical one. We remark that the use of a regularization term in mixed-integer optimization is an increasingly popular modeling choice which has been discussed in detail in other works; we refer to Bertsimas et al. (2021), Bertsimas and Cory-Wright (2022) for a more detailed discussion of this matter.

From a practical perspective, a strongly quadratic term in the objective can model quadratic transportation costs or can be used to increase the robustness of the solution to parameter uncertainty. Indeed, as we show in Proposition OS.1.1, the second-stage cost in Problem (1) is equivalent to a worst-case cost with an ellipsoidal uncertainty set around the second-stage transportation costs $f^k_{i,j}$. Furthermore, as advocated in Bertsimas et al. (2021), artificially introducing a quadratic penalty with $\gamma > 0$ is an efficient smoothing technique for MIO problems with logical constraints, approximating the nominal objective function (to arbitrary precision by taking $\gamma \to +\infty$) while improving computational tractability. Formally, if $v(\gamma)$ denotes the optimal objective value of (1), one can easily show that

$$v(\gamma) - \frac{1}{2\gamma} \sum_{(i,j) \in \mathcal{E}} u_{i,j}^2 \leq v(\infty) \leq v(\gamma).$$

Further, we should emphasize that the method we develop in this paper does not require $1/\gamma > 0$ and applies in the case where there is no quadratic term in the objective as well ($\gamma \to \infty$ and $1/\gamma = 0$), as discussed in Bertsimas et al. (2021, Remark 2.6).

We now provide some theoretical evidence to justify the presence of a smooth strongly convex term in the objective of (1). In particular, we show that adding this term can be interpreted as equivalent to considering a robust version of the linear objective.

Proposition OS.1.1. *Fix $\gamma > 0$ and $\boldsymbol{z} \in \{0,1\}^{\mathcal{E}}$. Let us denote $\mathcal{X}$ the set of feasible second-stage flow transportation variables $\boldsymbol{x}$, i.e.,*

$$\mathcal{X} := \left\{ \boldsymbol{x} \geq \boldsymbol{0} \;\middle|\; \begin{array}{ll} \boldsymbol{A}\boldsymbol{x}^{k,r} = \boldsymbol{d}^{k,r}, & \forall k \in \mathcal{K}, r \in \mathcal{R} \\ \sum_{k \in \mathcal{K}} x_{i,j}^{k,r} \leq u_{i,j}, & \forall (i,j) \in \mathcal{E}, r \in \mathcal{R} \\ x_{i,j}^{k,r} = 0 \textit{ if } z_{i,j} = 0, & \forall (i,j) \in \mathcal{E} \end{array} \right\}.$$

There exists a parameter value $\lambda \geq 0$ such that

$$\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} f_{i,j}^{k} x_{i,j}^{k,r} + \frac{1}{2\gamma} \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \left(\sum_{k \in \mathcal{K}} x_{i,j}^{k,r} \right)^2 \tag{1}$$

achieves the same optimal solution as

$$\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \max \left\{ \sum_{k \in \mathcal{K}} \tilde{f}_{i,j}^{k} x_{i,j}^{k,r} \,:\, \tilde{f}_{i,j}^{k} = f_{i,j}^{k} + \zeta_{i,j}^{r}, \; \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} (\zeta_{i,j}^{r})^2 \leq \lambda^2 \right\}. \tag{2}$$

Proof of Proposition OS.1.1 Problems (1) and (2) have the same feasible set. Consider a feasible solution $\boldsymbol{x}$. We have

$$\begin{aligned} \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} \tilde{f}_{i,j}^{k} x_{i,j}^{k,r} &= \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} f_{i,j}^{k} x_{i,j}^{k,r} + \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} \tilde{\zeta}_{i,j}^{r} x_{i,j}^{k,r} \\ &= \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} f_{i,j}^{k} x_{i,j}^{k,r} + \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \tilde{\zeta}_{i,j}^{r} \left(\sum_{k \in \mathcal{K}} x_{i,j}^{k,r} \right). \end{aligned}$$

Hence, the worst-case value with respect to all vectors $\tilde{\boldsymbol{\zeta}}$ such that $\sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} (\tilde{\zeta}_{i,j}^{r})^2 \leq \lambda^2$ (i.e., the value of the inner maximization problem in (2)) is equal to

$$\sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} f_{i,j}^{k} x_{i,j}^{k,r} + \lambda \sqrt{ \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \left(\sum_{k \in \mathcal{K}} x_{i,j}^{k,r} \right)^2 } =: \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} f_{i,j}^{k} x_{i,j}^{k,r} + \lambda \sqrt{q(\boldsymbol{x})}.$$

Hence, minimizing the worst-case transportation cost in (2) is equivalent to minimizing the nominal cost, $\sum_r \sum_{(i,j)} \sum_k f_{i,j}^{k} x_{i,j}^{k,r}$ plus a penalty term, which is equal to the square root of the quadratic regularization term in (1), $q(\boldsymbol{x})$. To conclude the proof, we need to show that for any $\lambda > 0$ there exists a parameter value $\gamma > 0$ such that the penalties $\lambda\sqrt{q(\boldsymbol{x})}$ and $\frac{1}{2\gamma} q(\boldsymbol{x})$ lead to the same optimal solution.

For any $\lambda > 0$, by duality, there exists a constant q_0 such that Problem (2) is equivalent to

$$\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} f_{i,j}^{k} x_{i,j}^{k,r} \quad \text{s.t.} \quad \sqrt{q(\boldsymbol{x})} \leq \sqrt{q_0}. \tag{3}$$

Since $t \mapsto \sqrt{t}$ is increasing over $\mathbb{R}_+$, constraint (3) is equivalent to $q(\boldsymbol{x}) \leq q_0$ and the resulting problem is in turn equivalent to a problem of the same form as (1). □

Proposition OS.1.1 shows that the regularized objective in (1) is equivalent to a robust linear objective with ellipsoidal uncertainty set, *for each first-stage design decision* $\boldsymbol{z}$. Optimizing (over $\boldsymbol{z}$) for each objective can result in different solutions $(\boldsymbol{z}, \boldsymbol{x})$ though, because the first stage decision $\boldsymbol{z}$ is discrete so the value of λ that makes the penalized formulation equivalent to the constrained one is $\boldsymbol{z}$-specific (the equivalence requires a duality type of argument, which does not hold in general when jointly optimizing for $(\boldsymbol{z}, \boldsymbol{x})$ with $\boldsymbol{z}$ binary). However, after reformulating the logical constraints via algebraic linear/second-order cone constraints, we can show an equivalence results between the Boolean relaxations (proof omitted):

PROPOSITION OS.1.2. *Fix $\gamma > 0$. Let us denote $\mathcal{P}$ the set of feasible relaxed decision variables $(\boldsymbol{z}, \boldsymbol{x})$, i.e.,*

$$\mathcal{P} := \left\{ (\boldsymbol{z}, \boldsymbol{x}) \;\middle|\; \begin{array}{ll} \boldsymbol{z} \in [0,1]^{\mathcal{E}}, & \\ \sum_{(i,j)\in\mathcal{E}} z_{i,j} \leq c_0, & \\ \boldsymbol{A}\boldsymbol{x}^{k,r} = \boldsymbol{d}^{k,r}, & \forall k \in \mathcal{K}, r \in \mathcal{R} \\ \sum_{k\in\mathcal{K}} x_{i,j}^{k,r} \leq u_{i,j} z_{i,j}, & \forall (i,j) \in \mathcal{E}, r \in \mathcal{R} \\ \boldsymbol{x} \geq \boldsymbol{0} & \end{array} \right\}.$$

There exists a parameter value $\lambda \geq 0$ such that

$$\min_{(\boldsymbol{z},\boldsymbol{x})\in\mathcal{P}} \quad \sum_{(i,j)\in\mathcal{E}} c_{i,j} z_{i,j} + \frac{1}{|\mathcal{R}|} \sum_{r\in\mathcal{R}} \sum_{(i,j)\in\mathcal{E}} \left(\sum_{k\in\mathcal{K}} f_{i,j}^k x_{i,j}^{k,r} + \frac{1}{2\gamma} \left(\sum_{k\in\mathcal{K}} x_{i,j}^{k,r} \right)^2 \right)$$

achieves the same optimal solution as

$$\min_{(\boldsymbol{z},\boldsymbol{x})\in\mathcal{P}} \quad \sum_{(i,j)\in\mathcal{E}} c_{i,j} z_{i,j} + \frac{1}{|\mathcal{R}|} \sum_{r\in\mathcal{R}} \sum_{(i,j)\in\mathcal{E}} \max \left\{ \sum_{k\in\mathcal{K}} \tilde{f}_{i,j}^k x_{i,j}^{k,r} \; : \; \tilde{f}_{i,j}^k = f_{i,j}^k + \zeta_{i,j}^r, \; \sum_{r\in\mathcal{R}} \sum_{(i,j)\in\mathcal{E}} (\zeta_{i,j}^r)^2 \leq \lambda^2 \right\}.$$

OS.2. Omitted Proofs

OS.2.1. Proof of Proposition 1

Proof of Proposition 1 The minimization problem defining $f(\boldsymbol{z};\boldsymbol{d})$ can be seen as the sum of two minimization problems

$$\min_{\boldsymbol{x}^k \in \mathbb{R}_+^{\mathcal{E}}, k\in\mathcal{K}} \sum_{k\in\mathcal{K}} \langle \boldsymbol{f}^k, \boldsymbol{x}^k\rangle \text{ s.t. } \boldsymbol{A}\boldsymbol{x}^k = \boldsymbol{d}^k, \forall k \in \mathcal{K},$$

and

$$\begin{aligned}\min_{\boldsymbol{y}\in\mathbb{R}^{\mathcal{E}}} \frac{1}{2\gamma} \sum_{(i,j)\in\mathcal{E}} y_{i,j}^2 \text{ s.t. } & y_{i,j} \le u_{i,j}, \forall (i,j)\in\mathcal{E},\\ & y_{i,j} = 0 \textit{ if } z_{i,j} = 0, \forall (i,j)\in\mathcal{E},\end{aligned}$$

coupled via the constraints $\sum_{k\in\mathcal{K}} x_{i,j}^k = y_{i,j}, \forall (i,j)\in\mathcal{E}$. Therefore, by associating a dual variable $\alpha_{i,j}\in\mathbb{R}$ with each coupling constraint, we rewrite $f(\boldsymbol{z};\boldsymbol{d})$ as

$$\min_{\substack{\boldsymbol{x}^k\in\mathbb{R}_+^{\mathcal{E}}, k\in\mathcal{K}:\\ \boldsymbol{A}\boldsymbol{x}^k=\boldsymbol{d}^k, \forall k\in\mathcal{K}}} \quad \min_{\substack{\boldsymbol{y}\in\mathbb{R}^{\mathcal{E}}:\\ y_{i,j}\le u_{i,j}, \forall (i,j)\in\mathcal{E}\\ y_{i,j}=0 \textit{ if } z_{i,j}=0, \forall (i,j)\in\mathcal{E}}} \quad \max_{\boldsymbol{\alpha}\in\mathbb{R}_+^{\mathcal{E}}} \sum_{k\in\mathcal{K}} \langle \boldsymbol{f}^k - \boldsymbol{\alpha}, \boldsymbol{x}^k\rangle + \sum_{(i,j)\in\mathcal{E}} \left(\alpha_{i,j} y_{i,j} + \tfrac{1}{2\gamma} y_{i,j}^2\right).$$

By invoking standard results on saddle-point theorems (see, e.g., Bertsekas 1999), the order of the minimization and maximization operators on the function $f(\boldsymbol{z},\boldsymbol{d})$ can be exchanged[1] without altering the objective value. Moreover, after exchanging these operators, we can compute the dual of each minimization problem separately. Indeed,

$$\min_{\substack{\boldsymbol{x}^k\in\mathbb{R}_+^{\mathcal{E}}, k\in\mathcal{K}:\\ \boldsymbol{A}\boldsymbol{x}^k=\boldsymbol{d}^k, \forall k\in\mathcal{K}}} \sum_{k\in\mathcal{K}} \langle \boldsymbol{f}^k - \boldsymbol{\alpha}, \boldsymbol{x}^k\rangle \quad = \quad \max_{\substack{\boldsymbol{p}^k\in\mathbb{R}^{\mathcal{N}}:\\ \boldsymbol{A}^\top \boldsymbol{p}^k \le \boldsymbol{f}^k - \boldsymbol{\alpha}, \forall k\in\mathcal{K}}} \sum_{k\in\mathcal{K}} \langle \boldsymbol{p}^k, \boldsymbol{d}^k\rangle.$$

Second, to dualize

$$\begin{aligned}\min_{\boldsymbol{y}\in\mathbb{R}^{\mathcal{E}}} \sum_{(i,j)\in\mathcal{E}} \left(\alpha_{i,j} y_{i,j} + \tfrac{1}{2\gamma} y_{i,j}^2\right) \text{ s.t. } & y_{i,j} \le u_{i,j}, \forall (i,j)\in\mathcal{E},\\ & y_{i,j} = 0 \textit{ if } z_{i,j} = 0, \forall (i,j)\in\mathcal{E},\end{aligned}$$

let us first observe that we can omit the logical constraints by considering the change of variables $y_{i,j} = z_{i,j} w_{i,j}$ for $\boldsymbol{w}\in\mathbb{R}^{\mathcal{E}}$. Hence, we obtain

$$\begin{aligned}&\min_{\boldsymbol{w}\in\mathbb{R}^{\mathcal{E}}} \sum_{(i,j)\in\mathcal{E}} \left[z_{i,j}\alpha_{i,j} w_{i,j} + \tfrac{1}{2\gamma} z_{i,j} w_{i,j}^2\right] \text{ s.t. } z_{i,j} w_{i,j} \le z_{i,j} u_{i,j}, \forall (i,j)\in\mathcal{E}\\
&= \max_{\boldsymbol{\beta}\in\mathbb{R}_+^{\mathcal{E}}} \min_{\boldsymbol{y}\in\mathbb{R}^{\mathcal{E}}} -\sum_{(i,j)\in\mathcal{E}} z_{i,j}\beta_{i,j} u_{i,j} + \sum_{(i,j)\in\mathcal{E}} \left[z_{i,j}(\alpha_{i,j}+\beta_{i,j}) w_{i,j} + \tfrac{1}{2\gamma} z_{i,j} w_{i,j}^2\right]\\
&= \max_{\boldsymbol{\beta}\in\mathbb{R}_+^{\mathcal{E}}} -\sum_{(i,j)\in\mathcal{E}} z_{i,j}\beta_{i,j} u_{i,j} - \frac{\gamma}{2}\sum_{(i,j)\in\mathcal{E}} z_{i,j}(\alpha_{i,j}+\beta_{i,j})^2.\end{aligned}$$

All together, we obtain the desired reformulation. □

[1] In general, we require that a constraint qualification holds to be able to exchange the order of minimization and maximization operators (see, e.g., Bertsekas 1999). However, all constraints in Problem (2) are linear and it has a convex quadratic objective. Therefore, we can exchange the order of the operators in an assumption-free manner.

OS.2.2. Proof of Proposition 2

In this section, we provide a proof of Proposition 2. To clarify the presentation, we adopt a lighter set of notations:

Fix $\boldsymbol{z}$. For any $r \in \mathcal{R}$, we denote $\boldsymbol{\xi}^r := (\boldsymbol{\alpha}^r, \boldsymbol{\beta}^r, \boldsymbol{p}^r)$ the optimal dual solutions of (4) for $\boldsymbol{d} = \boldsymbol{d}^{\cdot,r}$. For any subset $\mathcal{S} \subseteq \mathcal{R}$, let us denote $\bar{\boldsymbol{\xi}}^{\mathcal{S}} := \frac{1}{|\mathcal{S}|} \sum_{r \in \mathcal{S}} \boldsymbol{\xi}^r$ the average of the optimal dual solutions $\boldsymbol{\xi}^r$ for $r \in \mathcal{S}$. For a random $\mathcal{S} \subseteq \mathcal{R}$ of fixed size $|\mathcal{S}|$, we will analyze the sub-optimality gap of $\bar{\boldsymbol{\xi}}^{\mathcal{S}}$, i.e., the quantity $q(\boldsymbol{z}, \boldsymbol{\xi}^r; \boldsymbol{d}^r) - q(\boldsymbol{z}, \bar{\boldsymbol{\xi}}^{\mathcal{S}}; \boldsymbol{d}^r) \, (\geq 0)$, for scenarios $r \in \mathcal{S}^c := \mathcal{R} \setminus \mathcal{S}$.

Proof of Proposition 2 Let us denote $M := \max_{r \in \mathcal{R}} \|\boldsymbol{\xi}^r\|_\infty$. Since $\|\boldsymbol{\xi}^r\|_\infty \leq M$, then $\left\|\bar{\boldsymbol{\xi}}^{\mathcal{S}}\right\|_\infty \leq M$ and there exists some constant $L > 0$ such that, for any $\mathcal{S}$ and any $r \notin \mathcal{S}$

$$\left|q(\boldsymbol{z}, \bar{\boldsymbol{\xi}}^{\mathcal{S}}, \boldsymbol{d}^r) - q(\boldsymbol{z}, \boldsymbol{\xi}^r, \boldsymbol{d}^r)\right| \leq L \left\|\bar{\boldsymbol{\xi}}^{\mathcal{S}} - \boldsymbol{\xi}^r\right\|.$$

We further decompose the right-hand side via a triangle inequality and sum the inequalities above across all $r \notin \mathcal{S}$ to obtain

$$\sum_{r \in \mathcal{S}^c} \left|q(\boldsymbol{z}, \bar{\boldsymbol{\xi}}^{\mathcal{S}}, \boldsymbol{d}^r) - q(\boldsymbol{z}, \boldsymbol{\xi}^r, \boldsymbol{d}^r)\right| \leq L|\mathcal{S}^c| \left\|\bar{\boldsymbol{\xi}}^{\mathcal{S}} - \bar{\boldsymbol{\xi}}^{\mathcal{R}}\right\| + L \sum_{r \in \mathcal{S}^c} \left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \boldsymbol{\xi}^r\right\|.$$

The first term corresponds to the difference between $\bar{\boldsymbol{\xi}}^{\mathcal{R}}$ and an unbiased estimate obtained via sampling without replacement. Denote d the dimension of $\boldsymbol{\xi}$. Hence, since the $\boldsymbol{\xi}^r$ are uniformly bounded, by Bardenet and Maillard (2015, corollary 2.5), there exists some universal constant M_1 such that for any $\delta > 0$, we have, with probability $1 - \delta$ on the subset $\mathcal{S}$ of fixed size $|\mathcal{S}|$,

$$\left\|\bar{\boldsymbol{\xi}}^{\mathcal{S}} - \bar{\boldsymbol{\xi}}^{\mathcal{R}}\right\| \leq M_1 \sqrt{d \left(\frac{1}{|\mathcal{S}|} - \frac{1}{|\mathcal{R}|}\right) \log(1/\delta)}, \tag{4}$$

For the second term, we simply use the bound

$$\sum_{r \in \mathcal{S}^c} \left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \boldsymbol{\xi}^r\right\| \leq \sqrt{|\mathcal{S}^c|} \sqrt{\frac{1}{|\mathcal{S}^c|} \sum_{r \in \mathcal{S}^c} \left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \boldsymbol{\xi}^r\right\|^2}.$$

For interpretation, we denote $\nu^2 := \dfrac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \boldsymbol{\xi}^r\right\|^2$, which can be interpreted as the variance in optimal dual variables of our problem. Then, the term on the right-hand side of the inequality above can be viewed as a bootstrap estimator of ν, which intuitively converges to ν as $\mathcal{S}^c \to \mathcal{R}$. To formalize this intuition, let us expand the squared norm term and apply the triangle inequality:

$$\begin{aligned}
\left|\frac{1}{|\mathcal{S}^c|} \sum_{r \in \mathcal{S}^c} \left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \boldsymbol{\xi}^r\right\|^2 - \nu^2\right| &= \left|\frac{1}{|\mathcal{S}^c|} \sum_{r \in \mathcal{S}^c} \|\boldsymbol{\xi}^r\|^2 - 2\langle \bar{\boldsymbol{\xi}}^{\mathcal{R}}, \bar{\boldsymbol{\xi}}^{\mathcal{S}^c} \rangle - \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \|\boldsymbol{\xi}^r\|^2 + 2\langle \bar{\boldsymbol{\xi}}^{\mathcal{R}}, \bar{\boldsymbol{\xi}}^{\mathcal{R}} \rangle\right| \\
&\leq \left|\frac{1}{|\mathcal{S}^c|} \sum_{r \in \mathcal{S}^c} \|\boldsymbol{\xi}^r\|^2 - \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \|\boldsymbol{\xi}^r\|^2\right| + 2\left|\langle \bar{\boldsymbol{\xi}}^{\mathcal{R}}, \bar{\boldsymbol{\xi}}^{\mathcal{R}} - \bar{\boldsymbol{\xi}}^{\mathcal{S}^c} \rangle\right| \\
&\leq \left|\frac{1}{|\mathcal{S}^c|} \sum_{r \in \mathcal{S}^c} \|\boldsymbol{\xi}^r\|^2 - \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \|\boldsymbol{\xi}^r\|^2\right| + 2M \left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \bar{\boldsymbol{\xi}}^{\mathcal{S}^c}\right\|
\end{aligned}$$

By Bardenet and Maillard (2015, corollary 2.5) again, there exists $M_2 > 0$ such that, with probability $1-\delta$,

$$\left|\frac{1}{|\mathcal{S}^c|}\sum_{r\in\mathcal{S}^c}\|\boldsymbol{\xi}^r\|^2 - \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\|\boldsymbol{\xi}^r\|^2\right| \leq M_2\sqrt{\left(\frac{1}{|\mathcal{S}^c|}-\frac{1}{|\mathcal{R}|}\right)\log(1/\delta)},$$

and $\left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \bar{\boldsymbol{\xi}}^{\mathcal{S}^c}\right\|$ satisfies a similar inequality as (4). All together, with probability $1-2\delta$,

$$\left|\frac{1}{|\mathcal{S}^c|}\sum_{r\in\mathcal{S}^c}\left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \boldsymbol{\xi}^r\right\|^2 - \nu^2\right| \leq M_2\sqrt{\left(\frac{1}{|\mathcal{S}^c|}-\frac{1}{|\mathcal{R}|}\right)\log(1/\delta)} + M_1\sqrt{d\left(\frac{1}{|\mathcal{S}^c|}-\frac{1}{|\mathcal{R}|}\right)\log(1/\delta)},$$

yielding

$$\sum_{r\in\mathcal{S}^c}\left\|\bar{\boldsymbol{\xi}}^{\mathcal{R}} - \boldsymbol{\xi}^r\right\| \leq \sqrt{|\mathcal{S}^c|}\nu + \sqrt{|\mathcal{S}^c|}M_3\left(\frac{1}{|\mathcal{S}^c|}-\frac{1}{|\mathcal{R}|}\right)^{1/4}(\log(1/\delta))^{1/4}, \tag{5}$$

with $M_3 := M_2 + M_1\sqrt{d}$.

Combining (4) and (5), we obtain that, with probability $1-3\delta$ over the sample $\mathcal{S}$,

$$\sum_{r\in\mathcal{S}^c}\left|q(\boldsymbol{z},\bar{\boldsymbol{\xi}}^{\mathcal{S}},\boldsymbol{d}^r) - q(\boldsymbol{z},\boldsymbol{\xi}^r,\boldsymbol{d}^r)\right| \leq L\sqrt{|\mathcal{S}^c|}\nu + E$$

where E is a bootstrap error term equal to

$$\begin{aligned}E &= L|\mathcal{S}^c|M_1\sqrt{d\left(\frac{1}{|\mathcal{S}|}-\frac{1}{|\mathcal{R}|}\right)\log(1/\delta)} + \sqrt{|\mathcal{S}^c|}M_3L\left(\frac{1}{|\mathcal{S}^c|}-\frac{1}{|\mathcal{R}|}\right)^{1/4}(\log(1/\delta))^{1/4}\\ &\leq \sqrt{|\mathcal{S}^c|}M_3L\left[\sqrt{|\mathcal{R}|}\left(\frac{1}{|\mathcal{S}|}-\frac{1}{|\mathcal{R}|}\right)^{1/2} + \left(\frac{1}{|\mathcal{S}^c|}-\frac{1}{|\mathcal{R}|}\right)^{1/4}\right]\sqrt{\log(1/\delta)},\end{aligned}$$

because $M_3 \geq M_1\sqrt{d}$ and for δ such that $\log(1/\delta) > 1$.

To conclude the proof, let us observe that $M_3 = M_2 + M_1\sqrt{d}$ and $d = 2|\mathcal{E}| + |\mathcal{N}| \times |\mathcal{K}|$. □

OS.3. A τ-cut Implementation of Benders Decomposition

To successfully combine the best aspects of single and multi-cut approaches, a τ-cut approach was proposed by Trukhanov et al. (2010), Contreras et al. (2011). They observed that scenarios can often be partitioned into subsets (or clusters) that are very similar to one another. Moreover, aggregating the cuts in each partition successfully compresses information about the second-stage cost surface and, on a per-iteration basis, is almost as fast as a single-cut approach. Accordingly, let $\cup_{c\in[k]}\mathcal{S}_c$ be a partition of $\mathcal{R}$. Then, at each iteration, the τ-cut approach solves the MIO:

$$\min_{\substack{\boldsymbol{z}\in\mathcal{Z}\\ \{\eta_c\in\mathbb{R}\}_{c\in[\tau]}}} \langle \boldsymbol{c},\boldsymbol{z}\rangle + \frac{1}{|\mathcal{R}|}\sum_{c\in[k]}\eta_c \text{ s.t. } \eta_c \geq \sum_{r\in\mathcal{S}_c} f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r}) + \langle \nabla f(\boldsymbol{z}^t;\boldsymbol{d}^{\cdot,r}), \boldsymbol{z}-\boldsymbol{z}^t\rangle, \forall t\in[T], \forall c\in[k], \quad (6)$$

and constructs each cut similarly to the single and multi-cut approaches. At each iteration, the τ-cut approach adds τ linear constraints (one per cluster $c\in[\tau]$). If $\tau=1$ (resp. $|\mathcal{R}|$), we recover the single-cut (resp. multi-cut) algorithm.

OS.3.1. Stochastic Variant of τ-cut Benders' Decomposition

A stochastic variant of the τ-cut approach can be developed analogously to the stochastic single-cut approach in Section 2.4, by applying our method for stochastic single-cut to each cluster $c\in[\tau]$.

Namely, we partition the set of scenarios $\mathcal{R}$ into τ sets $\mathcal{S}_c : c\in[\tau]$, and impose valid constraints of the form (9) to each epigraph variable η_c:

$$\eta_c \geq \sum_{r\in\mathcal{S}_c} q(\boldsymbol{z}^t,\boldsymbol{\alpha}^r,\boldsymbol{\beta}^r,\boldsymbol{p}^r;\boldsymbol{d}^{\cdot,r}) + \sum_{r\in\mathcal{S}_c}\langle \nabla_{\boldsymbol{z}} q(\boldsymbol{z}^t,\boldsymbol{\alpha}^r,\boldsymbol{\beta}^r,\boldsymbol{p}^r;\boldsymbol{d}^{\cdot,r}), \boldsymbol{z}-\boldsymbol{z}^t\rangle,\ \forall c\in[\tau]. \quad (7)$$

Then, at each iteration, we sample and solve (4) for a random subset $\mathcal{R}_{t,c}\subseteq\mathcal{S}_c$ of scenarios in each cluster and set $(\boldsymbol{\alpha}^r,\boldsymbol{\beta}^r,\boldsymbol{p}^r)=(\bar{\boldsymbol{\alpha}}^{\mathcal{R}_{r,c}},\bar{\boldsymbol{\beta}}^{\mathcal{R}_{r,c}},\bar{\boldsymbol{p}}^{\mathcal{R}_{r,c}})$ for $r\in\mathcal{S}_c\setminus\mathcal{R}_{t,c}$. From Proposition 2 applied to each cluster separately, we obtain that the approximation error for cluster c is bounded, with high probability, by a term that depends on the variance in dual optimal variables within cluster c, ν_c^2, plus a bootstrap estimation error term. Hence, if the clustering successfully reduces total weighted variance $\sum_{c\in[\tau]}\sqrt{|\mathcal{S}_c|}\nu_c$, a τ-cut approach could improve the lower bound obtained by single-cut, while using the same number of samples per iteration.

In practice (and in our implementation), it is not feasible to cluster the set of scenarios $\mathcal{R}$ based on their associated optimal dual solutions $\boldsymbol{\alpha}^r$ *at the incumbent solution*, because the clustering cannot change throughout the algorithm (it has to be independent from the incumbent). Intuitively, however, the optimization problem defining $\boldsymbol{\alpha}^r$, (4), is smooth and parametrized by $\boldsymbol{d}^{\cdot,r}$. From sensitivity analysis, the clusters obtained by applying the τ-means algorithm on the demand vectors $\boldsymbol{d}^{\cdot,r}$ or on the optimal dual variables $\boldsymbol{\alpha}^r$ at some initial vector $\boldsymbol{z}^0$ should lead to relatively homogeneous clusters in terms of optimal dual variables $\boldsymbol{\alpha}^r$ throughout the algorithm. In our implementation, we use the latter clustering (on the optimal dual variables $\boldsymbol{\alpha}^r$ computed at the root node for the initial solution $\boldsymbol{z}^0$).

Table OS.3.1 Runtime (in seconds) and final optimality gap (in %) for stochastic τ-cut approach, averaged over instances with the same number of nodes $|\mathcal{N}|$.

	Stochastic τ-cut	
$\lvert\mathcal{N}\rvert$	Runtime	Gap
10	86.31	0.39
30	4030.85	4.17
50	4462.14	2.58
70	4851.60	7.59
100	4946.72	6.58
150	5103.10	10.96
200	6028.52	10.54
300	4734.46	18.88
500	5456.29	25.83
700	5551.47	37.24

OS.3.2. Numerical Performance

In this section, we report (Table OS.3.1) the performance of the stochastic τ-cut approach on the same instances as the one used in Section 4.3 (see Table 4). As for the other cutting-plane methods, we warm-start the algorithm with cuts obtained from solving the perspective relaxation with a single-cut stochastic cutting-plane algorithm and applying these cuts at the root node. Overall, we observe that the τ-cut implementation improves upon the single-cut implementation in terms of computational time and optimality gap, although the benefit is less acute as the size of the instance grows. In terms of optimality, it achieves optimality gaps that are similar to those of the accelerated multi-cut stochastic cutting plane for the small instances but that deteriorate more as the number of nodes in the network increases.

OS.4. Computational Effect of Regularizer γ

The regularizing constant γ plays a crucial role in the performance of decomposition algorithms like Benders' decomposition; see, e.g., Bertsimas et al. (2021). An appropriate value of γ is essential for achieving optimal convergence and solution quality. When set too high, the regularizing term has minimal impact on the objective function, making the problem more challenging to solve. Conversely, when set too low, the regularizing term dominates the objective function, resulting in easier but less accurate solutions.

To illustrate this, Figure OS.4.1 presents the results of our experiments on **R** instances with 160 scenarios, where we vary the value of γ. It displays the average runtime and the objective value for each value of γ to provide insights on how the choice of γ affects the solution quality and computational performance of the algorithm. In the experimental results presented in Section 4, we selected an appropriate value for the regularizing constant γ in order to strike a balance between the two extremes of the spectrum, as depicted in Figure OS.4.1.

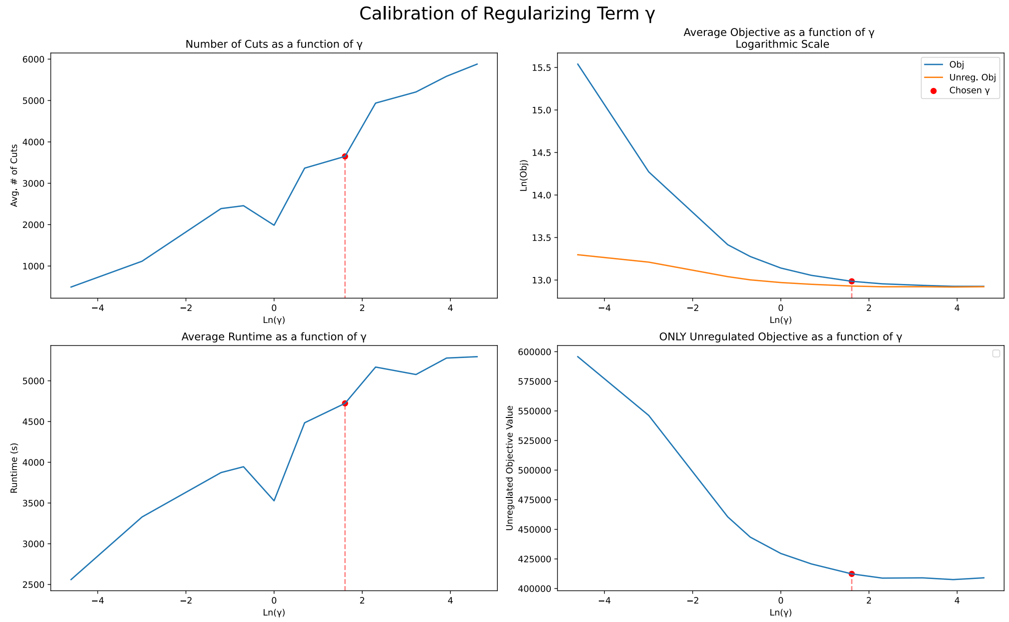


Figure OS.4.1 Effect of Regularizer on the Algorithm's Performance.

OS.5. Additional Numerical Results

In this section, we present additional numerical results that complement the results in Section 4.

OS.5.1. Instance Generation

OS.5.1.1. Synthetic instances We generate instances according to a methodology inspired by that of Günlük and Linderoth (2009). We construct a random graph by uniformly positioning $|\mathcal{N}|$ nodes over the unit square $[0,1]^2$ and randomly sampling edges to construct a set of feasible edges $\mathcal{E}_0$: We iteratively sample edges from the k-nearest neighbors graph (with $k = 6$) until we obtain a connected graph to ensure the feasibility of our instances. The construction cost for each edge, $c_{i,j}$ is drawn uniformly from $\mathcal{U}(1,4)$. Each commodity $k \in \mathcal{K}$ corresponds to an all-to-one shortest path problem with a single destination node $i_k \in \mathcal{N}$. For commodity k, we independently sample demands from all nodes $i' \in \mathcal{N}$, $d^k_{i'}$, uniformly between 5 to 20. We set $d^k_{i_k} := -\sum_{i' \neq i_k} d^k_{i'}$. We generate $\mathcal{R}$ demand scenarios for each commodity accordingly. This process is repeated for every scenario $r \in \mathcal{R}$. Flow circulation costs, $f^k_{i,j}$, are proportional to the edge length (by a factor 10). The capacity of each arc is scaled based on the maximum cumulative demand across all scenarios: $B_{i,j} := \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{E}} \max_{r \in \mathcal{R}} d^{k,r}_{ij}$. Formally, we sample the capacity for arc (i,j) according to $u_{ij} \sim \mathcal{U}(1,4) \cdot B_{i,j}/|\mathcal{E}_0|$. We fix the cardinality constraint to $c_0 = 2|\mathcal{E}_0|$.

OS.5.1.2. R instances As in Crainic et al. (2021, 2016), Boland et al. (2016), we use the **R** instances from classes 4 to 10. Each class corresponds to a particular network with its set of nodes, arcs, and commodities. Within each class, the library contains nine instances, associated with different set of arc capacities, edge construction costs, and flow transportation costs. Precisely, each instance within each class is associated with a 'class minor' ranging from 1 to 9 and corresponding to increasing ratios of fixed to variable cost and of total demand to total capacity (e.g., instance R4.1 or R10.9). Each class also contains one vector of nominal demands for each commodity (the same nomial for all instances within each class). However, we need samples of demand *scenarios*. Instead of implementing an ad-hoc sampling scheme, we use the same scenarios as those generated by (Rahmaniani et al. 2018) and available at `https://github.com/Ragheb2464/R-Instances`. For each class, these files contain 1,000 scenarios for different level of correlation between commodity demand.

Overall, each instance is characterized by a class number (e.g., R4), which determines the network, a class *minor* (e.g., R4.1), and a correlation between demands. In our experiments, we consider 7 classes, 3 class minors (1, 3, and 9), and 2 correlation values (0.0 and 0.8).

OS.5.2. Computing requirements for each experiment

This section provides a breakdown of the computational resources allocated for the experiments described in 4.1. Each CPU core of the MIT Supercloud Cluster corresponds to 4GB of allocated RAM.

OS.5.2.1. Synthetic instances For the experiments on synthetic instances, we request a number of CPU cores/memory that is increasing with the number of nodes in the network, $|\mathcal{N}|$, as described in Table OS.5.1.

Table OS.5.1 Number of CPU Cores and Memory (GB) allocated for the experiments on synthetic instances, as a function of the number of nodes in the network, $|\mathcal{N}|$.

$\vert\mathcal{N}\vert$	Memory (GB)	# Cores
10	4	1
30	8	2
50	16	4
70	20	5
100	28	7
150	40	10
200	52	13
300	76	19
500	128	32
700	176	44

OS.5.2.2. R instances For the experiments on the **R** instances, we request a number of CPU cores/memory that is increasing with the number of scenarios, $|\mathcal{R}|$, as summarized in Table OS.5.1.

Table OS.5.2 Number of CPU Cores and Memory (GB) allocated for the experiments on the R instances, as a function of the number of scenarios, $|\mathcal{R}|$.

$\vert\mathcal{R}\vert$	Memory (GB)	# Cores
10	20	5
30	20	5
50	20	5
100	20	5
200	28	7
500	64	16
1000	64	16

OS.5.3. Comparison of Different Stochastic Cutting-Plane Algorithms

In Section 4.2, we benchmark the performance of different variants of the stochastic cutting plane algorithm (namely the multi-, single-, and τ-cut algorithms) with different warm-starting strategies at the root node. Recall that we terminate our algorithm after 7,200 seconds or as soon as it achieves an optimality gap of with confidence level $\alpha = 0.90$.

Accordingly, the average computational time reported in Table 2 are capped at 7,200 seconds whenever the algorithm does not converge within this time limit. To appreciate this censoring issue, Table

OS.5.3 presents the fraction of instances solved to ϵ-optimality for each combination of algorithm and warm-start strategy.

Table OS.5.3 Percentage (in %) of instances for which the algorithm converged within the time limit (7,200 seconds), for the multi-, single-, and accelerated multi-cut stochastic cutting plane algorithms, with different warm-start strategies at the root node (none, single, or multi). Metrics are averaged across instances with the same number of nodes $|\mathcal{N}|$.

	Multi-Cut			Single-Cut			Accelerated Multi-Cut		
$\lvert\mathcal{N}\rvert$	None	Multi	Single	None	Multi	Single	None	Multi	Single
10	100	100.00	80.00	100.00	100.00	100.00	100.00	100.00	80.00
30	33.33	40.00	26.67	26.67	40.00	20.00	33.33	46.67	33.33
50	26.67	53.33	33.33	26.67	40.00	33.33	20.00	60.00	40.00
70	20.00	45.00	45.00	25.00	35.00	25.00	20.00	45.00	45.00
100	15.00	45.00	50.00	25.00	30.00	40.00	20.00	50.00	45.00
150	10.00	45.00	45.00	25.00	25.00	35.00	15.00	40.00	50.00
200	15.00	30.00	50.00	20.00	35.00	45.00	10.00	50.00	50.00

For the deterministic cutting-plane method (Benders' decomposition), numerous acceleration strategies have been proposed in the literature, e.g., based on valid inequalities (VI) or Magnanti-Wong Pareto-optimal cuts (MW; Magnanti and Wong 1984). We implemented different variants of the scheme based on this strategy and report their performance in Table OS.5.4. Consistent with Papadakos (2008), we find that Magnanti-Wong cuts often do more harm than good, and accordingly we do not include them in our implementation in the main text.

Table OS.5.4 Runtime (in seconds) and final optimality gap (in %) for different variants of the deterministic cutting-plane algorithm, averaged over instances with the same number of nodes $|\mathcal{N}|$. In addition to the naive implementation of the algorithm (with lazy callbacks), we consider adding valid inequalities (VI), Magnanti-Wong Pareto dominating cuts (MW), or both (VI+MW).

Speed-up	None		VI		MW		MW + VI	
$\|\mathcal{N}\|$	Runtime	Gap	Runtime	Gap	Runtime	Gap	Runtime	Gap
10	247.79	0.02	326.13	0.02	454.61	0.02	510.20	0.02
30	7163.94	6.22	7181.16	14.86	7200.00	7.72	7178.66	21.95
50	7200.00	4.87	7200.00	18.47	7200.00	6.29	7200.00	21.53
70	7200.00	11.85	7186.38	41.29	7200.00	12.93	7200.00	44.50
100	7165.78	16.37	7200.00	49.15	7200.00	16.47	7200.00	51.00
150	7186.61	23.49	7196.71	59.96	7200.00	26.59	7200.00	60.97
200	6853.71	26.68	7138.72	60.39	7039.69	28.77	7188.38	66.20
300	6237.87	23.11	6832.63	63.33	6818.90	24.39	7194.16	71.60
500	6441.49	49.09	6510.74	81.27	6874.99	51.11	7192.51	75.37
700	6499.08	53.39	6947.16	87.48	7012.66	69.53	6918.72	87.49

OS.5.4. Benchmarking Scalability on Synthetic Instances

To verify the correctness of our implementation, we use the smallest instances to verify that all methods terminate with the same optimal solution. To this end, Table OS.5.5 reports the optimality gap (in %) and final objective value for each algorithm, averaged over instances with the same number of nodes $|\mathcal{N}|$ and for which Gurobi converged to within 5% of optimality.

Table OS.5.5 Optimality gap (in %) and final objective value for each algorithm, averaged over synthetic instances with the same number of nodes $|\mathcal{N}|$, where Gurobi converged to within 5% of optimality.

	`Gurobi` with (1)		Deterministic		Stochastic	
$\|\mathcal{N}\|$	Gap	Objective	Gap	Objective	Gap	Objective
10	0.00	10,502.04	0.02	10,502.04	0.23	10,509.23
30	42.68	1,206,452.87	6.22	441,821.98	4.30	404,461.28
50	67.71	3,994,890.55	4.87	1,109,686.50	3.72	1,025,044.43
70	77.56	19,248,454.24	11.85	5,085,529.77	12.12	5,055,830.36

Figure OS.5.1 (resp. OS.5.2) illustrates the scalability of our single-cut method (resp. accelerated multi-cut) with respect to the number of scenarios and commodities and depicts the optimality gap achieved depending on the total number of nodes $|\mathcal{N}|$ (horizontal axis), the number of scenarios $|\mathcal{R}|$, and the number of commodities $|\mathcal{K}|$. We observe that the complexity of the problem (measured in

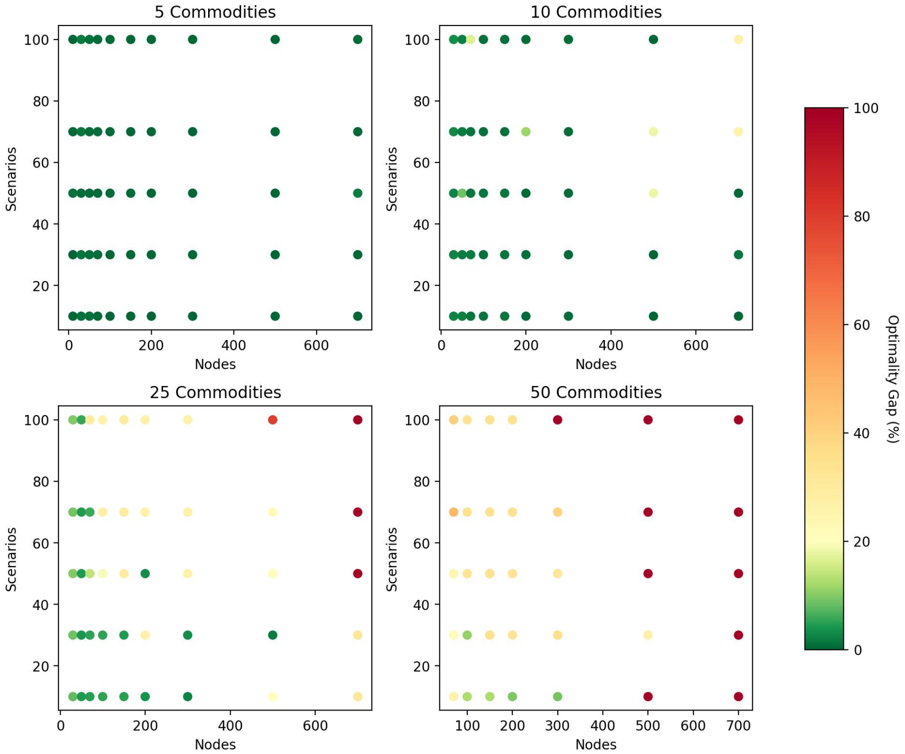


Figure OS.5.1 Optimality gaps achieved by the single-cut stochastic cutting plane algorithm on all synthetic instances. For each combination of number of nodes $|\mathcal{N}|$, number of commodities $|\mathcal{K}|$, and number of scenarios $|\mathcal{R}|$, results are averaged across 3 random instances.

terms of the final optimality gap) increases with all three problem dimensions, with the number of commodities and nodes having the most noticeable impact.

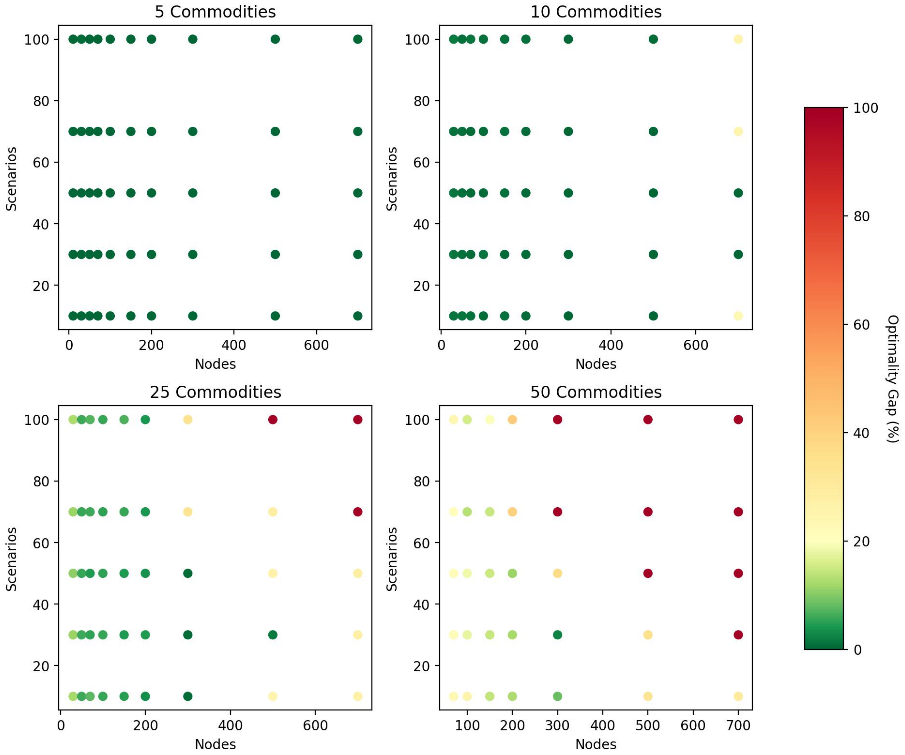


Figure OS.5.2 **Optimality gaps achieved by the accelerated multi-cut stochastic cutting plane algorithm on all synthetic instances. For each combination of number of nodes $|\mathcal{N}|$, number of commodities $|\mathcal{K}|$, and number of scenarios $|\mathcal{R}|$, results are averaged across 3 random instances.**

OS.5.5. Benchmarking on the Instances from Crainic et al. (2000)

In Section 4.4, we compare the performance of different cutting-plane algorithms for solving stochastic network design instances from the **R** instances. One alternative is to solve the perspective reformulation (1) with a mixed-integer second-order cone solver like Gurobi. As displayed in Figure OS.5.3, however, it performs worse than, e.g., a deterministic Benders decomposition scheme.

Figure OS.5.4 reports the distribution of the optimality gaps achieved by each method, over all **R** instances. We find that the ratio of fixed to variable cost and of total demand to total capacity, as controlled by the ‘class minor’ of each instance, is the main driver of the instance complexity, with a higher minor (i.e., higher ratios) resulting in larger optimality gaps (i.e., harder instances).

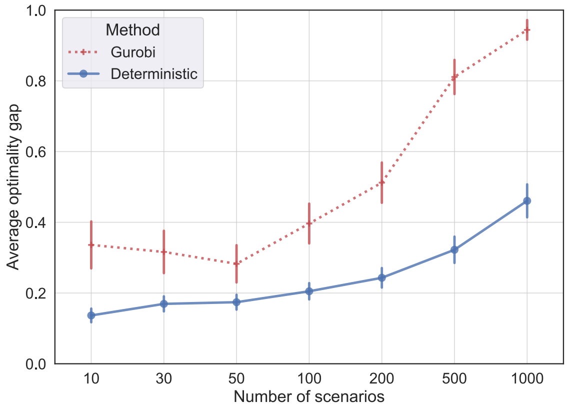


Figure OS.5.3 **Average optimality gap on the R instances, achieved by the Gurobi on the formulation (1) compared with the deterministic cutting-plane algorithms, for different number of scenarios. Bars represent standard errors.**

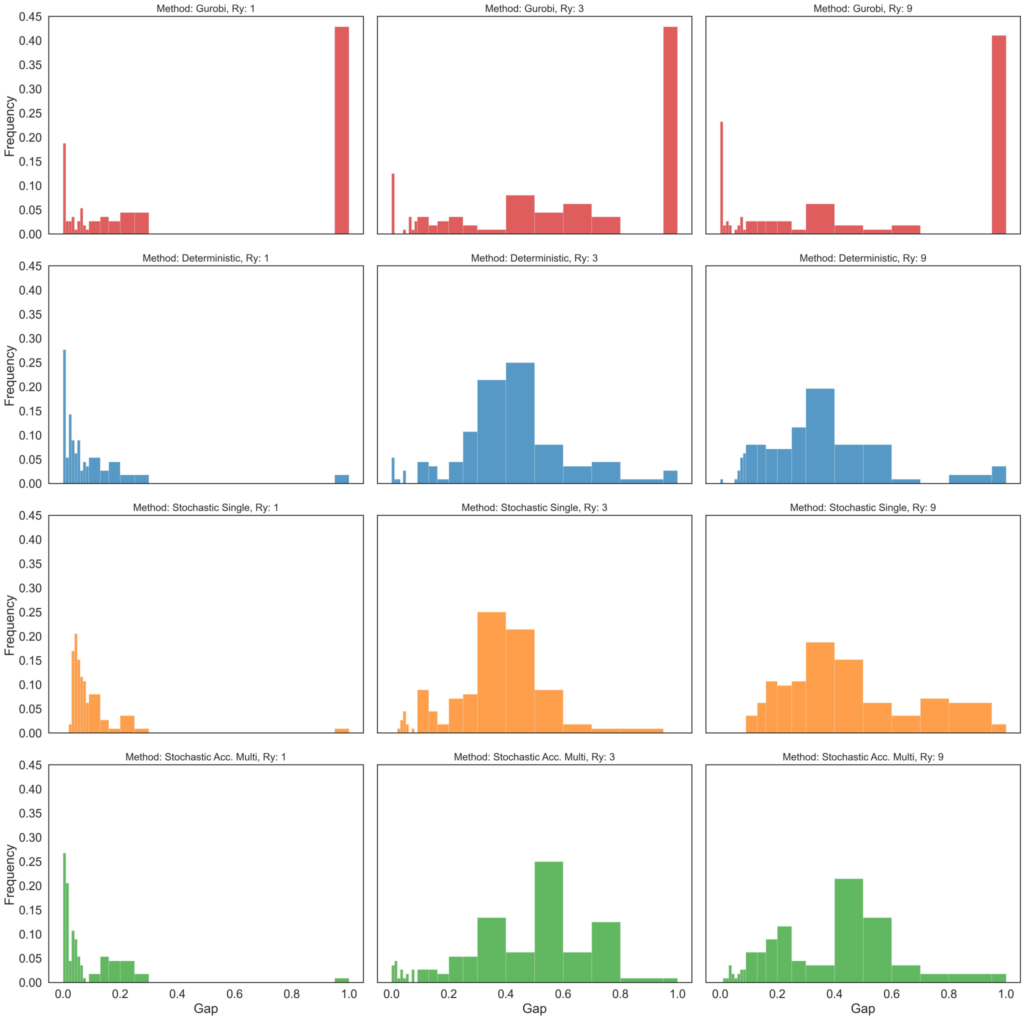


Figure OS.5.4 **Distribution of the optimality gap achieved on the R instances by the perspective reformulation, the deterministic Benders decomposition, and our stochastic (single-cut and accelerated multi-cut) cutting-plane algorithms. Results are grouped according to the instance class minor (see definition in Section OS.5.1).**

References

Bardenet R, Maillard OA (2015) Concentration inequalities for sampling without replacement. *Bernoulli* 21(3):1361–1385.

Bertsekas DP (1999) *Nonlinear Optimization* (Athena Scientific, Belmont).

Bertsimas D, Cory-Wright R (2022) A scalable algorithm for sparse portfolio selection. *INFORMS Journal on Computing* 34:1489–1511.

Bertsimas D, Cory-Wright R, Pauphilet J (2021) A unified approach to mixed-integer optimization problems with logical constraints. *SIAM Journal on Optimization* 31(3):2340–2367.

Boland N, Fischetti M, Monaci M, Savelsbergh M (2016) Proximity Benders: a decomposition heuristic for stochastic programs. *Journal of Heuristics* 22(2):181–198.

Contreras I, Cordeau JF, Laporte G (2011) Benders decomposition for large-scale uncapacitated hub location. *Operations Research* 59(6):1477–1490.

Crainic TG, Gendreau M, Farvolden JM (2000) A simplex-based Tabu search method for capacitated network design. *INFORMS Journal on Computing* 12(3):223–236.

Crainic TG, Hewitt M, Maggioni F, Rei W (2021) Partial Benders decomposition: general methodology and application to stochastic network design. *Transportation Science* 55(2):414–435.

Crainic TG, Rei W, Hewitt M, Maggioni F (2016) *Partial Benders Decomposition Strategies for Two-Stage Stochastic Integer Programs*, volume 37 (CIRRELT).

Günlük O, Linderoth J (2009) Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical Programming* 183–205.

Magnanti TL, Wong RT (1984) Network design and transportation planning: models and algorithms. *Transportation Science* 18(1):1–55.

Papadakos N (2008) Practical enhancements to the Magnanti–Wong method. *Operations Research Letters* 36(4):444–449.

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2018) Accelerating the Benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization* 28(1):875–903.

Trukhanov S, Ntaimo L, Schaefer A (2010) Adaptive multicut aggregation for two-stage stochastic linear programs with recourse. *European Journal of Operational Research* 206(2):395–406.