Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Partial Benders Decomposition: General Methodology and Application to Stochastic Network Design

Teodor Gabriel Crainic, Mike Hewitt, Francesca Maggioni, Walter Rei

TRANSPORTATION SCIENCE
Vol. 55, No. 2, March–April 2021, pp. 414–435
ISSN 0041-1655 (print), ISSN 1526-5447 (online)
http://pubsonline.informs.org/journal/trsc

Partial Benders Decomposition: General Methodology and Application to Stochastic Network Design

Teodor Gabriel Crainic,[a,b] Mike Hewitt,[c] Francesca Maggioni,[d] Walter Rei[a,b]

[a] Département d'analytique, opérations et technologies de l'information, École des Sciences de la Gestion, Université du Québec à Montréal, Montréal, Québec H2X 3X2, Canada; [b] Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Québec H3T 1J4, Canada; [c] Department of Information Systems and Supply Chain Management, Quinlan School of Business, Loyola University Chicago, Chicago, Illinois 60611; [d] Department of Economics, Università degli studi di Bergamo, 24127 Bergamo, Italy

Contact: crainic.teodor@uqam.ca, https://orcid.org/0000-0002-4424-0984 (TGC); mhewitt3@luc.edu, https://orcid.org/0000-0002-9786-677X (MH); francesca.maggioni@unibg.it (FM); rei.walter@uqam.ca, https://orcid.org/0000-0001-6626-8251 (WR)

Received: November 26, 2018
Revised: April 29, 2020
Accepted: August 19, 2020
Published Online in Articles in Advance: December 24, 2020

https://doi.org/10.1287/trsc.2020.1022

Abstract. Benders decomposition is a broadly used exact solution method for stochastic programs, which has been increasingly applied to solve transportation and logistics planning problems under uncertainty. However, this strategy comes with important drawbacks, such as a weak master problem following the relaxation step that confines the dual cuts to the scenario subproblems. In this paper, we propose a *partial Benders decomposition* methodology, based on the idea of including explicit information from the scenario subproblems in the master. To investigate the benefits of this methodology, we apply it to solve a general class of two-stage stochastic multicommodity network design models. Specifically, we solve the challenging variant of the model where both the demands and the arc capacities are stochastic. Through an extensive experimental campaign, we clearly show that the proposed methodology yields significant benefits in computational efficiency, solution quality, and stability of the solution process.

Funding: Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC) through its Discovery Grant program. The authors also gratefully acknowledge the support of Fonds de recherche du Québec through their infrastructure grants and Calcul Québec and Compute Canada for access to their high-performance computing infrastructure. F. Maggioni acknowledges the Research Council of the University of Bergamo for the 2015 vsiting professor research grant support .

Keywords: partial Benders decomposition • L-shaped algorithm • stochastic network design models

1. Introduction

Since its introduction in 1962, Benders decomposition (Benders 1962) has become one of the most used exact solution approaches for large-scale optimization, mixed integer programming (MIP) in particular (Costa 2005, Rahmaniani et al. 2017), while also being successfully adapted for, for example, nonlinear optimization (Geoffrion 1972) and stochastic programming (Van Slyke and Wets 1969). Benders decomposition is actually an essential methodology when addressing stochastic programming formulations (Birge and Louveaux 2011), the range of application areas being extremely broad and continuously growing: hub location (de Camargo, de Miranda, and Luna 2009; Contreras, Cordeau, and Laporte 2011), production routing (Adulyasak, Cordeau, and Jans 2015), inventory routing (Wheatley, Gzara, and Jewkes 2015), supply chain design in the health sector (Pishvaee, Razmi, and Torabi 2014), and capacity planning (MirHassani et al. 2000), to name a few examples. As clearly illustrated in a recent literature review (Rahmaniani et al. 2017), two interrelated trends have been observed in the last two decades. On the one hand, there has been an increased focus to address stochastic models for transportation and logistic applications. On the other hand, there has also been a growing interest from the operations research community in Benders decomposition, which has been successfully used to solve such models.

In a stochastic program, decisions are defined in stages according to when the values of the *stochastic parameters* (i.e., the subset of parameters involving uncertainty) become known. One thus distinguishes the so-called *first stage* or a priori decisions, to be made before any information is known, from those to be taken once the informational flow begins, that is, the *recourse* decisions proper to the second stage and onward. The objective function can then be defined as finding an a priori solution minimizing its associated cost plus a probabilistic measure of the recourse cost it entails, for example, the expected cost, a value at risk, or an expected shortfall cost (Rockafellar and Uryasev 2013).

Benders decomposition applied to stochastic programs, also called the *L-shaped algorithm* (Van Slyke and Wets 1969), enables such programs to be decomposed according to the realizations of the random events that set the values of the associated stochastic parameters. A finite set of representative scenarios is generally used to approximate the possible outcomes for the values of the stochastic parameters, the stochastic model being then formulated in an extensive form by duplicating the second stage decisions for each scenario (Birge and Louveaux 2011). Given that the large-scale nature of such models is due, in large part, to the number of scenarios used to represent uncertainty, Benders decomposition promises to simplify the resolution.

However, this strategy also comes with important drawbacks that need to be addressed to produce an overall efficient solution procedure. Recall that Benders decomposition for stochastic programs relies on the application of three main steps (Geoffrion 1970a, b): *projection* of the model onto the subspace defined by the first stage decision variables; *dualization* of the projected term, producing an equivalent model expressed as a set of valid inequalities (cuts) that define the feasibility requirements (*feasibility cuts*) and projected costs (*optimality cuts*) for the first stage decision variables; *relaxation*, where a *master* problem and the *scenario subproblems* are iteratively solved to guide the search process and generate violated cuts, respectively. The main drawback when applying this L-shaped algorithm is that the initial relaxation step produces a weak master-problem formulation. Indeed, the cuts included in the equivalent model reflect the second stage of the stochastic model and, once relaxed, make the master problem lose all relevant information concerning the recourse decisions, in terms of both the projected costs and the feasibility of the scenario subproblems. These cuts are reintroduced progressively (iteratively) during the run of the L-shaped algorithm by solving each time the relaxed master problem. This leads to various computational problems, such as instability with respect to the cuts that are generated (especially at the beginning of the solution process), erratic progression of the bounds generated by the algorithm, and an overall slow convergence of the procedure. (The relaxed master problem is solved each time a cut is generated.)

Our goal is to introduce mechanisms addressing these challenges, while focusing on solving stochastic network design models. Stochastic network design models represent an important and general combinatorial optimization problem class known for its inherent complexity (the deterministic version is already NP-hard and computationally challenging) and wide applicability, in particular, in the context of planning transportation and logistics operations. We will show how the proposed mechanisms can greatly enhance the L-shaped algorithm when applied to solve such models. Our goal is achieved via the four main contributions that are made in the present paper.

The first main contribution is to propose a comprehensive *partial Benders decomposition* (*PBD*) methodology, based on the idea of including explicit information from the scenario subproblems in the master. It should be noted that the developed methodology is presented using the general formulation for two-stage integer stochastic programs with continuous recourse to emphasize its general applicability. The proposed PBD thus strengthens the master formulation which, in turn, helps overcome the challenges evoked above. While previous studies have applied similar ideas to improve the performance of the L-shaped algorithm for particular stochastic optimization problems (see MirHassani et al. 2000; Bihlmaier, Koberstein, and Obst 2009; Batun et al. 2011; Liu, Ferris, and Zhao 2014), a comprehensive methodology for general settings does not exist.

The methodology we propose is built on two types of PBD strategies, based on scenario retention and creation, respectively. The second contribution of this paper is thus to present theoretical results supporting the principles on which these strategies are based. *Scenario-retention* strategies follow on the idea of projecting the original two-stage stochastic integer program onto a larger subspace, one that includes a subset of the second stage scenario variables. By doing so, a smaller part of the original problem is relaxed at the beginning of the solution process. Previous studies (Birge 1982; Sandikçi, Kong, and Schaefer 2013) have shown that high-quality lower bounds can be derived for stochastic MIPs by formulating alternative problems using subsets of scenarios chosen from the representative set. When applied in the context of the L-shaped algorithm, our proposed strategy also directly improves the lower bound generated by the algorithm at each iteration of the search process. In addition, by retaining more of the original problem structure in the master, the algorithm will benefit from integer programming solution techniques, such as preprocessing and cut generation, that are available in state-of-the-art MIP solvers. As for how the selection of the subproblems to keep in the master is performed, we propose two strategies that not only eliminate the feasibility cuts for the retained scenarios, but also significantly reduce the number of feasibility cuts for the nonretained ones.

Scenario-creation strategies, on the other hand, are designed to specifically improve the lower bound provided by the optimality cuts that are included in the master to approximate the projected costs. Using an aggregation approach (Litvinchev and Tsurkov 2003), we show how valid artificial scenario subproblems can be created and included in the master to directly impact the values of the projection variables that approximate the recourse cost. The use of aggregation strategies has shown great promise in the context of solving stochastic programs, for example, scenario constraint aggregation has been applied to derive, and then tighten, both lower and upper bounds for multistage stochastic programs (Rosa and Takriti 1999). Aggregation has also been applied over the scenario subproblems (i.e., variables and constraints) to obtain efficient bounds for stochastic models (Birge 1985). This general strategy was further developed to design solution methods for stochastic programs (under value at risk and expected cost objective functions, respectively) that apply scenario subproblem aggregation using partitions of the scenario set (Espinoza and Moreno 2014, Song and Luedtke 2015). Such partitions were then iteratively refined to improve the obtained bounds. In the present paper, we show that scenario subproblem aggregation can also be used to derive a set of valid inequalities that can be directly included in the master problem to strengthen the relaxation throughout the solution process.

The methodology is completed by two *hybrid* strategies created by combining the two pure scenario-retention and the pure scenario-creation strategies. These hybrid strategies, along with optimization models for implementing them, constitute the third contribution of this paper.

The fourth contribution of the present paper is to apply the PBD strategy to specifically solve two-stage stochastic multicommodity network design problems. Recent studies have shown that Benders decomposition can be efficiently applied to solve stochastic network design problems. Shen, You, and Ma (2017) proposed an efficient cutting-plane method to solve the single-commodity variant under both stochastic demands and arc capacities, while also imposing robust constraints that limit the worst-case cost for the networks obtained. In Rahmaniani et al. (2018), a Benders decomposition method, which includes a series of enhancements (among which is an application of a preliminary version of our scenario-retention strategy (Crainic et al. 2016)), was proposed for the multicommodity variant under stochastic demands. We apply our methodology to solve the more general and challenging variant of the problem, where multicommodities are considered and both stochastic demands and stochastic arc capacities are assumed.

An extensive experimental campaign is performed and the conducted analysis clearly shows the significant benefit in computational efficiency and solution quality of the proposed methodology. Furthermore, our computational experiments qualify the relative merits of our approach and provide insights in the behavior of the proposed PBD. In particular, when compared with either CPLEX and CPLEX's Benders implementation, the PBD hybrid strategies significantly reduce the optimality gap and the number of cuts generated and accelerate the converge, while increasing the stability of the solution process and identifying high-quality feasible solutions faster when solving the considered stochastic network design model.

The rest of the paper is structured as follows. Section 2 briefly recalls Benders decomposition when applied to stochastic models, while Section 3 reviews the different strategies proposed in the literature to improve the algorithm, which enables us to state the novelty of the contributions made in the present paper. Section 4 is dedicated to the description of the proposed PBD methodological framework. Section 5 describes the experimental design, which includes the specific problem class on which PBD is applied, the solution procedures implemented, and the different instances used in the experiments, while Section 6 presents the analysis of the computational results. We conclude in Section 7.

2. Benders Decomposition for Two-Stage Stochastic Programs

Let $\Re^n$ stand for the space of all n-dimensional real vectors, and let $\Re^{m\times n}$ stand for the space of real $m \times n$ matrices. Let $y \in \mathcal{Y} \subseteq \Re_+^{n_1}$ define a set of first stage decision variables that must take on integer values and satisfy the constraint set $Ay = b$, where $A \in \Re^{m_1\times n_1}$ is a known matrix and $b \in \Re^{m_1}$ is a known vector. Let S be the set of possible scenarios, with cardinality $|S|$; each scenario $s \in S$ has a probability p_s of occurring. We associate with each scenario $s \in S$ a set of second stage decision variables $x^s \in \Re_+^{n_2}$ that, together with the first stage decisions y, must satisfy the constraints $B^s y + Dx^s = d^s$. It should be noted that we consider a fixed recourse matrix $D \in \Re^{m_2\times n_2}$, with both $B^s \in \Re^{m_2\times n_1}$ and $d^s \in \Re^{m_2}$ defining the stochastic parameters. With objective coefficients $f \in \Re^{n_1}$ and $c \in \Re^{n_2}$ associated with the y and x^s variables, respectively, for each scenario s, we have the optimization problem P, where $v^\star$ defines the associated optimal value.

$$(P) \quad v^\star = \min f^\top y + \sum_{s\in S} p_s c^\top x^s \tag{1}$$

subject to

$$Ay = b, \tag{2}$$
$$B^s y + Dx^s = d^s, \quad \forall s \in S, \tag{3}$$
$$y \in \mathcal{Y},\ x^s \geq 0, \quad \forall s \in S. \tag{4}$$

We note that we assume bounded second stage problems, that is, that $c^\top x^s \geq 0$ for all feasible x^s. While we focus on the case of continuous recourse decisions in this paper, we discuss the generalization of the method to the case where integrality requirements are present in both of the decision stages in the conclusion.

The optimal values of the variables x^s given fixed values for the y variables are obtained by solving the subproblem $SP(y)_s$ for each scenario s:

$$\left(SP(y)_s\right) \quad z^s(y) = \min c^\top x^s \text{ s.t. } Dx^s = d^s - B^s y, x^s \geq 0\,,$$

and we may reformulate P (projection step) as

$$\min f^\top y + \sum_{s \in S} p_s z^s(y) \text{ s.t. } Ay = b,\ y \in \mathcal{Y}.$$

Taking the dual of $SP(y)_s$, with dual variables $g \in \Re^{m_2}$, yields problem $DSP(y)_s$ for each $s \in S$:

$$\left(DSP(y)_s\right) \quad \max g^\top \left(d^s - B^s y\right) \text{ s.t. } g^\top D \leq c.$$

Because the cost vector c does not vary by scenario, $Q = \{g : g^\top D \leq c\}$, the feasible region of $DSP(y)_s$, is the same for all scenarios. We assume that Q is not empty and has extreme points q^i, $i \in \mathcal{I}$, with cardinality $|\mathcal{I}|$, and extreme rays w^j, $j \in \mathcal{J}$, with cardinality $|\mathcal{J}|$. A valid reformulation of P (dualization step), which we refer to as BP, the *master problem* (or *Benders problem*), then is

$$\min f^\top y + \sum_{s \in S} p_s z^s \tag{5}$$

subject to

$$Ay = b, \tag{6}$$
$$q^{i\top}\left(d^s - B^s y\right) \leq z^s, \quad \forall i \in \mathcal{I}, s \in S\,, \tag{7}$$
$$w^{j\top}\left(d^s - B^s y\right) \leq 0, \quad \forall j \in \mathcal{J}, s \in S\,, \tag{8}$$
$$y \in \mathcal{Y}, z^s \geq 0, \quad s \in S, \tag{9}$$

where Constraints (7) and (8) represent the optimality and feasibility cuts, respectively.

Solving the Problem (5)–(9) (relaxation step) directly requires enumerating all the extreme points and rays of Q. Consequently, Benders decomposition rather solves repeatedly a relaxation, wherein only a subset of Constraints (7) and (8) are considered. Let $I \subseteq \mathcal{I}$ and $J \subseteq \mathcal{J}$ be such subsets of extreme points q^i, $i \in I$, and rays w^j, $j \in J$, respectively, and let $BP(I, J)$ be the relaxed master problem with the corresponding optimality and feasibility cuts. Solving the relaxation produces a vector $\overline{y}$. Dual subproblems $DSP(\overline{y})_s$ are then formed and solved to determine whether any optimality or feasibility cuts are violated. If so, they are added to the relaxation and the process is repeated. Otherwise, problem P has been solved to optimality. This solution process sums up what is often referred to as the *multicut* version of the L-shaped method (Birge and Louveaux 1988; see Van Slyke and Wets 1969 for the *single-cut* version wherein the scenario-based cuts generated at each iteration are aggregated into a single cut).

In all cases, while Benders decomposition converges to the optimal solution, the problem structure associated with the linking Constraints (3) is lost. As a result, many of the valid inequalities that have been developed for the deterministic (single-scenario) version are inapplicable. Furthermore, the relaxation of Constraints (7)–(8) eliminates from BP all guiding information for y with respect to the second stage of the problem. Therefore, when the solution process begins, the first stage solutions obtained may be arbitrarily poor with respect to their recourse cost. They may also be far from feasible in the second stage. Given that violated cuts are only introduced after the current relaxed $BP(I, J)$ is solved, the overall solution process can be excessively slow. In the next section, we present the different strategies that have been proposed to improve the Benders decomposition algorithm.

3. Accelerating Benders Decomposition—A Brief Literature Review

We briefly review the different strategies proposed in the literature to improve and accelerate Benders decomposition in both the general MIP context and those specifically designed to enhance the L-shaped algorithm. We refer the reader to Rahmaniani et al. (2017) for a detailed review. These strategies can be classified in four categories.

The first category groups methods that are based on the idea of generating cuts without the numerical burden of optimally solving the master problem each time. As an example, Geoffrion and Graves (1974) proposed to solve the relaxed master problem to obtain a feasible solution that is within an optimality gap of ϵ. Similar ideas can be applied to the scenario subproblems, by using suboptimal extreme points of the dual region of the scenario subproblems to generate valid cuts (Zakeri, Philpott, and Ryan 2000), as well as to the generation of valid feasibility and optimality cuts solving the linear relaxation of the original problem (McDaniel and Devine 1977). Following a different idea, it was shown in Côté and Laughton (1984) that Lagrangian relaxation can be applied to obtain optimality and feasibility cuts whenever the remaining constraint set in the master problem presents a special structure that is amenable to specialized algorithms. Similarly, in Rei et al. (2009) and Poojari and Beasley (2009), this idea was followed and Benders algorithms generating multiple cuts at each iteration by solving the master problem using local branching and a genetic procedure, respectively, were proposed with significant numerical improvements for deterministic (Poojari and Beasley 2009, Rei et al. 2009) and stochastic (Rei et al. 2009) integer programs.

The second category includes strategies that define alternate formulations for the master problem. Cross decomposition (Roy 1983) belongs to this category and is based on combining primal (Benders) and dual (Lagrangian) decompositions to solve MIPs. The author showed that a sequence of solutions to the Benders master problem can be obtained by alternately solving the Benders and Lagrangian subproblems. Then, solving the Benders master problem, on occasion only, maintains convergence to optimality and accelerates the search. In Holmberg (1994), a comparison of the performance of different Lagrangian relaxation approximations applied to the Benders master problem was conducted.

The third category encompasses strategies that aim to improve, or strengthen, the cut-generation process. In Magnanti and Wong (1981), the authors were the first to propose to generate nondominated optimality (*Pareto-optimal*) cuts whenever, for a given master-problem feasible solution, multiple optimal solutions (i.e., extreme points) are associated to the dual subproblem (scenario subproblems in the stochastic case). An additional dual subproblem instantiated using a core point of the master was solved to identify a nondominated cut. The addition of Pareto-optimal cuts greatly improved the value of the lower bound obtained by the algorithm. An improved approach was developed in Papadakos (2008), where the author demonstrated how alternative points can be used as proxies for core ones in the single subproblem solved to produce Pareto-optimal cuts. A different strategy to generate nondominated cuts was proposed in Fischetti, Salvagnin, and Zanette (2010) that reformulates the subproblem as a feasibility problem where cuts, both optimality and feasibility, are obtained by searching for minimal infeasible subsystems. In Sherali and Lunday (2013), these strategies were further enhanced to generate maximal nondominated cuts.

The Benders cut-generation process can also be strengthened by adding other valid inequalities to complement the feasibility and optimality cuts. Such enhancements often require the development of cutting planes that are problem specific. For example, in the context of solving stochastic network design problems, in Shen, You, and Ma (2017) and Rahmaniani et al. (2018), the authors showed how valid inequalities, which are based on network connectivity and capacity requirements, can be added to the master problem to further improve the quality of the lower bound that is obtained. New forms of optimality cuts have also been proposed in the context of the L-shaped method. Angulo, Ahmed, and Dey (2016) developed a general framework to produce a set of new valid optimality cuts for the integer L-shaped method that, at any given iteration of the method, explicitly consider the pool of feasible solutions encountered through the search process (and not just the current solution to be eliminated). Such cuts provide a better approximation of the expected recourse cost of the first stage solutions.

Rahmaniani et al. (2020) showed how the feasibility and optimality cuts generated by the Benders method can be strengthened by including copies of the master variables in the scenario subproblems. This is obtained by applying Lagrangian relaxation to the nonanticipativity constraints. A hybrid solution approach that leverages the strengths of both Benders decomposition and Lagrange relaxation is thus proposed. Recently, in Pay and Song (2020) and van Ackooij, de Oliveira, and Song (2018), the scenario partition aggregation strategy developed in Song and Luedtke (2015) was used to improve the cut-generation process of decomposition methods applied to solve stochastic programs. Specifically, the authors showed that, when applying either Benders or level decomposition, the adaptive scenario partition aggregations can be utilized to compute coarse optimality cuts. Considering that such cuts do not require the full set of scenario subproblems to be evaluated, they can be obtained more rapidly.

The fourth category groups the methods that adjust the decomposition strategy itself. In the case of the L-shaped algorithm when applied to solve two-stage stochastic integer programs, this category groups those methods that included information from the scenario subproblems in the master problem as a means to strengthen its formulation. In the context of solving network capacity planning problems under demand uncertainty, in MirHassani et al. (2000) and Bihlmaier, Koberstein, and Obst (2009), it was proposed to retain a single scenario subproblem. In both cases, it was numerically shown that choosing the scenario for which the total demand was highest produced the best results. In Liu, Ferris, and Zhao (2014), a feasibility checker model was developed to assess the challenge in obtaining a feasible solution to the master with respect to each scenario subproblem. Subproblems thus identified as "difficult" were then included in the master. Finally, in Batun et al. (2011), the authors showed that the mean value scenario could be used in the master formulation to provide a valid lower bound on the recourse cost of first stage solutions, which proved to be instrumental to successfully apply the L-shaped method on the problem of operating room scheduling when the surgery durations are stochastic.

Two general conclusions may be drawn from reviewing the field. First, there is not as yet a generally accepted design for highly performing Benders decomposition. As the results of this paper show, there is still significant room for improvement. Second, Benders decomposition has been traditionally applied to two-stage stochastic problems by involving in the projection the first stage decisions only. Yet, as illustrated by the very few contributions making up the fourth category above, including scenario information in the master formulation appears to be a distinctive advantage in mitigating the drawbacks of the L-shaped algorithm. Moreover, the significant and steady advances in computing and algorithmic power for MIPs, reflected in the efficiency of off-the-shelf solvers, now enable such decomposition strategies to be successfully implemented. For a different perspective on using this increased efficiency in enhancing Benders, Fischetti, Ljubić, and Sinnl (2017) alleviated the worry of making the master difficult to address by adding such information.

We therefore focus on the fourth type of enhancement for the L-shaped method and propose the first comprehensive PBD methodology. It generalizes the previous work in the area and offers a formal framework supporting the development of particular strategies. It also proposes a general methodology to apply these strategies for the L-shaped method. As supported by the results of extensive numerical experiments, the methodology is extremely efficient at boosting the performance of the L-shaped algorithm. The next section details the proposed PBD methodology, associated strategies, and implementation method.

4. Partial Benders Decomposition

We present in this section the proposed PBD method and the methodological framework to apply it to two-stage stochastic programs. We begin by defining a modified formulation for the master problem (Section 4.1). The presentation of the two general PBD strategies we propose, together with the underlying theoretical results supporting them, follows in Section 4.2 for scenario-retention strategies and in Section 4.3 for the scenario-creation one. Hybrid strategies combining the previous ones are introduced in Section 4.4, while Section 4.5 is dedicated to defining the MIPs used to implement the proposed strategies. Finally, it should be noted that the proofs of the theoretical results that support the PBD strategies have all been included in the appendices of the paper (i.e., in Appendices A and B for the results related to the scenario-retention and scenario-creation strategies, respectively).

4.1. Defining the Master Problem

As previously stated, PBD is based on the idea of including information related to the scenario subproblems in the master problem to strengthen its formulation. For the sake of generality, we define $\bar{S}$ as a set of scenarios, which may or may not be part of the original set S. Therefore, we distinguish the scenarios $s \in S \cap \bar{S}$, which are the original ones included in $\bar{S}$, from the scenarios $s' \in \bar{S}$, which are artificially generated (i.e. $s' \notin S$). We show in detail in Section 4.3 how to generate valid artificial scenarios for the original problem P: (1)–(4). For now, we note that validity can be achieved by taking a convex combination of scenarios $s \in S$. As such, the values $\alpha_s^{s'} \geq 0$ indicate how such a convex combination is taken. By using sets $\bar{S}$, $I \subseteq \mathcal{I}$, and $J \subseteq \mathcal{J}$, the relaxed master problem is reformulated as $BP(\bar{S},I,J)$:

$$(BP(\bar{S}, I, J)) \quad \min f^\top y + \sum_{s \in S \cap \bar{S}} p_s c^\top x^s + \sum_{s \in S \setminus \bar{S}} p_s z^s \tag{10}$$

subject to

$$Ay = b, \tag{11}$$
$$B^s y + Dx^s = d^s, \quad \forall s \in \bar{S}, \tag{12}$$
$$c^\top x^{s'} = \sum_{s \in S \setminus \bar{S}} \alpha_s^{s'} z^s, \quad \forall s' \in \bar{S}, \tag{13}$$
$$q^{i\top} d^s \leq q^{i\top} B^s y + z^s, \quad \forall i \in I, s \in S \setminus \bar{S}, \tag{14}$$
$$w^{j\top} d^s \leq w^{j\top} B^s y, \quad \forall j \in J, s \in S \setminus \bar{S}, \tag{15}$$
$$y \in \mathcal{Y},\ x^s \geq 0 \ \ \forall s \in \bar{S}, z^s \geq 0, \quad \forall s \in S \setminus \bar{S}. \tag{16}$$

There are several differences between the traditional relaxed master problem $BP(I,J)$ and $BP(\bar{S},I,J)$. First, the subproblems associated with the original scenarios included in set $\bar{S}$ (i.e., $s \in S \cap \bar{S}$) are retained in the master formulation. Therefore, the law of total expectation entails the expected recourse cost of a first stage solution y to be expressed as $\mathbb{E}(z^s(y) \mid s \in S \cap \bar{S}) \times p(S \cap \bar{S}) + \mathbb{E}(z^s(y) \mid s \in S \setminus \bar{S}) \times p(S \setminus \bar{S})$, where $p(S \cap \bar{S})$ and $p(S \setminus \bar{S})$ represent the probabilities of observing in the second stage a scenario in $S \cap \bar{S}$ and $S \setminus \bar{S}$, respectively, and $\mathbb{E}(\cdot)$ defines the expectation operator. When $S \cap \bar{S} \neq \emptyset$, the term $\mathbb{E}(z^s(y) \mid s \in S \cap \bar{S}) \times p(S \cap \bar{S})$ remains unchanged in the formulation of the master problem as defined in the objective function (10). Only the term $\mathbb{E}(z^s(y) \mid s \in S \setminus \bar{S}) \times p(S \setminus \bar{S})$ is dualized and then relaxed, thus reducing the number of optimality and feasibility cuts included in (14) and (15), respectively. Therefore, the approximation of the expected recourse cost provided by $BP(\bar{S},I,J)$ is more accurate than the one defined in $BP(I,J)$. In addition, greater stability with respect to the obtained solutions (and optimality cuts) is also anticipated.

The second difference between $BP(\bar{S},I,J)$ and $BP(I,J)$ is related to the inclusion of Constraints (12) in $BP(\bar{S},I,J)$. These constraints ensure that all solutions to the master problem are feasible with respect to the scenario subproblems associated to $s \in \bar{S}$, which eliminates the need to generate feasibility cuts for the scenario subproblems $s \in S \cap \bar{S}$. Furthermore, by retaining these linking constraints in the formulation of the master, valid inequalities for the polyhedron $P_s = \{(x^s, y) : Ay = b, B^s y + Dx^s = d^s, y \in \mathcal{Y}, x^s \geq 0\}$ can be added to $BP(\bar{S},I,J)$ for each $s \in \bar{S}$ to further strengthen the model. As for the artificially generated scenarios ($s' \in \bar{S}$ and $s' \notin S$), additional valid equations can be added to the master problem in the form of (13). These constraints provide a direct link in the master problem between the variables $x^{s'}$ and the variables z^s, for $s \in S \setminus \bar{S}$, whose values provide the approximation of the expected recourse cost. As shown in Section 4.3, the inclusion of Constraints (13) improves the quality of this approximation.

As outlined above, $BP(\bar{S},I,J)$ provides numerous advantages over $BP(I,J)$ when applying the L-shaped method to problem P. Moreover, previous studies have numerically pointed to the efficiency of this approach (MirHassani et al. 2000; Bihlmaier, Koberstein, and Obst 2009; Batun et al. 2011; Liu, Ferris, and Zhao 2014). We now focus on the open question of how to apply PBD in a general context and propose a theoretical framework to do so. To develop this framework, we begin by defining a set of general strategies to form set $\bar{S}$, through either the inclusion of original scenarios (Section 4.2) or the creation of valid artificial scenarios (Section 4.3).

4.2. Scenario-Retention Strategies

The strategies developed in this subsection are used to guide the selection of the scenarios $s \in S$ to add to $\bar{S}$ and, thus, retain their associated subproblems in the master formulation. Retaining a scenario from S removes the need to generate feasibility cuts for its subproblem. We go further and propose two strategies that can also obviate the need for generating feasibility cuts for scenarios that are not retained, that is, scenarios in $S \setminus \bar{S}$.

The first strategy, which we call *row covering*, adds scenarios to $\bar{S}$ that collectively dominate those in $S \setminus \bar{S}$ the most. The second strategy, which we call *convex hull*, seeks to choose scenarios for $\bar{S}$ that best approximate the uncertainty in the parameter values of the entire stochastic problem. We next define these strategies precisely, proving in both cases how to ensure feasibility in nonretained scenario subproblems (Propositions 1 and 2).

Row Covering. Recalling that $B^s \in \Re^{m_2 \times n_1}, \forall s \in S$, let us first define the *covering* concept.

Definition 1. Consider two scenarios: $s \in \bar{S}$ and $s' \in S \setminus \bar{S}$. We state that *s covers s′* if $\forall y \in \mathcal{Y}$ such that there exists a feasible solution to the subproblem $SP(y)_s$ there exists a feasible solution to the subproblem $SP(y)_{s'}$.

In the next proposition, we use weak duality to show how we can characterize when one scenario covers another.

Proposition 1. *Consider $s \in \bar{S}$ and $s' \in S \setminus \bar{S}$ such that $d_l^s \geq d_l^{s'}$ and $B_l^{s'\top} y \geq B_l^{s\top} y$, $\forall y \in \mathcal{Y}$ for all row indices $l = 1, \ldots, m_2$. If ($\bar{y}$; $\bar{x}^s, \forall s \in \bar{S}$; $\bar{z}^s, \forall s \in S \setminus \bar{S}$) is a feasible solution to model $BP(\bar{S},I,J)$, then $w^{j\top} d^{s'} \leq w^{j\top} B^{s'} \bar{y}$, $\forall j \in \mathcal{J}$, such that $w^j \geq 0$.*

We note that if $y \geq 0$, $\forall y \in \mathcal{Y}$, then the condition $B_l^{s'\top} y \geq B_l^{s\top} y$, $\forall y \in \mathcal{Y}$, is equivalent to requiring that $B_{li}^{s'} \geq B_{li}^s$, $\forall i = 1, \ldots, n_1$. In the specific application we study in our computational analysis, the same characterization of when one scenario covers another is true, but the presumption that $w^j \geq 0$ does not. As such, we present an alternate proof there.

Proposition 1 illustrates how to ensure feasibility in specific nonretained scenario subproblems. If a scenario $s' \in S \setminus \bar{S}$ is covered by a scenario $s \in \bar{S}$, then any first stage solution $\bar{y}$ to $BP(\bar{S},I,J)$ is such that a subset of feasibility cuts are necessarily enforced. Therefore, the more scenarios in $S \setminus \bar{S}$ are covered by scenarios in $\bar{S}$, the less feasibility cuts need to be generated. While it is unlikely that one scenario will cover another, Proposition 1 does suggest a way to choose the set $\bar{S}$. Namely, the scenarios in $\bar{S}$ should *collectively* cover those in $S \setminus \bar{S}$ as much as possible.

Convex Hull. Alternatively we can pursue a strategy that is similar to the pursuit of convex hulls in integer programming, as having a representation of the convex hull of the feasible region reduces the complexity of solving the resulting integer program. In this approach, the scenarios added to $\bar{S}$ are the ones that include in their associated convex hull the scenarios in $S \setminus \bar{S}$. However, exactly representing a scenario in $S \setminus \bar{S}$ is not necessary. Instead, we seek to retain in $\bar{S}$ scenarios whose convex combinations exhibit dominance relationships with scenarios in $S \setminus \bar{S}$. To simplify the presentation of the present strategy, it is assumed here that $\bar{S} \subseteq S$. Proposition 2 shows the value of this idea in the present context.

Proposition 2. *Consider $\bar{S} \subseteq S$ and $s' \in S \setminus \bar{S}$ such that $\exists\, \alpha_s^{s'} \geq 0, s \in \bar{S}$: $\sum_{s \in \bar{S}} \alpha_s^{s'} = 1$, $\sum_{s \in \bar{S}} \alpha_s^{s'} d^s \geq d^{s'}$, and $\sum_{s \in \bar{S}} \alpha_s^{s'} B^s \leq B^{s'}$. If ($\bar{y}$; $\bar{x}^s, \forall s \in \bar{S}$; $\bar{z}^s, \forall s \in S \setminus \bar{S}$) is a feasible solution to model $BP(\bar{S},I,J)$, then $w^{j\top} d^{s'} \leq w^{j\top} B^{s'} \bar{y}, \forall j \in \mathcal{J}$.*

Thus, if there exists a set of weights $\alpha_s^{s'}$ with which one can express a convex combination of the vectors $d^s, s \in \bar{S}$, that dominates the vector $d^{s'}$, as well as a convex combination of the matrices $B^s, s \in \bar{S}$, that is dominated by the matrix $B^{s'}$, then a feasible solution $\bar{y}$ to $BP(\bar{S},I,J)$ necessarily induces a feasible scenario subproblem for s' (i.e., $DSP(\bar{y})_{s'}$ is bounded). In this case, there will be no feasibility cuts generated for $SP(y)_{s'}$. Therefore, the more scenarios in $S \setminus \bar{S}$ wherein such convex combinations can be formed from the scenarios in $\bar{S}$, the fewer feasibility cuts need to be generated. Such a condition may be hard to impose. However, as in the case of Proposition 1, it again suggests a criterion that can be used to choose the set $\bar{S}$.

4.3. Scenario-Creation Strategy

In the previous section, we presented two strategies designed to choose the scenarios of set S to be retained in $\bar{S}$. These strategies are based on the principle of reducing the need to generate feasibility cuts for the scenarios used in the decomposition of the stochastic problem (i.e., $S \setminus \bar{S}$). We now propose a strategy whose objective is instead to improve the quality of the lower bound provided by the projection variables z^s for the expected recourse cost. Following this strategy, we create artificial scenarios $s' \notin S$ such that the inclusion of subproblems defined on s' (with associated variables $x^{s'}$, matrix $B^{s'}$, and vector $d^{s'}$) impacts the values of the projection variables.

To define the present strategy, the first step is to show how valid artificial scenarios $s' \notin S$ can be obtained for the original stochastic model (1)–(4). To do so, we develop the strategy assuming that a single scenario s' is created. The general case, where an arbitrary number of such scenarios are created, is a straightforward extension of the presented results. Let $P(s')$ be the original problem P for which a subproblem defined on a scenario s' is added, and let $v^\star_{s'}$ be its associated optimal value. Problem $P(s')$ is written as follows:

$$v^\star_{s'} = \min f^\top y + \sum_{s\in S} p_s c^\top x^s \tag{17}$$

subject to

$$Ay = b, \tag{18}$$
$$B^s y + Dx^s = d^s, \quad \forall s \in S, \tag{19}$$
$$B^{s'} y + Dx^{s'} = d^{s'}, \tag{20}$$
$$c^\top x^{s'} = \sum_{s\in S} \alpha_s^{s'} c^\top x^s, \tag{21}$$
$$y \in \mathcal{Y},\ x^s \ge 0, \quad \forall s \in S, \tag{22}$$
$$x^{s'} \ge 0. \tag{23}$$

We now state the conditions by which the vectors $d^{s'}$ and the matrix $B^{s'}$ can be created such that the optimal value of $P(s')$ is equal to the optimal value of P.

Lemma 1. *Let values $\alpha_s^{s'} \ge 0, s \in S$, be such that $\sum_{s\in S} \alpha_s^{s'} = 1$. If $d^{s'} = \sum_{s\in S} \alpha_s^{s'} d^s$ and $B^{s'} = \sum_{s\in S} \alpha_s^{s'} B^s$, then $v^\star_{s'} = v^\star$.*

As stated in Lemma 1, a subproblem s' that is defined as a convex combination of the subproblems $s \in S$ can be added to P, through the inclusion of Constraints (20) and (21), without modifying its optimal value. As a special case, s' can be set as the mean value scenario, which was the idea originally proposed in Batun et al. (2011). However, this result remains true for any convex combination considered. Lemma 1 is also a direct result of applying aggregation on both the scenario constraints and variables. We next investigate the overall theoretical benefits that are associated with applying this strategy to strengthen the master problem used in the L-shaped method.

Let $BP(y, I, J)$ and $BP(y, \bar{S}, I, J)$ define the restrictions obtained from $BP(I, J)$ and $BP(\bar{S},I,J)$ when the first stage variables y are fixed. Given $\bar{y}$, a feasible first stage solution for (1)–(4) (i.e., $\bar{y} \in \{y \in \mathcal{Y} \mid Ay = b, w^{j\top} d^s \le w^{j\top} B^s y, \forall j \in \mathcal{J}, s \in S\}$), we let $\bar{z}^s$, $s \in S$, be the optimal solution to $BP(\bar{y}, I, J)$. We next make an observation regarding this solution.

Lemma 2. We have $\bar{z}^s = \max\{0, q^{i_s^\star\top} d^s - q^{i_s^\star\top} B^s \bar{y}\}$, *where $i_s^\star \in \arg\max_{i\in I}\{q^{i\top} d^s - q^{i\top} B^s \bar{y}, \forall s \in S\}$.*

We now show a relationship between the lower bounds provided for the expected recourse cost of solution $\bar{y}$ when an artificial scenario s' is used in the relaxed master problem versus not used. In order to simplify the presentation of the results, we again assume that a single artificial scenario is created and that no scenarios from S are retained to obtain $BP(\bar{S},I,J)$, $\bar{S} = \{s'\}$. Our aim is to assess the opportunity loss of not including the subproblem s' in the master problem. Therefore, the following result holds.

Proposition 3. *Let $\bar{z}^s$, $s \in S$, and $\bar{x}_A^{s'}$, $\bar{z}_A^s$, $s \in S$, be optimal solutions to $BP(\bar{y}, I, J)$ and $BP(\bar{y}, \bar{S}, I, J)$, respectively. Then $\bar{z}_A^s \ge \bar{z}^s$, $\forall s \in S$.*

Given Proposition 3, it follows that the inclusion of an artificial scenario in $\bar{S}$ implies that, for all feasible first stage solutions to (1)–(4), the lower bound provided by $BP(\bar{S}, I, J)$ on the recourse cost value associated to each subproblem $s \in S$ is at least as great as the one obtained using $BP(I, J)$. We next investigate the actual value of the opportunity loss of not using the scenario subproblem s'.

For a given first stage feasible solution $\bar{y}$, let $\Delta_A(\bar{y}, I, J) = \sum_{s\in S} p_s(\bar{z}_A^s - \bar{z}^s)$ be the value of the overall improvement in the recourse cost lower bound when the artificial scenario subproblem is included in the master formulation. Specifically, we focus on solutions $\bar{y}$ for which, in the optimal solution to $BP(\bar{y}, \bar{S}, I, J)$, all variables z^s, $s \in S$, cannot be simultaneously set to the lower bound values derived from either the optimality cuts associated to set I or the nonnegativity requirements. Considering Proposition 3, such cases necessarily entail that $\Delta_A(\bar{y}, I, J) > 0$. Furthermore, all feasible first stage solutions $\bar{y}$ for which $\Delta_A(\bar{y}, I, J) > 0$ can be characterized using the condition that $c^\top \bar{x}_A^{s'} \neq \sum_{s\in S} \alpha_s^{s'} \bar{z}^s$. The following proposition thus defines the value of the overall improvement.

Proposition 4. *Let $\bar{y}$ be a feasible first stage solution, and let I and J be the subsets of extreme points and extreme rays, respectively. If $c^\top \bar{x}_A^{s'} \neq \sum_{s\in S}\alpha_s^{s'}\bar{z}^s$, then $\Delta_A(\bar{y},I,J)=\frac{p^{\tilde{s}}}{\alpha_{\tilde{s}}^{s'}}(c^\top \bar{x}_A^{s'} - \sum_{s\in S}\alpha_s^{s'}\bar{z}^s)$, where $\tilde{s}\in \arg\min_{s\in S}\{\frac{p^s}{\alpha_s^{s'}}\}$.*

4.4. Hybrid Strategies

The previous subsections proposed two categories of PBD strategies: (1) those that retain scenarios from the set of scenarios (S) defining the instance, and (2) those that create an artificial scenario by taking a convex combination of the scenarios in S. However, one can also perform hybrid strategies that both retain scenarios from S and create an artificial scenario with those not retained. In the next section, we present MIPs for implementing these different (hybrid and pure) PBD strategies.

4.5. Implementing the Decomposition Strategies

Having outlined the strategies we employ for strengthening the master problem with scenario information, we next describe how we implement them. Specifically, we present two MIPs. The first one will support both pure and hybrid strategies for when scenarios are to be retained through the row covering strategy; the second is for when the convex hull strategy is used to retain scenarios. Both MIPs take the following two parameters: (1) $R < |S|$ (an integer), which indicates the number of scenarios to retain, and (2) C (binary) which takes the value 1 (0) when an artificial scenario is (is not) to be created. We note that these MIPs assume that the probabilities associated with each scenario are the same, that is, $p_s = p_{\tilde{s}}$, $\forall s,\tilde{s}\in S$, and $s\neq \tilde{s}$. A MIP can be formulated that chooses the weights $\alpha_s^{s'}$ associated with scenarios that are not retained. However, preliminary experiments indicated it was best to use $\alpha_s^{s'} = p/[p(|S|-R)] = 1/(|S|-R)$. As such, we enforce this condition in the MIPs presented next.

Row Covering-Based Strategies. The first MIP we present seeks to choose which scenarios from S to add to $\bar{S}$ and (potentially) creates an artificial scenario out of those not retained in order to best cover scenarios in $S\setminus\bar{S}$. The MIP makes these choices jointly (instead of first determining which to retain and then creating an artificial scenario from the remaining) because the validity of the results regarding the benefits of covering a scenario with one that is retained does not depend on the retained scenario being from S. The underlying motivation for solving a MIP to implement this strategy is the observation that retaining two scenarios s^1, s^2 that cover the same nonretained scenario is unnecessary. As such, the MIP seeks to maximize the number of distinct scenarios covered.

We first define the parameters of the MIP that encode when scenario $\tilde{s}\in S$ covers a different scenario, $s\in S$. To that effect we define the values $\delta_l^{\tilde{s}s}$, $\tilde{s},s\in S$, such that $\tilde{s}\neq s$ and $l=1,\dots,m_2$, as follows:

$$\delta_l^{\tilde{s}s} = \begin{cases} 1 & \text{if } d_l^{\tilde{s}} \geq d_l^s, \\ 0 & \text{otherwise.} \end{cases}$$

These values represent the degree to which $\tilde{s}$ covers s with respect to the right-hand-side vector, d. Regarding the matrices B^s, we next define the values $\gamma_{li}^{\tilde{s}s}$, $\forall \tilde{s},s\in S$, such that $\tilde{s}\neq s$ and $l=1,\dots,m_2$, $i=1,\dots,n_1$, as follows:

$$\gamma_{li}^{\tilde{s}s} = \begin{cases} 1 & \text{if } B_{li}^{\tilde{s}} \leq B_{li}^s, \\ 0 & \text{otherwise.} \end{cases}$$

Fundamentally, we can say that scenario $\tilde{s}$ covers scenario s with respect to row l of the constraint matrix if the values $\delta_l^{\tilde{s}s}, \gamma_{li}^{\tilde{s}s}$, $i=1,\dots,n_1$, are 1.

To formulate the integer program proposed, we first define the binary variables r_s, $s\in S$, which express whether scenario $s\in S$ is retained for inclusion in $\bar{S}$. Regarding the artificial scenario created, we use the continuous variable $\alpha_s^{s'}$ to represent the weight associated to scenario $s\in S$ when creating the artificial scenario s'. (Recall that we do not know a priori which scenarios should be used to construct the artificial scenario.) Similarly, we use the continuous variables $d_l^{s'}, l=1,\dots,m_2$, and $B_{li}^{s'}, l=1,\dots,m_2;\ i=1,\dots,n_1$, to represent the d vector and B matrix of the artificial scenario.

To model how chosen or artificial scenarios cover those that are not chosen (in $S\setminus\bar{S}$), we define the binary variables b_l^s and $\bar{b}_l^s$, $s\in S$ and $l=1,\dots,m_2$. Variable b_l^s indicates whether row index l of the vector d for scenario s is covered by a scenario in $\bar{S}$ (either a retained scenario from S or the artificial scenario) and $\bar{b}_l^s$ indicates that it is covered by the artificial scenario. Whereas these binary variables model the degree to which right-hand-side vectors are covered, we next define binary variables to model the degree to which B matrices of nonretained scenarios are covered. Specifically, we define the binary variables c_{li}^s and $\bar{c}_{li}^s$, $s\in S, l=1,\dots,m_2, i=1,\dots,n_1$, with c_{li}^s indicating whether B_{li}^s is covered by a scenario in $\bar{S}$ and $\bar{c}_{li}^s$ indicating that it is covered by the artificial scenario.

Finally, we define continuous variables to measure the degree to which the artificial scenario does not cover elements of nonretained scenarios. Specifically, we define the continuous variables e_l^s, $s\in S, l=1,\dots,m_2$, to measure the degree to which the artificial scenario does not cover the value d_l^s of a nonretained scenario. We also define the continuous variables f_{li}^s, $s\in S, l=1,\dots,m_2, i=1,\dots,n_1$, to measure the degree to which the artificial scenario does not cover the value B_{li}^s of a nonretained scenario.

With the above-defined variables and parameters, the following MIP is solved to obtain $\bar{S}$:

$$\max \sum_{s\in S}\sum_{l=1}^{m_2} b_l^s + \sum_{s\in S}\sum_{l=1}^{m_2}\sum_{i=1}^{n_1} c_{li}^s$$

subject to

$$b_l^s \le \sum_{\tilde{s}\in S} \delta_l^{\tilde{s}s} r_{\tilde{s}} + \bar{b}_l^s,\ l=1,\dots,m_2, \forall s\in S, \tag{24}$$

$$c_{li}^s \le \sum_{\tilde{s}\in S} \gamma_{li}^{\tilde{s}s} r_{\tilde{s}} + \bar{c}_{li}^s, l=1,\dots,m_2,\ i=1,\dots,n_1, \forall s\in S, \tag{25}$$

$$d_l^{s'} = \sum_{s\in S} d_l^s \alpha_s^{s'},\quad l=1,\dots,m_2, \tag{26}$$

$$d_l^s \le d_l^{s'} + e_l^s \quad l=1,\dots,m_2,\ \forall s\in S, \tag{27}$$

$$\bar{b}_l^s \le 1 - \frac{e_l^s}{d_l^s} \quad l=1,\dots,m_2,\ \forall s\in S \mid d_l^s > 0, \tag{28}$$

$$B_{li}^{s'} = \sum_{s\in S} B_{li}^s \alpha_s^{s'} \quad l=1,\dots,m_2,\ i=1,\dots,n_1, \tag{29}$$

$$B_{li}^{s'} - f_{li}^s \le B_{li}^s \quad l=1,\dots,m_2,\ i=1,\dots,n_1, \forall s\in S, \tag{30}$$

$$\bar{c}_{li}^s \le 1 - \frac{f_{li}^s}{B_{li}^s} \quad l=1,\dots,m_2,\ i=1,\dots,n_1, \forall s\in S \mid B_{li}^s > 0, \tag{31}$$

$$\sum_{s\in S} r_s = R, \tag{32}$$

$$\alpha_s^{s'} \le C\frac{1}{(|S|-R)}(1-r_s) \quad \forall s\in S, \tag{33}$$

$$\alpha_s^{s'} \ge C\left(\frac{1}{(|S|-R)} - r_s\right) \quad \forall s\in S, \tag{34}$$

$$\sum_{s\in S} \alpha_s^{s'} = C, \tag{35}$$

$$\alpha_s^{s'} \ge 0, r_s \in \{0,1\}, \forall s\in S, \tag{36}$$

$$d_l^{s'} \ge 0, \quad l=1,\dots,m_2, \tag{37}$$

$$b_l^s \in \{0,1\}, \bar{b}_l^s \in \{0,1\}, 0 \le e_l^s \le d_l^s, \quad l=1,\dots,m_2, \forall s\in S, \tag{38}$$

$$B_{li}^{s'} \ge 0 \quad l=1,\dots,m_2, i=1,\dots,n_1, \tag{39}$$

$$c_{li}^s \in \{0,1\}, \bar{c}_{li}^s \in \{0,1\}, 0 \le f_{li}^s \le B_{li}^s,\\ l=1,\dots,m_2, i=1,\dots,n_1, \forall s\in S. \tag{40}$$

The objective of this optimization problem is to maximize the number of elements d_l^s and B_{li}^s that are covered by scenarios in $\bar{S}$ (either retained or artificial). Constraints (24) and the objective ensure that the variables b_l^s take on the value 1 if either a retained scenario $\tilde{s}$ is included in $\bar{S}$ such that $d_l^{\tilde{s}} \ge d_l^s$ or the artificial scenario is constructed in such a way that $d_l^{s'} \ge d_l^s$. Constraints (25) play a similar role albeit with respect to element B_{li}^s being covered by a retained or artificial scenario.

Constraints (26) calculate the d vector for the artificial scenario, whereas Constraints (27) calculate the amount (represented by e_l^s) by which the artificial scenario does not cover d_l^s. Note that without sacrificing the optimal solution to this MIP we can presume that $e_l^s \le d_l^s$. As such, Constraint (28) ensures that, when we do not have $d_l^{s'} \ge d_l^s$ (and thus $e_l^s > 0$), $\bar{b}_l^s$ must take on the value 0. (Recall that $\bar{b}_l^s$ is binary.) Constraints (29), (30), and (31) serve a similar purpose, albeit for determining $\bar{c}_{li}^s$.

Constraint (32) limits the number of scenarios from S that are retained to be at most R. Constraints (33) and (34) ensure that, when an artificial scenario is to be created, it is constructed from scenarios that are not retained. Finally, Constraint (35) ensures that, when an artificial scenario is created, it is a convex combination of scenarios from S. The remaining constraints define the domains of the decision variables.

The MIP just presented can be used to implement both pure and hybrid row covering strategies. By setting $C=0$, the MIP implements the pure row covering strategy, which we denote by RC($|\bar{S}|$). It can also be used to implement the hybrid row covering and scenario-creation strategy and in two ways. The first, which we refer to as Joint-RC($|\bar{S}|$)+A involves solving the MIP presented above, which jointly determines which scenarios to retain while explicitly recognizing the impact of the artificial scenario created from the scenarios that are not retained. The second, which we refer to as TwoStage-RC($|\bar{S}|$)+A involves solving the MIP presented above with $C=0$, meaning the impact of the artificial scenario is not recognized, and then creating the artificial scenario from the scenarios not retained.

Convex Hull-Based Strategies. The next MIP we present also seeks to, jointly, choose which scenarios from S to add to $\bar{S}$ as well as create an artificial scenario out of those not chosen. However, in this case, the MIP seeks to maximize the degree by which non-retained scenarios have the dominance relationships with a convex combination of scenarios in $\bar{S}$ outlined in Proposition 2.

To define the MIP, we again use the binary variable r_s to represent whether scenario s is retained for inclusion in $\bar{S}$. We note that a scenario can contribute to a convex combination that dominates the vector $d^{\tilde{s}}$ as well as a convex combination that is dominated by the matrix $B^{\tilde{s}}$ in one of two ways: (1) when it is retained, wherein we let the continuous variables $\alpha_s^{\tilde{s}}$ represent the weight used, and (2) when it is not retained but is used to create the artificial scenario. For this second way, we model the weight given the artificial scenario when approximating scenario $\tilde{s}$ with the continuous variable $\beta_{s'}^{\tilde{s}}$.

Regarding scenario $\tilde{s}$, we define the continuous variables $u_l^{\tilde{s}}, \tilde{s}\in S, l=1,\dots,m_2$, to represent element l of the convex combination for $\tilde{s}$ of the vectors $d^s, s\in\bar{S}$, that, by Proposition 2, we seek to have be at least as great as $d_l^{\tilde{s}}$. We then define the continuous variables $e_l^{\tilde{s}}$, $\tilde{s} \in S, l = 1, \ldots, m_2$, to measure the amount by which $d_l^{\tilde{s}}$ exceeds that convex combination of vectors $d^{\bar{s}}, s \in \bar{S}$. Similarly, we define continuous variables $v_{li}^{\tilde{s}}$, $\tilde{s} \in S$, $l = 1, \ldots, m_2$, $i = 1, \ldots, n_1$ to represent the (l, i) matrix element of the convex combination for $\tilde{s}$ of the matrices $B^s, s \in \bar{S}$, that, by Proposition 2, we seek to have be no greater than $B_{li}^{\tilde{s}}$. We then define the continuous variables $f_{li}^{\tilde{s}}, \tilde{s} \in S$, $l = 1, \ldots, m_2$, $i = 1, \ldots, n_1$, to compute the amount by which that convex combination of matrices $B^s, s \in \bar{S}$, exceeds $B_{li}^{\tilde{s}}$. As such, the MIP we solve to construct the set $\bar{S}$ is as follows:

$$\min \sum_{s\in S} \sum_{l=1}^{m_2} e_l^s + \sum_{s\in S} \sum_{l=1}^{m_2} \sum_{i=1}^{n_1} f_{li}^s$$

subject to

$$u_l^{\tilde{s}} = \sum_{s\in S} \alpha_s^{\tilde{s}} d_l^s, \; l = 1, \ldots, m_2, \; \forall \tilde{s} \in S, \tag{41}$$

$$v_{li}^{\tilde{s}} = \sum_{s\in S} \alpha_s^{\tilde{s}} B_{li}^s, \; l = 1, \ldots, m_2, i = 1, \ldots, n_1, \; \forall \tilde{s} \in S, \tag{42}$$

$$\alpha_s^{\tilde{s}} \le \frac{\beta_{s'}^{\tilde{s}}}{|S| - R} + r_s, \quad \forall s, \tilde{s} \in S, \tag{43}$$

$$\alpha_s^{\tilde{s}} \ge \frac{\beta_{s'}^{\tilde{s}}}{|S| - R} - r_s, \quad \forall s, \tilde{s} \in S, \tag{44}$$

$$\sum_{s\in S} \alpha_s^{\tilde{s}} = 1, \; \forall \tilde{s} \in S, \tag{45}$$

$$\sum_{s\in S} r_s = R, \tag{46}$$

$$e_l^{\tilde{s}} \ge d_l^{\tilde{s}} - u_l^{\tilde{s}}, \quad l = 1, \ldots, m_2, \; \forall \tilde{s} \in S, \tag{47}$$

$$f_{li}^{\tilde{s}} \ge v_{li}^{\tilde{s}} - B_{li}^{\tilde{s}}, \quad l = 1, \ldots, m_2, i = 1, \ldots, n_1, \; \forall \tilde{s} \in S, \tag{48}$$

$$r_s \in \{0, 1\}, \; \forall s \in S, \tag{49}$$

$$1 \ge \alpha_s^{\tilde{s}} \ge 0, \quad \forall s, \tilde{s} \in S, \tag{50}$$

$$C \ge \beta_{s'}^{\tilde{s}} \ge 0, \; \forall \tilde{s} \in S, \tag{51}$$

$$e_l^{\tilde{s}} \ge 0, \; u_l^{\tilde{s}} \in \Re, \; l = 1, \ldots, m_2, \; \forall \tilde{s} \in S, \tag{52}$$

$$f_{li}^{\tilde{s}} \ge 0, \; v_{li}^{\tilde{s}} \in \Re, \; l = 1, \ldots, m_2, i = 1, \ldots, n_1, \; \forall \tilde{s} \in S. \tag{53}$$

The objective is to minimize the total error associated with the representations $u_l^{\tilde{s}}$ and $v_{li}^{\tilde{s}}$ and the corresponding elements of $d_l^{\tilde{s}}$ and $B_{li}^{\tilde{s}}$ for the scenarios $\tilde{s} \in S \setminus \bar{S}$. The values of $u_l^{\tilde{s}}$ and $v_{li}^{\tilde{s}}$ are determined by Constraints (41) and (42), respectively. Constraints (43) and (44) ensure that a scenario is assigned the appropriate weight based upon whether it is retained or used to create the artificial scenario. Constraints (45) ensure that a convex combination of retained scenarios is created for each $\tilde{s} \in S$. Constraint (46) limits the number of scenarios that are included in $\bar{S}$ to be at most R. Constraints (47), coupled with the objective, define that $e_l^{s'} = \max\{d_l^{s'} - u_l^{s'}, 0\}$.3 Similarly, Constraints (48), coupled with the objective, define that $f_{li}^{s'} = \max\{v_{li}^{s'} - B_{li}^{s'}, 0\}$. Finally, Constraints (49), (50), (51), (52), and (53) define the decision variables and their domains.

The MIP just presented can be used to implement both pure and hybrid convex hull strategies. By setting $C = 0$, the MIP implements the pure convex hull strategy, which we denote by CH($|\bar{S}|$). It can also be used to implement the hybrid convex hull and scenario-creation strategy, which we refer to as CH($|\bar{S}|$)+A.

Random-Based Strategies. Lastly, one can consider implementing pure and hybrid strategies that are based on randomly determining which scenarios to retain. We refer to the pure scenario retention strategy that randomly determines which scenarios to retain as Rand($|\bar{S}|$). Similarly, we also consider hybrid strategies wherein the scenarios retained are chosen randomly, which we refer to as Rand($|\bar{S}|$)+A.

5. Experimental Design

To assess the computational advantages that can be achieved by using PBD and to complement the general analysis presented in the previous section, we now apply the strategy to solve an important class of stochastic optimization problems (i.e., stochastic fixed charge multicommodity network design problems) and perform an extensive computational study. In this section, we describe the specific problem class, how the algorithms were implemented, and the characteristics of the test instances.

5.1. Stochastic Network Design

We study the effectiveness of the PBD strategies presented on the fixed charge multicommodity network design problem wherein there are two sets of stochastic parameters: (1) the demand associated with each commodity and (2) the capacity associated with each potential arc. These problems naturally appear in many applications, for example, they are an essential part of the optimization methodologies used to perform supply chain management under uncertainty (Klibi, Martel, and Guitouni 2010; Klibi and Martel 2012). Moreover, such models are notoriously hard to solve, and the exact methods that have been used to solve them have required decomposition strategies to be applied (Crainic, Frangioni, and Gendron 2001; Crainic et al. 2011).

We consider a directed network with node set N, arc set A, commodity set K, and scenario set S. Regarding the arcs adjacent to a node, we let $N^+(i)$ represent the set of arcs emanating from i (i.e., $N^+(i) = \{(i,j) \in A\}$) and let $N^-(i)$ represent the arcs terminating at node i (i.e., $N^-(i) = \{(j,i) \in A\}$). Each commodity k must be routed from an origin node, o_k, to a destination node, d_k. We consider the general setting where both capacities and demands are stochastic. In such a model, as both demands and capacity are not known a priori, neither are revealed until the second stage of the problem. These parameters are thus indexed by scenario. The formulation of the stochastic fixed charge multicommodity network design problem, *CMND*(S), is

$$\min \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{s\in S} p_s \left(\sum_{k\in K} \sum_{(i,j)\in A} c^k_{ij} x^{ks}_{ij} \right) \tag{54}$$

subject to

$$\sum_{j\in N^+(i)} x^{ks}_{ij} - \sum_{j\in N^-(i)} x^{ks}_{ji} = d^{ks}_i, \quad \forall i \in N, k \in K, s \in S, \tag{55}$$

$$\sum_{k\in K} x^{ks}_{ij} \le u^s_{ij} y_{ij}, \quad \forall (i,j) \in A, s \in S, \tag{56}$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A, \tag{57}$$

$$x^{ks}_{ij} \ge 0, \quad \forall (i,j) \in A, k \in K, s \in S, \tag{58}$$

where y_{ij} indicates whether arc $(i,j) \in A$ is selected (i.e., installed in the network) in the first stage of the problem, and f_{ij} is the cost (often called the fixed charge) of including arc (i,j) in the network. In the second stage of the problem, the obtained network is used to flow the commodities to meet the observed demands. Variable x^{ks}_{ij} is the amount of the demand of commodity $k \in K$ that flows on arc (i,j), considering that scenario $s \in S$ is observed in the second stage of the problem, c^k_{ij} being the cost per unit of demand k flowed on arc (i,j). Constraints (55) are flow-conservation equations ensuring that each commodity's demand may be routed from its origin node to its destination node in each scenario s. Therefore, assuming that v^{ks} is the volume of commodity k in scenario s, the parameter d^{ks}_i is set to either v^{ks} if node i is the origin of the commodity k, $-v^{ks}$ if node i is the destination of the commodity k, or 0 otherwise. Constraints (56) guarantee that the same design is used in each scenario and that the arc capacity (u^s_{ij}) is not violated in any scenario. Finally, Constraints (57) and (58) impose the necessary integrality and nonnegativity requirements on the decision variables of the model.

Recall that we presented the decomposition strategies as being used in the course of solving the (more generally defined) problem with Objective (1) and Constraints (2), (3), and (4). We note that the stochastic parameters B^s and d^s in that problem correspond to the arc capacities, u^s_{ij}, and commodity demands, d^{ks}_i, in the *CMND*(S), respectively. Thus, in the *CMND*(S), scenario $\tilde{s}$ covers scenario s when it models smaller arc capacities ($u^{\tilde{s}} \le u^s$) and larger commodity demands ($d^{\tilde{s}} \ge d^s$). Thus, while Proposition 1 does not apply to the *CMND*(S) (the constraints associated with the vector d^s are equality constraints, and hence, the dual variables are not restricted to be nonnegative), the conclusion still holds. Namely, if y induces a design that enables the routing of commodity demands for scenario $\tilde{s}$, which in turn covers scenario s, then that design will also enable the routing of commodity demands for scenario s.

5.2. Benders Implementation

We have implemented a Benders algorithm that includes many of the enhancements developed for the original solution procedure. We implemented the *multicut* version of the L-shaped method (Birge and Louveaux 1988). Preliminary tests showed that this version of the method outperformed the original *single-cut* version (Van Slyke and Wets 1969) on the instances used. It has often been observed that the lack of structure in the master problem hampers the ability of a Benders implementation to solve instances in reasonable run times. Thus, problem-specific inequalities for the *CMND*(S) are added to further strengthen the formulation of the master problem. Specifically, we add inequalities of the form $\sum_{j\in N^+(i)} y_{ij} \ge 1$, where node i is the origin for some commodity k. Similar inequalities are added for destination nodes for commodities.

In addition, we also opted for the two-phase Benders solution approach proposed in (McDaniel and Devine 1977). Specifically, the first phase of our implementation solves the linear relaxation of Problem (54)–(58) via a Benders-based cutting plane approach. The cuts collected while solving the linear relaxation are used to strengthen the formulation of the master problem solved in the second phase, wherein the integrality Constraints (57) are reintroduced, and a Benders-type algorithm is again applied to produce an optimal solution to the *CMND*(S).

Regarding the second phase, P, an integer program, is solved via the search of a single branch-and-bound tree (via CPLEX). This is similar to what was done in Fischetti, Ljubić, and Sinnl (2016) and is sometimes referred to as "branch-and-Benders-cut" (Rahmaniani et al. 2017). In this search, in addition to the valid inequalities CPLEX generates at a node to render fractional solutions infeasible, optimality and feasibility cuts are added at nodes of the tree whenever integer feasible solutions are found. This approach is related to the strategy proposed in Geoffrion and Graves (1974) where suboptimal solutions are used to generate cuts. It is also similar to the hybrid method proposed in Hooker and Ottosson (2003), which combines Benders decomposition and constraint programming to solve a larger class of problems. Finally, as local branching was shown to speed up the execution of

Benders decomposition (Rei et al. 2009), we turn on CPLEX's implementation of local branching when solving a master problem in the second phase of the algorithm.

Regarding the generation of optimality and feasibility cuts, we have implemented the approach originally proposed in Magnanti and Wong (1981), guaranteeing that only nondominated cuts are added to the master problem. Both types of inequalities are included in the master formulation at the beginning of the first phase of the algorithm. Furthermore, it is well known that, when $d_i^{ks} < u_{ij}^s$, adding constraints of the form $x_{ij}^{ks} \le d^{ks} y_{ij}$ to (54)–(58) greatly strengthens the formulation. At the same time, even in a deterministic setting (where $|S| = 1$), there are often too many of these inequalities to add them beforehand. Consequently, it is necessary to add them dynamically in a cutting plane algorithm fashion (Nemhauser and Wolsey 1988). When PBD is applied, by retaining variables x_{ij}^{ks} associated with scenarios, we are able to dynamically add these inequalities when solving the linear relaxation of the $CMND(S)$. The collected inequalities $x_{ij}^{ks} \le d^{ks} y_{ij}, s \in \bar{S}$, are then kept in the master formulation when the second phase of the algorithm is performed.

5.3. Instances Used

For our computational study we consider seven instance classes (4–10) from the set of R instances seen in Crainic et al. (2011). Each class uses the same network, with the attributes of those networks (number of nodes, arcs, and commodities) given in Table 1. The five instances within a class differ with respect to their (increasing) ratio of fixed to variable costs and total demand to capacity. (The instances were originally proposed for the deterministic fixed charge multicommodity network design problem.) A detailed description of the instances can be found in Crainic, Frangioni, and Gendron (2001).

We generate four sets of scenarios for each given instance class that differ in the number of scenarios. Specifically, we generate sets with $|S| = 96, 128, 160, 192, 224$, or 256 scenarios. It is important to note that these sets define the largest instances of the $CMND(S)$ that have been solved in the field (see Crainic et al. 2011, Crainic, Hewitt, and Rei 2014). To generate scenarios, we used the algorithm presented in Høyland, Kaut, and Wallace (2003). In doing so, we presumed no correlation between the random variables. In summary, we test our strategies on 210 instances of varying structure and stochasticity.

Table 1. Instance Class Characteristics

Class	$\lvert N\rvert$	$\lvert A\rvert$	$\lvert K\rvert$
4	10	60	10
5	10	60	25
6	10	60	50
7	10	82	10
8	10	83	25
9	10	83	50
10	20	120	40

6. Computational Results

We seek to answer multiple questions with our computational study. We first seek to understand whether a partial decomposition should be used and, if so, which pure or hybrid strategy should be employed. We then seek to understand the source of the computational benefits (if any) associated with performing a PBD. Finally, we assess how the performance of a PBD correlates to the number of scenarios defining an instance. Throughout this section, we benchmark the partial decomposition strategies mentioned above against two available solution methods: (1) the branch-and-cut solver, CPLEX, and (2) the "automatic" Benders implementation present in CPLEX, which we refer to as CPLEX-Benders.

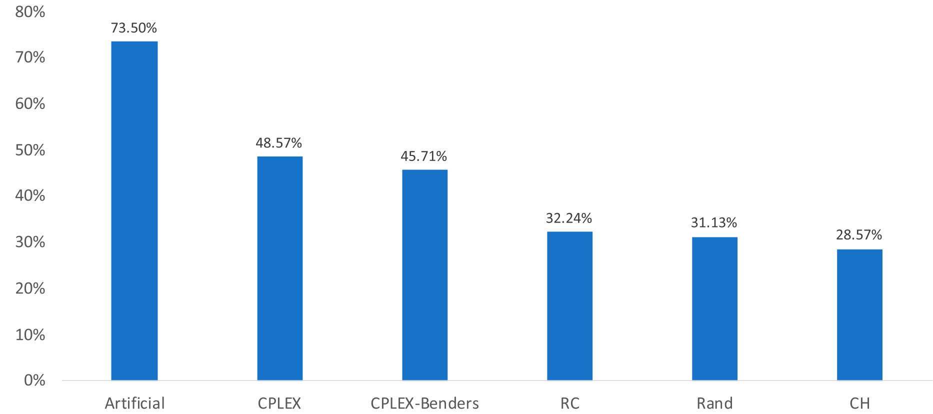
In all experiments, we executed our implementations of Benders decomposition on a machine with 64 Intel Xeon CPUs running at 2.30 GHz with 64 GB RAM. All linear integer programs and MIPs were solved with CPLEX 12.8. All algorithms for solving P were executed with an optimality gap tolerance of 1% and a time limit of two hours. All MIPs used to implement the hybrid strategies were solved to within an optimality gap tolerance of 1% and a ten-minute time limit. All computation times reported are in seconds.

When PBD is applied, the reported times include the computational effort that is necessary to solve the MIP to determine the set $\bar{S}$. For all but the convex hull-based partial decomposition strategies we consider including in the master problem between one and seven scenarios, either retained or created. For the convex hull-based strategies we consider between two and seven scenarios. For the strategies that involve randomness, we execute the strategy five times on each instance and report averages over those executions.

6.1. Benchmarking the Use of Partial Decomposition

We first focus on the ability of the pure PBD strategies, as well as our benchmark methods, to solve instances with 96 scenarios to within an optimality gap tolerance of 1%. We will consider larger numbers of scenarios later. We illustrate in Figure 1 the percentage of such solved instances, by strategy. In Table 2, we present for each strategy the average optimality gap at termination and the average amount of time until termination. For the strategies that involve scenario retention we present averages over executions of

Figure 1. (Color online) Pure Strategies, Percent of Instances Solved to Within 1%, $|S| = 96$



each strategy with different numbers of scenarios retained.

We see in these results that the scenario retention strategies RC, Rand, and CH perform worse than the benchmarks, CPLEX and CPLEX-Benders, but the scenario-creation strategy Artificial performs much better. Specifically, Artificial is able to solve far more instances and in less time. It is also able to produce solutions that are of provably higher quality than both the scenario-retention strategies and the benchmarks.

We next turn our attention to the hybrid strategies and consider the same statistics. Figure 2 is analogous to Figure 1, while Table 3 is analogous to Table 2. To facilitate comparison, we repeat results for Artificial and the two CPLEX-based benchmarks.

In these results, we see that, while the Artificial strategy outperformed our benchmarks and the scenario retention strategies did not, a combination of the two strategies yields the best performing method overall. However, as the convex hull and random-based hybrid strategies perform worse than the benchmarks, while the row covering-based hybrid strategies perform better, we clearly see that the manner in which the scenarios to retain are chosen is crucial to the performance of a hybrid strategy. Ultimately, we see that the row covering-based hybrid strategies are able to solve more instances in less time as well as yield solutions that are of provably higher quality. We also note that there appears to be negligible difference between the performances of the two row covering-based hybrid strategies.

Lastly, we recall that we ran each algorithmic strategy until it produced a primal solution and lower bound that were within a given optimality tolerance of 1%. In Table 4 we report the average optimality gap of instances that a given strategy was unable to solve. Note this means that the results reported in Table 4 for each strategy reflect averages over different instances and numbers of instances. Regardless, we see that, when the hybrid row covering-based strategies are unable to solve an instance, they produce a solution that is of provably higher quality than the two benchmarks.

Having analyzed the performance of the hybrid strategies in aggregate, we next study the sensitivity of their performance to the number of scenarios retained. In Figure 3(a), we report the percentage of instances solved to within a 1% optimality tolerance, by the number of scenarios retained. Figure 3(b) is similar, but reports the average time to termination.

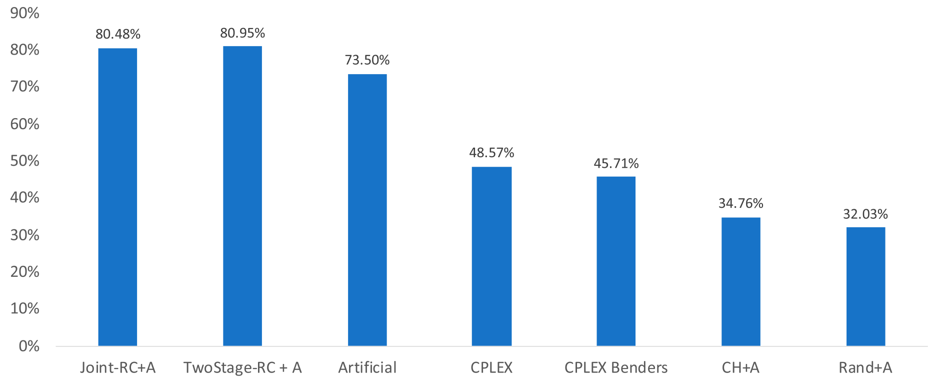
We see in these figures that the performance of each hybrid strategy does not depend significantly on the number of scenarios retained. However, there is a slight degradation of performance as the number of scenarios retained increases. We conclude from these figures that the performance of the hybrid strategies, particularly the best performing row covering-based hybrid strategies, is not particularly sensitive to the number of scenarios retained.

Next, we recall that many of the proposed partial decomposition strategies are implemented by solving MIPs. We first note that only 8% of the MIPs used to implement the CH strategies could be solved to within the desired 1% optimality gap in ten minutes. On average, the reported optimality gap at that ten-minute time limit was 11.23%. We hypothesize that

Table 2. Pure Strategies, $|S| = 96$

Strategy	Artificial	CPLEX	CPLEX-Benders	RC	Rand	CH
Time to termination	3,221.40	4,012.86	4,306.29	5,183.89	5,201.91	4,946.39
Optimality gap	4.13%	9.95%%	7.76%	8.07%	8.71%	11.94%

Figure 2. (Color online) Hybrid Strategies, Percent of Instances Solved to Within 1%, $|S| = 96$



the poor performance of the convex hull-based strategies may be attributed to these MIPs not yielding the best scenarios to retain.

Turning to the row covering-based hybrid strategies, we note that all MIPs were solved to within a 1% optimality gap. We report in Table 5 the time required to solve these MIPs to that tolerance. We see that the time needed to solve these MIPs is fairly small and does not depend greatly on the number of scenarios to retain. Also, we note that the solution times for the pure row covering strategy never exceeded five seconds.

We conclude based on the results in this section that the hybrid row covering-based hybrid strategies are the best performing and consistently outperform the two benchmarks. In addition, it is best to retain just a few scenarios when implementing such strategies.

6.2. Analyzing the Impact of Partial Decomposition

We next seek a fuller picture of the performance seen when performing partial decomposition. We begin by analyzing the convergence rate of the Benders algorithm. Recall that, in our Benders implementation, we first solve the linear programming relaxation of P to a tolerance of 1%. After doing so, we solve P, augmented with the cuts found while solving its linear relaxation, with a branch-and-bound-based algorithm wherein cuts are generated throughout the tree search. As such, we first analyze the impact of partial decomposition on each phase of the overall approach.

First, we illustrate in Figure 4 a comparison of the number of iterations required (on average) to solve the linear programming relaxation of P for each strategy. As noted previously, when a strategy involves retaining a given number of scenarios, we average over executions of that strategy with different numbers of retained scenarios. We see that the use of an artificial scenario, either on its own or as part of a hybrid strategy, often greatly speeds up convergence.

We next turn our attention to the convergence of the second, integer programming-based, phase of the Benders approach. Here, to measure convergence, we consider the average number of open and explored (i.e., pruned) nodes in the branch-and-bound tree at termination. We illustrate these averages by strategy in Figure 5 and see that the use of an artificial scenario often enables the method to converge by exploring much smaller branch-and-bound trees. These results illustrate that including in the master problem an artificial scenario can yield high-quality primal and dual bounds.

Another statistic related to convergence is the number of Benders cuts generated. As such, we next report in Figure 6 the number of feasibility and optimality cuts generated for different strategies. We note that we consider both the cuts generated when solving the linear programming relaxation of P (i.e., phase one) and when solving P via the branch-and-bound-based Benders implementation (i.e., phase two). We also note that the CPLEX implementation of Benders reports the total number of Benders cuts generated but not a breakdown into feasibility and optimality cuts. However, on average, CPLEX reported generating 26,291.91 Benders cuts in total.

Table 3. Hybrid Strategies, $|S| = 96$

Strategy	Joint-RC+A	TwoStage-RC+A	Artificial	CPLEX	CPLEX-Benders	CH+A	Rand+A
Time to termination	2,276.79	2,270.02	3,221.40	4,012.86	4,306.29	5,113.12	5,203.33
Optimality gap	1.34%	1.29%	4.13%	9.95%%	7.76%	7.72%	8.78%

Table 4. Optimality Gap of Instances Not Solved, $|S| = 96$

Strategy	CPLEX	CPLEX-Benders	Artificial	Joint-RC+A	TwoStage-RC + A
Optimality gap	19.23%	13.57%	4.40%	3.48%	3.25%

These results show that performing a PBD strategy yields a significant decrease in the number of feasibility and optimality cuts. We see that, while creating a single artificial scenario reduces the need to generate either kind of cut, the hybrid strategies do so even more. Specifically, the row covering-based hybrid strategies dramatically reduce the number of feasibility cuts generated, even when compared with the pure Artificial strategy.

To understand why performing a partial decomposition can have such an impact on the number of optimality cuts generated, we next measure, by strategy, the relative difference between the lower bound values associated with the expected recourse term at the first iteration of solving the linear programming relaxation (LPR) of P and when the LPR has been solved. We report these values in Table 6 and see that the row covering-based hybrid strategies closely

Figure 3. (Color online) Performance of Hybrid Strategies by $|\bar{S}|$

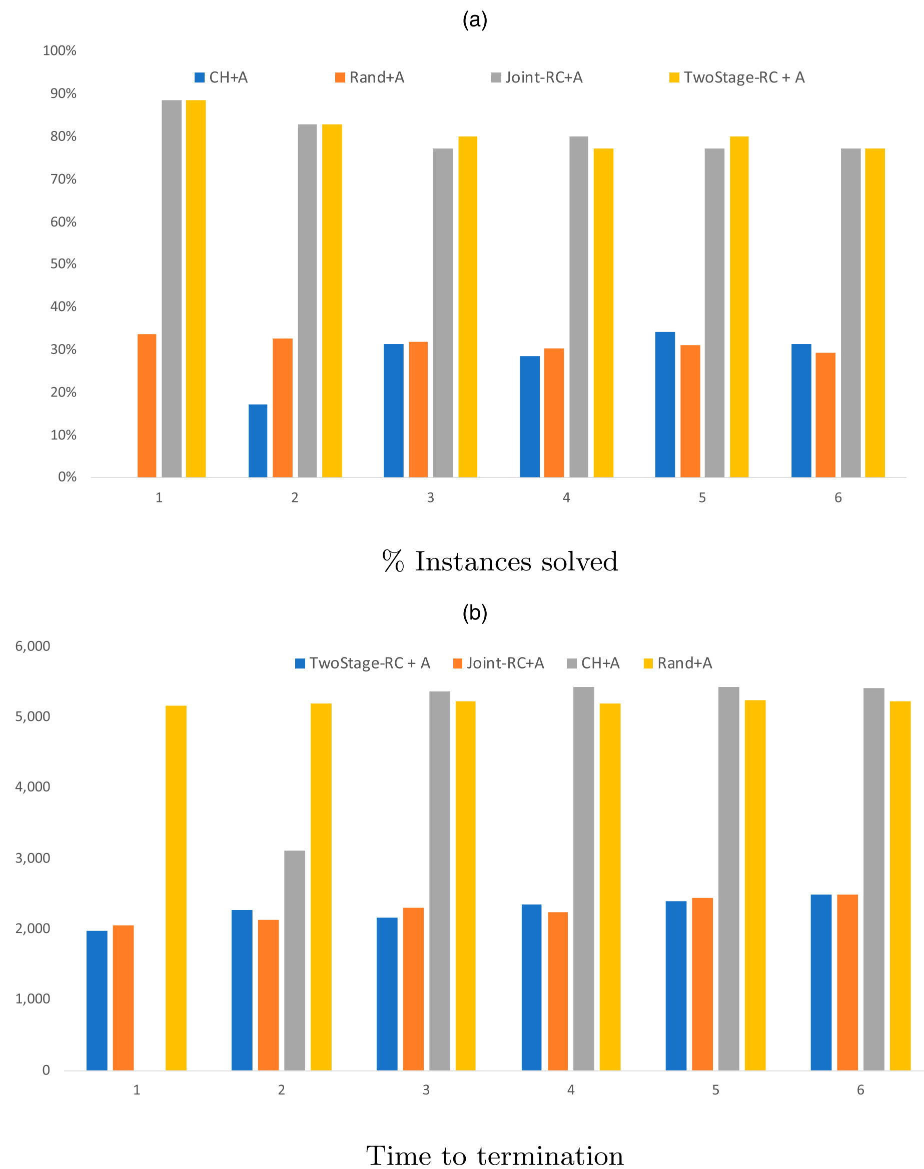


Table 5. Time to Solve Row Covering MIP, by Number of Scenarios

	Number of scenarios retained, $\lvert\bar{S}\rvert$					
	1	2	3	4	5	6
Joint-RC+A	11.89	9.71	8.54	7.80	7.49	7.14
TwoStage-RC+A	6.46	6.29	6.23	6.34	6.20	6.09

approximate the expected recourse cost in the optimal solution to the LPR in just the first iteration.

One computational issue that is often encountered when employing Benders is instability, namely, that the values of the first stage variables y fluctuate wildly from one iteration to the next. We next study whether performing a partial decomposition mitigates this issue. To do so, we consider two statistics related to the vector of first stage values associated with a feasible solution to P, which is a network design in our application. The first, Δ_{next}, is the average Hamming distance between successive designs found in the course of searching the branch-and-bound tree to solve P. The second, Δ_{final}, is the average of the Hamming distances from each design found by the algorithm to the last design found. We report these results in Table 7.

Comparing the values of Δ_{next} we see that the artificial scenario can contribute to a much more stable search than the pure scenario retention strategies. Comparing the values of Δ_{final} we see that the artificial scenario can direct the search toward better solutions. Both observations are particularly true when the artificial scenario is used in a hybrid, row covering-based strategy.

6.3. Sensitivity to the Number of Scenarios $|S|$

We finish our computational study by assessing the ability of the two row covering-based hybrid strategies to solve instances with a larger number of scenarios. Specifically, we consider the strategies Joint-RC(2)+A and TwoStage-RC(2)+A (i.e. $|\bar{S}| = 2$). We again compare the performance of these hybrid strategies with that of our two benchmarks. In Figure 7, we illustrate the percentage of instances each method was able to solve to within a 1% optimality gap. We see that, while the performance of all methods degrades as the number of scenarios increases, the row covering-based hybrid strategies are still able to solve a clear majority of the instances, even those with 256 scenarios (i.e., 68.57% of the instances).

We next report in Table 8 the amount of time each method needed to terminate, by number of scenarios. Table 9 reports the average provable optimality gap reported by each method and for each number of scenarios. We see that the row covering-based hybrid strategies are consistently able to produce provably higher quality solutions than the two benchmarks, and in less time.

Lastly, we recall that our method for implementing the row covering-based hybrid strategies involved first solving a MIP. We note that in our experiments all of these integer programs were solved to within a 1% optimality gap. In Table 10 we report the average time to solve these integer programs for each number of scenarios. We see that the time required to solve these MIPs increases in conjunction with the number of scenarios, but is still less than one minute with 256 scenarios. Therefore, the hybrid strategies remain efficiently applicable even when the size of the instances, with respect to the number of scenarios, significantly increases.

7. Conclusions and Future Work

We have established, both theoretically and computationally, the benefits associated with performing a partial Benders decomposition (PBD) strategy, particularly when applied to a stochastic network design problem, namely, retaining scenario subproblem information in the master problem solved during the execution of a Benders-based algorithm. In addition to proposing the PBD strategy, we also establish how to implement it and numerically validate the effectiveness of those implementation choices. The numerical experiments conducted on the stochastic fixed charge multicommodity network design problem show that the use of PBD enabled both the time to

Figure 4. (Color online) Phase One Convergence Rate by Strategy, $|S| = 96$

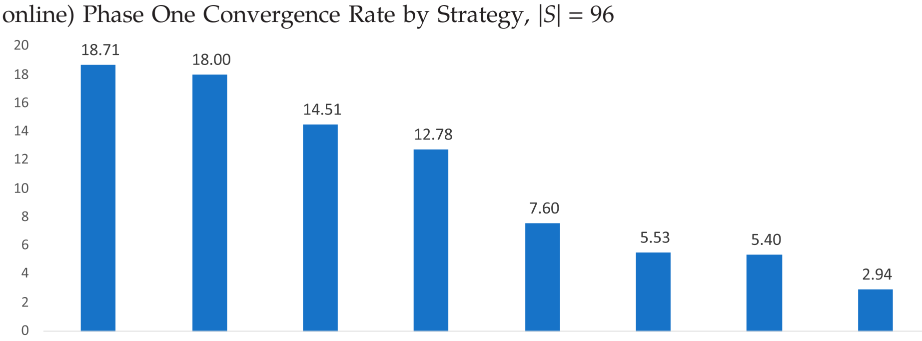
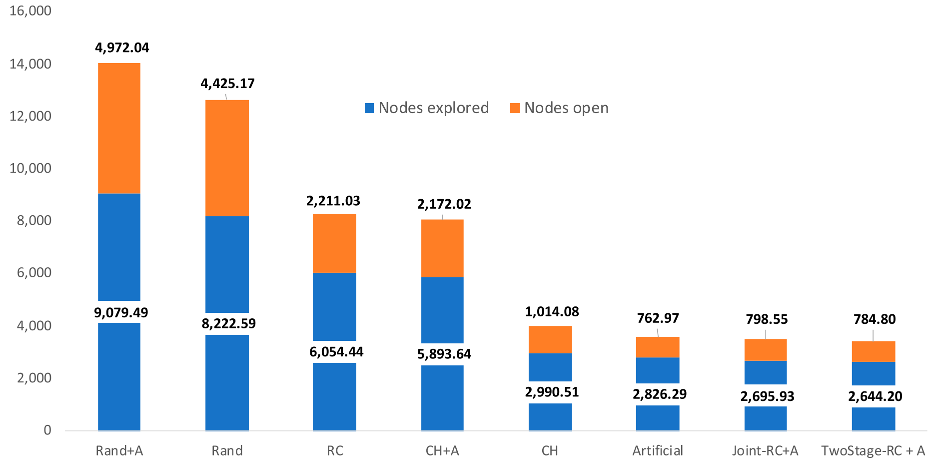


Figure 5. (Color online) Phase Two Convergence by Strategy, $|S| = 96$



termination of the algorithm and the optimality gap obtained at termination to be significantly reduced, while also increasing the overall number of instances solved to optimality in the maximum allotted time. It was further observed that the hybrid strategies, which combine both scenario-retention and -creation strategies, outperform all variants of PBD. The numerical analyses highlighted the fact that the hybrid strategies reduce by orders of magnitude the number of optimality and feasibility cuts generated by the algorithm, while also providing more stability in the search process. While the computational results clearly indicate the benefits of performing a PBD, there are other avenues for improving computational performance of a Benders-based algorithm through PBD-type ideas that we will explore in future research papers.

Firstly, the PBD strategy proposed retains whole scenarios in the master problem. Yet, one could instead pick and choose information (rows and variables) from each scenario to retain. Secondly, to date, we have based the PBD strategy solely on instance data; we could instead examine information related to solutions to the stochastic program to determine how to implement the PBD strategy. Thirdly, we make the decisions regarding how to perform the decomposition once, before beginning the execution of the Benders-based solution approach. We could instead make these choices dynamically, changing the master problem solved during the course of execution of the Benders-based algorithm. Fourthly, we have examined the effectiveness of a PBD strategy on one class of stochastic programs and one model within that class. The strategy can also be applied to other two-stage stochastic programs with continuous recourse, such as stochastic facility location models. Yet the strategy can also be applied to models that include more than two decision stages and/or where there are integrality requirements imposed on the recourse decisions. The extension of the PBD strategy to these models would allow it to be applied on a larger group of possible applications. Such problems would require both theoretical and computational development, which we intend to pursue in future research papers. Lastly, we intend to adapt other techniques designed for stochastic network design problems wherein there is demand uncertainty to the more general problem studied in this paper.

Figure 6. (Color online) Number of Optimality and Feasibility Cuts by Strategy, $|S| = 96$

Transportation Science, 2021, vol. 55, no. 2, pp. 414–435, © 2020 INFORMS

Table 6. Impact of Strategy on Expected Recourse in Phase One, $|S| = 96$

	CH	Rand+ A	Rand	RC	CH+A	Artificial	Joint-RC+A	TwoStage-RC+A
Expected recourse gap	91.96%	88.98%	87.50%	85.79%	80.79%	1.15%	0.14%	0.12%

Table 7. Stability (Measured by Hamming Distance) by Strategy, $|S| = 96$

Strategy	TwoStage-RC+A	Joint-RC+A	Artificial	RC	CH	Rand	Rand+A	CH+A
Δ_{next}	9.05	9.06	9.51	23.64	23.67	23.67	23.88	24.44
Δ_{final}	7.53	7.65	7.54	13.9	13.63	14.48	14.76	14.47

Figure 7. (Color online) Percent of Instances Solved to Within 1%, by $|S|, |\bar{S}| = 2$

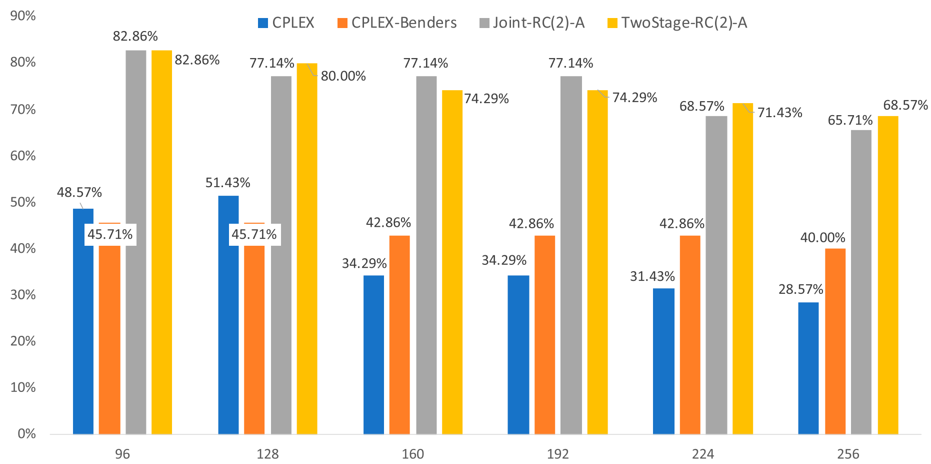


Table 8. Time to Termination, by Number of Scenarios

	Number of scenarios, $\lvert S\rvert$					
	96	128	160	192	224	256
CPLEX	4,012.86	4,507.17	4,920.89	5,216.06	5,365.89	5,333.34
CPLEX-Benders	4,306.29	4,507.66	4,550.69	4,619.43	4,708.17	4,759.00
Joint-RC(2)+A	2,133.51	2,468.31	2,473.34	2,565.91	3,112.23	3,412.80
TwoStage-RC(2)+A	2,276.20	2,425.63	2,620.23	2,659.14	2,812.06	2,955.20

Table 9. Optimality Gap at Termination, by Number of Scenarios

	Number of scenarios, $\lvert S\rvert$					
	96	128	160	192	224	256
CPLEX	9.95%	11.65%	14.91%	15.85%	16.24%	15.56%
CPLEX-Benders	7.76%	8.15%	8.85%	9.38%	9.65%	9.46%
Joint-RC(2)+A	1.18%	1.29%	1.53%	1.75%	2.27%	2.36%
TwoStage-RC(2)+A	1.13%	1.36%	1.56%	1.44%	1.70%	2.10%

Table 10. Time to Solve Row Covering MIP, by Number of Scenarios

	Number of scenarios, $\lvert S \rvert$					
	96	128	160	192	224	256
Joint-RC(2)+A	9.71	17.29	26.29	37.29	45.49	53.97
TwoStage-RC(2)+A	6.29	11.66	17.89	25.60	33.60	42.66

Acknowledgments

While working on this project, the first author was adjunct professor with the Department of Computer Science and Operations Research, Université de Montréal. While working on this project, W. Rei was the Canada Research Chair in Stochastic Optimization of Transport and Logistics Systems.

Appendix A. Scenario-Retention Strategy Proofs

The following is the proof of Proposition 1.

Proof of Proposition 1. Given $s \in \bar{S}$, $s' \in S \setminus \bar{S}$, and $w^j \geq 0$, we have that $w^{j\top} d^{s'} \leq w^{j\top} d^s$, considering that $d_l^s \geq d_l^{s'}$, $l = 1, \ldots, m_2$, by the assumption that s covers s' (Definition 1). It is easy to see that the constraints $w^{j^\top}(d^s - B^s\bar{y}) \leq 0$ are satisfied $\forall j \in \mathcal{J}$ such that $w^j \geq 0$. We thus obtain that $w^{j\top} d^{s'} \leq w^{j\top} d^s \leq w^{j^\top} B^s \bar{y} \leq w^{j^\top} B^{s'} \bar{y}$. □

The following is the proof of Proposition 2.

Proof of Proposition 2. One can easily see that the constraints $w^{j\top} d^s \leq w^{j\top} B^s \bar{y}$ are satisfied $\forall j \in \mathcal{J}$ and $s \in \bar{S}$. Thus, given $s \in \bar{S}$ and $s' \in S \setminus \bar{S}$, we have that

$$\begin{aligned} w^{j\top} d^{s'} &= w^{j\top} \left(\sum_{s \in \bar{S}} \alpha_s^{s'} d^s \right) \\ &= \sum_{s \in \bar{S}} \alpha_s^{s'} \left(w^{j\top} d^s \right) \\ &\leq \sum_{s \in \bar{S}} \alpha_s^{s'} \left(w^{j\top} B^s \bar{y} \right) \\ &= \sum_{s \in \bar{S}} w^{j\top} \left(\alpha_s^{s'} B^s \right) \bar{y} \\ &= w^{j\top} B^{s'} \bar{y}. \end{aligned}$$

We can thus conclude that $w^{j\top} d^{s'} \leq w^{j\top} B^{s'} \bar{y}$, $\forall j \in \mathcal{J}$. □

Appendix B. Scenario-Creation Strategy Proofs

The following is the proof of Lemma 1.

Proof of Lemma 1. Let $(y; x^s : s \in S)$ be a solution to P. To prove the result, we show that a solution $(y; x^s : s \in S; x^{s'})$ to $P(s')$ can be obtained with the same cost by setting $x^{s'} = \sum_{s \in S} \alpha_s^{s'} x^s$. This is done by first establishing that these variable values satisfy Constraints (20) and (21). We have that $B^{s'} y + Dx^{s'} = \sum_{s \in S} \alpha_s^{s'} B^s y + D \sum_{s \in S} \alpha_s^{s'} x^s = \sum_{s \in S} \alpha_s^{s'} (B^s y + \ Dx^s) = \sum_{s \in S} \alpha_s^{s'} d^s = d^{s'}$, and thus, Constraints (20) are satisfied. Constraint (21) is automatically satisfied by the previous assumption concerning the definition of $x^{s'}$. Since (17) does not contain coefficients for the variables $x^{s'}$, the cost of the solution $(y; x^s : s \in S; x^{s'})$ is the same in $P(s')$ as the cost of $(y; x^s : s \in S)$ in P. Given that the previous reasoning can be applied to all feasible solutions to P, one concludes that $v^\star_{s'} = v^\star$. □

The following is the proof of Lemma 2.

Proof of Lemma 2. $BP(\bar{y}, I, J)$ is defined as the following model:

$$\begin{aligned} \min \quad & \sum_{s \in S} p_s z^s \\ \text{subject to} \quad & q^{i\top} d^s - q^{i\top} B^s \bar{y} \leq z^s, \quad \forall i \in I, s \in S, \\ & z^s \geq 0, \qquad\qquad\qquad \forall s \in S. \end{aligned}$$

Given that $p_s \geq 0$, $s \in S$, model $BP(\bar{y}, I, J)$ decomposes via the scenarios and can be equivalently formulated as:

$$\begin{aligned} &= \sum_{s \in S} p_s \{ \min\ z^s \mid q^{i\top} d^s - q^{i\top} B^s \bar{y} \leq z^s, i \in I, z^s \geq 0 \} \\ &= \sum_{s \in S} p_s \left\{ \arg\max_{i \in I} \left\{ q^{i\top} d^s - q^{i\top} B^s \bar{y} \bigvee 0 \right\} \right\} \\ &= \sum_{s \in S} p_s \left(\max\{0, q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y}\} \right) \quad \text{with} \\ &\qquad i_s^\star \in \arg\max_{i \in I} \{ q^{i\top} d^s - q^{i\top} B^s \bar{y}, \ \forall s \in S \} \\ &= \sum_{s \in S} p_s \bar{z}^s. \quad \square \end{aligned}$$

The following is the proof of Proposition 3.

Proof of Proposition 3. Given the definition of $\bar{x}_A^{s'}$, $BP(\bar{y}, \bar{S}, I, J)$ can be formulated as follows:

$$\min \quad \sum_{s \in S} p_s z^s \tag{59}$$

$$\text{subject to} \quad c^\top \bar{x}_A^{s'} = \sum_{s \in S} \alpha_s^{s'} z^s, \tag{60}$$

$$q^{i\top} d^s - q^{i\top} B^s \bar{y} \leq z^s, \ \forall i \in I, s \in S, \tag{61}$$

$$z^s \geq 0, \qquad \forall s \in S. \tag{62}$$

Constraints (61) can, however, be reformulated as

$$\begin{aligned} &\bigwedge_{i \in I} q^{i\top} d^s - q^{i\top} B^s \bar{y} \leq z^s, \forall s \in S, \\ &q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y} \leq z^s, \forall s \in S. \end{aligned}$$

By recalling that $i_s^* \in \arg\max_{i \in I} \{ q^{i\top} d^s - q^{i\top} B^s \bar{y}, \forall s \in S \}$, and given the presence of the nonnegativity requirements (62), the constraint set (61)–(62) can be expressed as $\max\{0, q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y}\} \leq z^s$, $\forall s \in S$. In turn, given Lemma 2, this is equivalent to $\bar{z}^s \leq z^s$, $\forall s \in S$. Therefore, $\bar{z}^s \leq z^s$, $\forall s \in S$, are enforced for all feasible solutions to $BP(\bar{y}, \bar{S}, I, J)$. Additionally, considering that the problem also includes the constraint (60), we necessarily have that the optimal values $\bar{z}_A^s$ satisfy $\bar{z}_A^s \geq \bar{z}^s$, $\forall s \in S$. □

The following is the proof of Proposition 4.

Proof of Proposition 4. By invoking Lemma 2, one obtains that

$$\begin{gathered} c^\top \bar{x}_A^{s'} \neq \sum_{s \in S} \alpha_s^{s'} \bar{z}^s \\ \Updownarrow \\ c^\top \bar{x}_A^{s'} \neq \sum_{s \in S} \alpha_s^{s'} \left(\max\left\{ 0, q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y} \right\} \right). \end{gathered}$$

Recalling that $i_s^\star \in \arg\max_{i \in I} \{ q^{i\top} d^s - q^{i\top} B^s \bar{y}, \forall s \in S \}$, one directly derives that the variables z^s, $s \in S$, cannot all be set to their respective lower bounds, which are specified by the optimality cuts and nonnegativity requirements included in $BP(\bar{y}, \bar{S}, I, J)$. As a direct consequence, $c^\top \bar{x}_A^{s'} > \sum_{s\in S} \alpha_s^{s'}(\max\{0, q_s^{i_s^\star \top} d^s - q_s^{i_s^\star \top} B^s \bar{y}\})$, and in the optimal solution to $BP(\bar{y}, \bar{S}, I, J)$, at least one variable z^s will take on a value that is strictly higher than the lower bound provided by the optimality cuts and nonnegativity constraints. Furthermore, it is important to note that $c^\top \bar{x}_A^{s'} > 0$ also applies in the present case.

The next step is to determine the variables that will not be set to their respective lower bounds and the actual values that they will take. To do so, we show that $BP(\bar{y}, \bar{S}, I, J)$ can be expressed as a continuous knapsack problem. Given the definition of $\bar{x}_A^{s'}$ and by setting $\beta = c^\top \bar{x}_A^{s'}$ and $\sigma^s = \max\{0, q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y}\}$, $\forall s \in S$ (which we define to alleviate the notation that is used to develop the next steps of the proof), then $BP(\bar{y}, \bar{S}, I, J)$ can be defined as:

$$\min \sum_{s\in S} p^s z^s$$
$$\text{s.t.} \sum_{s\in S} \alpha_s^{s'} z^s = \beta,$$
$$\sigma^s \le z^s, \forall s \in S.$$

We further define the following variables:

$$v^s = \alpha_s^{s'} \left(\frac{z^s - \sigma^s}{\beta}\right), \forall s \in S,$$
$$\Updownarrow$$
$$z^s = \frac{v^s}{\alpha_s^{s'}} \beta + \sigma^s, \forall s \in S.$$

Therefore, $BP(\bar{y}, \bar{S}, I, J)$ can be redefined as follows:

$$\min \quad \sum_{s\in S} p^s \left(\frac{\beta}{\alpha_s^{s'}} v^s + \sigma^s\right)$$
$$\text{s.t.} \quad \sum_{s\in S} \alpha_s^{s'} \left(\frac{\beta}{\alpha_s^{s'}} v^s + \sigma^s\right) = \beta,$$
$$0 \le v^s \le 1, \forall s \in S.$$
$$\Updownarrow$$
$$\min \quad \sum_{s\in S} \frac{p^s}{\alpha_s^{s'}} v^s \tag{63}$$
$$\text{s.t.} \sum_{s\in S} v^s = \frac{\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s}{\beta}, \tag{64}$$
$$0 \le v^s \le 1, \forall s \in S. \tag{65}$$

Model (63)–(65) represents a continuous knapsack problem. Furthermore, since $0 < (\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s)/\beta \le 1$, the optimal solution to (63)–(65) is expressed as follows:

$$v^{\tilde{s}} = \frac{\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s}{\beta}, \quad \tilde{s} \in \operatorname*{argmin}_{s\in S} \left\{\frac{p^s}{\alpha_s^{s'}}\right\}, \tag{66}$$
$$v^s = 0, \quad \forall s \in S \setminus \{\tilde{s}\}. \tag{67}$$

By replacing (66)–(67) in $z^s = (v^s/\alpha_s^{s'})\beta + \sigma^s$, $\forall s \in S$, the optimal solution to $BP(\bar{y}, \bar{S}, I, J)$ is expressed as:

$$\bar{z}_A^{\tilde{s}} = \frac{\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s}{\alpha_{\tilde{s}}^{s'}} + \sigma^{\tilde{s}}, \quad \tilde{s} \in \arg\min_{s\in S} \left\{\frac{p^s}{\alpha_s^{s'}}\right\}, \tag{68}$$
$$\bar{z}_A^s = \sigma^s, \quad \forall s \in S \setminus \{\tilde{s}\}. \tag{69}$$

Considering that $\bar{z}^s = \max\{0, q_s^{i_s^\star \top} d^s - q_s^{i_s^\star \top} B^s \bar{y}\} \Rightarrow \bar{z}^s = \sigma^s$, $\forall s \in S$, and given the optimal solution detailed by (68)–(69), one obtains $\Delta_A(\bar{y}, I, J) = \sum_{s\in S} p_s(\bar{z}_A^s - \bar{z}^s) = \frac{p^{\tilde{s}}}{\alpha_{\tilde{s}}^{s'}}(c^\top \bar{x}_A^{s'} - \sum_{s\in S} \alpha_s^{s'} \bar{z}^s)$. □

References

Adulyasak Y, Cordeau JF, Jans R (2015) Benders decomposition for production routing under demand uncertainty. *Oper. Res.* 63(4):851–867.

Angulo G, Ahmed S, Dey SS (2016) Improving the integer L-shaped method. *INFORMS J. Comput.* 28(3):483–499.

Batun S, Denton BT, Huschka TR, Schaefer AJ (2011) Operating room pooling and parallel surgery processing under uncertainty. *INFORMS J. Comput.* 23(2):220–237.

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* 4:238–252.

Bihlmaier R, Koberstein A, Obst R (2009) Modeling and optimizing of strategic and tactical production planning in the automotive industry under uncertainty. *OR Spectrum* 31:311–336.

Birge JR (1982) The value of the stochastic solution in stochastic linear programs with fixed recourse. *Math. Programming* 24:314–325.

Birge JR (1985) Aggregation bounds in stochastic linear programming. *Math. Programming* 31:25–41.

Birge JR, Louveaux FV (1988) A multicut algorithm for two-stage stochastic linear programs. *Eur. J. Oper. Res.* 34(3):384–392.

Birge JR, Louveaux FV (2011) *Introduction to Stochastic Programming* (Springer Science and Business Media, New York).

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

Costa AM (2005) A survey on Benders decomposition applied to fixed-charge network design problems. *Comput. Oper. Res.* 32(6):1429–1450.

Côté G, Laughton MA (1984) Large-scale mixed integer programs: Benders-type heuristic. *Eur. J. Oper. Res.* 16(3):327–333.

Crainic TG, Frangioni A, Gendron B (2001) Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Appl. Math.* 112(1):73–99.

Crainic TG, Hewitt M, Rei W (2014) Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Comput. Oper. Res.* 43:90–99.

Crainic TG, Hewitt M, Maggioni F, Rei W (2016) Partial Benders decomposition strategies for two-stage stochastic integer programs. Technical Report CIRRELT-2016-37, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT), Montreal.

Crainic TG, Fu X, Gendreau M, Rei W, Wallace SW (2011) Progressive hedging-based metaheuristics for stochastic network design. *Networks* 58(2):114–124.

de Camargo RS, de Miranda G Jr, Luna HPL (2009) Benders decomposition for hub location problems with economies of scale. *Transportation Sci.* 43(1):86–97.

Espinoza D, Moreno E (2014) A primal-dual aggregation algorithm for minimizing conditional value-at-risk in linear programs. *Comput. Optim. Appl.* 59(3):617–638.

Fischetti M, Ljubić I, Sinnl M (2016) Benders decomposition without separability: A computational study for capacitated facility location problems. *Eur. J. Oper. Res.* 253(3):557–569.

Fischetti M, Ljubić I, Sinnl M (2017) Redesigning Benders decomposition for large-scale facility location. *Management Sci.* 63(7):2146–2162.

Fischetti M, Salvagnin D, Zanette A (2010) A note on the selection of Benders' cuts. *Math. Programming* 124:175–182.

Geoffrion AM (1970a) Elements of large-scale mathematical programming part 1: Concepts. *Management Sci.* 16(11):652–675.

Geoffrion AM (1970b) Elements of large-scale mathematical programming part 2: Synthesis of algorithms and bibliography. *Management Sci.* 16(11):676–691.

Geoffrion AM (1972) Generalized Benders decomposition. *J. Optim. Theory Appl.* 10(2):237–260.

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

Holmberg K (1994) On using approximations of the Benders master problem. *Eur. J. Oper. Res.* 77(1):111–125.

Hooker JN, Ottosson G (2003) Logic-based Benders decomposition. *Math. Programming* 96:33–60.

Høyland K, Kaut M, Wallace SW (2003) A heuristic for moment-matching scenario generation. *Comput. Optim. Appl.* 24(2-3): 169–185.

Klibi W, Martel A (2012) Scenario-based supply chain network risk modeling. *Eur. J. Oper. Res.* 223(3):644–658.

Klibi W, Martel A, Guitouni A (2010) The design of robust value-creating supply chain networks: A critical review. *Eur. J. Oper. Res.* 203(2):283–293.

Litvinchev I, Tsurkov V (2003) *Aggregation in Large-Scale Optimization* (Kluwer Academic Publishers, Dordrecht, Netherlands).

Liu Y, Ferris MC, Zhao F (2014) Computational study of security constrained economic dispatch with multi-stage rescheduling. *IEEE Trans. Power Systems* 30(2):920–929.

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

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

MirHassani SA, Lucas C, Mitra G, Messina E, Poojari CA (2000) Computational solution of capacity planning models under uncertainty. *Parallel Comput.* 26(5):511–538.

Nemhauser G, Wolsey L (1988) *Integer and Combinatorial Optimization* (Wiley, New York).

Papadakos N (2008) Practical enhancements to the Magnanti-Wong method. *Oper. Res. Lett.* 36(4):444–449.

Pay BS, Song Y (2020) Partition-based decomposition algorithms for two-stage stochastic integer programs with continuous recourse. *Ann. Oper. Res.* 284:583–604.

Pishvaee M, Razmi J, Torabi S (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 Res. Part E: Logist. Transportation Rev.* 67:14–38.

Poojari CA, Beasley JE (2009) Improving Benders decomposition using a genetic algorithm. *Eur. J. Oper. Res.* 199(1):89–97.

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The Benders decomposition algorithm: A literature review. *Eur. J. Oper. Res.* 259(3):801–817.

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

Rahmaniani R, Ahmed S, Crainic TG, Gendreau M, Rei W (2020) The Benders dual decomposition method. *Oper. Res.* 68(3): 878–895.

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

Rockafellar RT, Uryasev S (2013) The fundamental risk quadrangle in risk management, optimization and statistical estimation. *Surveys Oper. Res. Management Sci.* 18(1–2):33–53.

Rosa CH, Takriti S (1999) Improving aggregation bounds for two-stage stochastic programs. *Oper. Res. Lett.* 24(3):127–137.

Roy TJV (1983) Cross decomposition for mixed integer programming. *Math. Programming* 25(1):46–63.

Sandikçi B, Kong N, Schaefer AJ (2013) A hierarchy of bounds for stochastic mixed-integer programs. *Math. Programming* 138: 253–272.

Shen S, You M, Ma Y (2017) Single-commodity stochastic network design under demand and topological uncertainties with insufficient data. *Naval Res. Logist.* 64(2):154–173.

Sherali HD, Lunday BJ (2013) On generating maximal nondominated Benders cuts. *Ann. Oper. Res.* 210:57–72.

Song Y, Luedtke J (2015) An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. *SIAM J. Optim.* 25(3):1344–1367.

van Ackooij W, de Oliveira W, Song Y (2018) Adaptive partition-based level decomposition methods for solving two-stage stochastic programs with fixed recourse. *INFORMS J. Comput.* 30(1):57–70.

Van Slyke R, Wets RJB (1969) L-shaped programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.* 17(4):638–663.

Wheatley D, Gzara F, Jewkes E (2015) Logic-based Benders decomposition for an inventory-location problem with service constraints. *Omega* 55:10–23.

Zakeri G, Philpott AD, Ryan DM (2000) Inexact cuts in Benders decomposition. *SIAM J. Optim.* 10(3):643–657.