Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Partial Benders Decomposition Strategies for Two-Stage Stochastic Integer Programs

Teodor Gabriel Crainic
Mike Hewitt
Francesca Maggioni
Walter Rei

July 2016

CIRRELT-2016-37

Partial Benders Decomposition Strategies for Two-Stage Stochastic Integer Programs†

Teodor Gabriel Crainic[1,*], Mike Hewitt[2], Francesca Maggioni[3], Walter Rei[1]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

[2] Department of Information Systems & Operations Management, Loyola University Chicago, 820 N. Michigan Ave, Chicago, IL, USA 60611

[3] Department of Management, Economics and Quantitative Methods, University of Bergamo, Via dei Caniana 2, 24127, Bergamo, Italy

Abstract. Benders decomposition is a broadly used used exact solution method for stochastic programming, enabling such programs to be decomposed according to the realizations of the random events that set the values of their stochastic parameters. This strategy also comes with important drawbacks, however, such as a weak master problem following the relaxation step that confines the dual cuts to the scenario sub-problems. We propose the first comprehensive Partial Benders Decomposition methodology for two-stage integer stochastic program, based on the idea of including explicit information from the scenario sub-problems in the master. We propose two scenario-retention strategies that include a subset of the second stage scenario variables in the master, aiming to significantly reduce the number of feasibility cuts generated. We also propose a scenario-creation strategy to improve the lower-bound provided by the master problem, as well as three hybrids obtained by combining these pure strategies. We report the results of an extensive experimental campaign on two-stage stochastic multicommodity network design problems. The analysis clearly shows the proposed methodology yields significant benefits in computation efficiency, solution quality, and stability of the solution process.

Keywords: Benders decomposition, partial Benders decomposition, scenario retention strategies, scenario creation strategies, two-stage stochastic programs, stochastic network design.

Acknowledgements. Partial funding for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Discovery Grant program. We also gratefully acknowledge the support of Fonds de recherche du Québec through their infrastructure grants and of Calcul Québec and Compute Canada for access to their high-performance computing infrastructure. The third author expresses her acknowledgments to the Research Council of the University of Bergamo for the visiting professor research grant support 2015. While working on this project, the first author was Adjunct Professor with the Department of Computer Science and Operations Research of the Université de Montréal.

† Revised version of the CIRRELT-2014-13.

---

* Corresponding author: teodorgabriel.crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

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. 2016), while also being successfully adapted for, e.g., non-linear optimization (Geoffrion 1972) and stochastic programming (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, as illustrated by, e.g., de Camargo et al. (2009), Contreras et al. (2011) for hub location, Adulyasak et al. (2015) for production-routing, Batun et al. (2011) for operating room scheduling, Wheatley et al. (2015) for inventory-routing, Pishvaee et al. (2014) for supply chain design in the health sector, and MirHassani et al. (2000a) for capacity planning.

Decisions in a stochastic program 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-caled *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, e.g., 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* (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, for a large part, to the number of scenarios used to represent uncertainty, Benders decomposition greatly simplifies their resolution. This strategy also comes with important drawbacks, however, that need to be addressed to produce an overall efficient solution procedure. Our goal is to introduce mechanisms addressing this challenge, focusing on two stage stochastic integer programs, which define a family of optimization problems extremely challenging to address.

Benders decomposition (Geoffrion 1970a,b) for stochastic programs relies on the application of three main steps: *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 sub-problems* 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 sub-problems. 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 contribution in this paper is to propose the first comprehensive *Partial Benders Decomposition* (*PBD*) methodology for two-stage integer stochastic programs, based on the idea of including explicit information from the scenario sub-problems in the master. The proposed PBD thus strengthens the master formulation which, in turn, helps overcome the challenges evoked above. A few previous studies had applied rather straightforwardly this idea to improve the performance of the L-Shaped algorithm for particular stochastic optimization problems (MirHassani et al. 2000b, Bihlmaier et al. 2009, Batun et al. 2011, Liu et al. 2014). A comprehensive methodology for general settings had yet to be developed, however.

The methodology we propose is built on two types of PBD strategies, based on scenario retention and creation, respectively. *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. We propose two strategies on how to select the sub-problems to keep in the master that not only eliminate the feasibility cuts for the retained scenarios, but also significantly reduce the number of feasibility cuts for the non retained ones. *Scenario-creation* strategies, on the other hand, are designed to improve the lower-bound provided by the master problem. We thus show how valid artificial scenario sub-problems can be created and included in the master to directly impact the values of the projection variables that approximate the recourse cost. The second contribution of this paper is thus to present theoretical results supporting the principles on which the scenario retention and creation strategies are based.

The methodology is completed by two hybrid strategies created by combining the two pure scenario-retention and the pure scenario-creation strategies. Together with the general method we provide to implement the proposed PBD strategies, the hybrids make up the third contribution of the paper.

As a fourth contribution, we report the results of an extensive experimental campaign on two-stage stochastic multicommodity network design problems, an important combinatorial optimization problem class known for its inherent complexity (the deterministic version is already NP-Hard and computationally challenging) and wide applicability. The analysis clearly show the significant benefit in computation efficiency and solution quality of the proposed methodology, qualifies the relative merits of the pure and hybrid strategies, and provides insights in the behavior of the proposed partial Benders decomposition. In particular, compared to a standard implementation of the L-Shaped algorithm and a straightforward version of the scenario-retention idea, all proposed PBD strategies significantly reduced the optimality gap, the number of cuts generated, and the number of iterations required to converge, while increasing the stability of the solution process and identifying high-quality feasible solutions faster. They also illustrate that best results were in fact obtained by the combined, hybrid strategies.

The rest of the paper is structured as follows. Section 2 briefly recalls Benders decomposition applied to the problem class considered, 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 partial Benders decomposition methodological framework. The experimental design is described in Section 5, while Section 6 presents the analysis of the computational results. We conclude in Section 7.

2 Benders Decomposition for Two-Stage Stochastic Integer Programs

Let $\Re^n$ stand for the space of all n-dimensional real vectors, $\Re^{n\times m}$ the space of real $n\times m$ matrices. Let $y \in \mathscr{Y} \subseteq \Re^{n_1}$ 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.

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

subject to

$$Ay = b, \tag{2}$$

$$B^s y + Dx^s = d^s, \quad \forall s \in S, \tag{3}$$

$$y \in \mathscr{Y},\ x^s \geq 0,\ \forall s \in S. \tag{4}$$

We address the case of continuous recourse decisions in this paper, as it facilitates the application of standard duality theory. We discuss he generalization of the method to the case where integrality requirements are present in the two 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 sub-problem $SP(y)_s$, for each scenario s

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

and we may reformulate P (Projection Step 1 of Benders decomposition) as

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

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

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

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, $i \in \mathscr{I}$, with cardinality $|\mathscr{I}|$, and extreme rays w^j, $j \in \mathscr{J}$, with cardinality $|\mathscr{J}|$. A valid reformulation of P (Dualization Step), which we refer to as BP, the *master problem*, then is

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

subject to

$$Ay = b, \tag{6}$$

$$q^{i\top}(d^s - B^s y) \leq z^s, \quad \forall i \in \mathscr{I},\ s \in S, \tag{7}$$

$$w^{j\top}(d^s - B^s y) \leq 0, \quad \forall j \in \mathscr{J},\ s \in S, \tag{8}$$

$$y \in \mathscr{Y}, \tag{9}$$

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

Solving 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 $I \subseteq \mathscr{I}$ and $J \subseteq \mathscr{J}$ 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 $\bar{y}$. Dual sub-problems $DSP(\bar{y})_s$ 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 repeats. Otherwise, problem P has been solved.

The presentation above sums up what is often referred to as the *Multi-cut* version of the L-shaped method (Birge and Louveaux 1988), and it is illustrated in Algorithm 1 (see Slyke

Algorithm 1 Benders decomposition

Set $I = \emptyset$ and $J = \emptyset$
Create $BP(I,J)$
while P not solved **do**
  Solve $BP(I,J)$ to get vector $\bar{y}$
  for $s \in S$ **do**
    Solve dual sub-problem $DSP(\bar{y})_s$
    if constraints from sets (7) or (8) are violated **then**
      Update I, in the case of a violated optimality cut (7)
      Update J, in the case of a violated feasibility cut (8)
    end if
  end for
  if no violated constraints found **then**
    Stop {Solved P}
  end if
end while

and Wets (1969) for the *Single-cut* version wherein the scenario-based cuts generated at each iteration are aggregated in 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 algorithm.

3 Accelerating the Benders Decomposition - A Brief Literature Review

We briefly review the different strategies proposed in the literature to improve, accelerate, Benders decomposition both in the general mixed integer programming context and those specifically designed to enhance the L-Shaped algorithm (for a detailed review see Rahmaniani et al. 2016). 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. 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 sub-problems, by using suboptimal extreme points of the dual region of the scenario sub-problems to generate valid cuts (Zakeri et al. 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, Côté and Laughton (1984) showed that Lagrangean 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. Rei et al. (2009) and Poojari and Beasley (2009) followed this idea and proposed Benders algorithms generating multiple cuts at each iteration by solving the master problem using local branching and a genetic procedure, respectively, with significant numerical improvements for deterministic (Rei et al. 2009, Poojari and Beasley 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 (Lagrangean) decompositions to solve mixed integer programs. The author showed that a sequence of solutions to the Benders master problem can be obtained by alternately solving the Benders and Lagrangean sub-problems. Then, solving the Benders master problem on occasion only maintains convergence to optimality and accelerates the search. Holmberg (1994) compared the performance of different Lagrangean relaxation approximations applied to the Benders master problem.

The third category encompasses strategies that aim to strengthen the quality of the cuts generated by the algorithm. Magnanti and Wong (1981) first proposed to generate non-dominated optimality (*Pareto-optimal*) cuts whenever, for a given master-problem feasible solution, multiple optimal solutions (i.e., extreme points) are associated to the dual sub-problem (scenario sub-problems in the stochastic case). An additional dual sub-problem instantiated using a core point of the master was solved to identify a non-dominated cut. The addition of Pareto-optimal cuts greatly improved the value of the lower bound obtained by the algorithm. Papadakos (2008) improved the approach by showing how alternative points can be used as proxies for core ones in the single sub-problem solved to produce Pareto-optimal cuts. Fischetti et al. (2010) proposed a different strategy to generate non-dominated cuts, by reformulating the sub-problem as a feasibility problem where cuts, both optimality and feasibility, are obtained by searching for minimal infeasible subsystems. Sherali and Lunday (2013) further enhanced these strategies to generate maximal non-dominated cuts.

The fourth category groups the methods specifically tailored for the L-Shaped algorithm and two-stage stochastic integer programs, in particular those that included information from the scenario sub-problems in the master problem as a means to strengthen its formulation. MirHassani et al. (2000b) (see also Bihlmaier et al. 2009) proposed to retain a single scenario sub-problem and numerically showed that choosing the scenario for which the total demand was highest produced the best results. Liu et al. (2014) developed a a feasibility checker model to assess the challenge in obtaining a feasible solution to the master with respect to each scenario sub problem. Sub-problems thus identified as ”difficult” were then included in the master. Finally, Batun et al. (2011) 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 considered problem.

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 mixed integer programs (MIPs), reflected in the efficiency of off-the-shelf solvers (see Fischetti et al. 2016, for a different perspective on using this increased efficiency in enhancing Benders), alleviates 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 Partial Benders Decomposition 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 boosting the performance of L-Shaped algorithm. The next section details the proposed PBD methodology, the associated strategies and implementation method.

4 Partial Benders Decomposition

We present in this section the proposed Partial Benders Decomposition 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 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 strategies proposed.

4.1 Defining the master problem

As previously stated, PBD is based on the idea of including information related to the scenario sub-problems 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 $s' \in \bar{S}$, such that $s' \notin S$, which are artificially generated. We show in Section 4.3 how to generate valid artificial scenarios for the original problem P (1)-(4). For now, it is simply assumed they are available. Using sets $\bar{S}$, $I \subseteq \mathscr{I}$ and $J \subseteq \mathscr{J}$, the relaxed master problem is reformulated as $BP(\bar{S}, I, J)$

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

subject to

$$Ay = b, \tag{11}$$

$$B^s y + Dx^s = d^s, \quad \forall s \in \bar{S}, \tag{12}$$

$$c^\top x^{s'} = \sum_{s \in S \setminus \bar{S}} \alpha_s^{s'} z^s, \quad \forall s' \in \bar{S}, s' \notin S, \tag{13}$$

$$q^{i\top} d^s \leq q^{i\top} B^s y + z^s, \quad \forall i \in I, s \in S \setminus \bar{S}, \tag{14}$$

$$w^{j\top} d^s \leq w^{j\top} B^s y, \quad \forall j \in J, s \in S \setminus \bar{S}, \tag{15}$$

$$y \in \mathscr{Y}, \; x^s \geq 0, \; \forall s \in \bar{S}. \tag{16}$$

There are several differences between the traditional relaxed master problem $BP(I,J)$ and $BP(\bar{S},I,J)$. First, the sub problems associated to 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 turn, one can expect that the L-Shaped algorithm based on this decomposition will reach better solutions faster when compared to the traditional decomposition (where $\bar{S} = \emptyset$). In addition, greater stability with respect to the solutions (and optimality cuts) obtained 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 sub-problems associated to $s \in \bar{S}$, which eliminates the need to generate feasibility cuts for the scenario sub-problems $s \in S \cap \bar{S}$. Furthermore, by retaining these linking constraints in the formulation of the master, valid inequalities for the polyhedron $P_s = \{(x^s, y) : Ay = b, B^s y + Dx^s = d^s, y \in \mathscr{Y}, x^s \geq 0\}$ 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 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. 2000b, Bihlmaier et al. 2009, Batun et al. 2011, Liu et al. 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 sub-problems in the master formulation. Retaining a scenario from S removes the need to generate feasibility cuts for its sub-problem. We go further and propose two strategies that can also obviate the need for generating feasibility cuts for scenarios that are not retained, i.e., 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. It does so by focusing on the vectors that contain the stochastic parameters defining each scenario and choosing those scenarios that ensure the convex hull of their vectors contains the vectors of the scenarios not retained. We next define these strategies precisely, proving in both cases how to ensure feasibility in non-retained scenario sub-problems (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 *Given two scenarios $s \in \overline{S}$ and $s' \in S \setminus \overline{S}$, if a row index $l \in \{1, \ldots, m_2\}$ is such that $d_l^s \geq d_l^{s'}$ and $B_l^{s'\top} y \geq B_l^{s\top} y, \ \forall y \in \mathscr{Y}$, then we state that s covers s' with respect to l. We note that if $y \geq 0$, $\forall y \in \mathscr{Y}$, then the condition $B_l^{s'\top} y \geq B_l^{s\top} y$, $\forall y \in \mathscr{Y}$, is equivalent to requiring* *that* $B^{s'}_{li} \geq B^{s}_{li}$, $\forall i = 1,\ldots,n_1$. *If s covers* s' *with respect to all indices* $l = 1,\ldots,m_2$, *then we declare that scenario s covers* s'.

The motivating result behind the proposed covering strategy is then contained in Proposition 1.

Proposition 1 *Consider* $s \in \bar{S}$ *and* $s' \in S \setminus \bar{S}$ *such that s covers* 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* $w^{j\top} d^{s'} \leq w^{j\top} B^{s'} \bar{y}$, $\forall j \in \mathscr{J}$ *such that* $w^j \geq 0$.

Proof 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 $d^s_l \geq d^{s'}_l$, $l = 1,\ldots,m_2$, 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 $\forall j \in \mathscr{J}$ such that $w^j \geq 0$. We thus obtain that $w^{j\top} d^{s'}_l \leq w^{j\top} d^s_l \leq w^{j^\top} B^s \bar{y} \leq w^{j^\top} B^{s'} \bar{y}$.**Q.E.D.**

Proposition 1 illustrates how to ensure feasibility in specific non-retained scenario subproblems. If a scenario $s' \in S \setminus \overline{S}$ 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 (i.e., $w^{j\top} d^{s'} \leq w^{j\top} B^{s'} \bar{y}$, $\forall j \in \mathscr{J}$ such that $w^j \geq 0$). 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 analogous 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. Following this approach, the scenarios added to $\bar{S}$ are the ones that include in their associated convex hull the scenarios in $S \setminus \bar{S}$. 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* $\exists\, \alpha^{s'}_s \geq 0, s \in \overline{S}$: $\sum_{s \in \bar{S}} \alpha^{s'}_s = 1$, $\sum_{s \in \bar{S}} \alpha^{s'}_s d^s = d^{s'}$, *and* $\sum_{s \in \bar{S}} \alpha^{s'}_s B^s = B^{s'}$. *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* $w^{j\top} d^{s'} \leq w^{j\top} B^{s'} \bar{y}, \forall j \in \mathscr{J}$.

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

$\forall j \in \mathscr{J}$ and $s \in \bar{S}$. Thus, given $s \in \bar{S}$ and $s' \in S \setminus \bar{S}$, we have that

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

We can thus conclude that $w^{j\top} d^{s'} \leq w^{j\top} B^{s'} \bar{y}$, $\forall j \in \mathscr{J}$.**Q.E.D.**

Thus, if there exists a set of weights $\alpha_s^{s'}$ with which one can express the vector $d^{s'}$ and matrix $B^{s'}$ as a convex combination of the vectors d^s and matrices B^s, for $s \in \bar{S}$, then a feasible solution $\bar{y}$ to $BP(\bar{S}, I, J)$ necessarily induces a feasible scenario sub problem 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 $S \setminus \bar{S}$ can be expressed as convex combinations of the scenarios in $\bar{S}$, the less feasibility cuts need to be generated.

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 sub-problems 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 sub-problem defined on a scenario s' is added and let $v_{s'}^{\star}$ be its associated optimal value. Problem $P(s')$ is written as follows:

$$v_{s'}^{\star} = \text{minimize } f^{\top} y + \sum_{s \in S} p_s c^{\top} x^s \tag{17}$$

subject to

$$Ay = b, \tag{18}$$

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

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 $\alpha_s^{s'} \geq 0, s \in S$ 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 $v^\star_{s'} = v^\star$.*

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 $x^{s'}$. Considering that (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 $v^\star_{s'} = v^\star$. **Q.E.D.**

As stated in Lemma 1, a sub-problem s', that is defined as a convex combination of the sub-problems $s \in S$, 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. 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., $\bar{y} \in \{y \in \mathscr{Y} \mid Ay = b, w^{j\top} d^s \leq w^{j\top} B^s y, \forall j \in \mathscr{J}, s \in S,\})$, 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 *$\bar{z}^s = q^{i_s^*\top} d^s - q^{i_s^*\top} B^s \bar{y}$, where $i_s^* \in \arg\max_{i\in I} q^{i\top} d^s - q^{i\top} B^s \bar{y}$, $\forall s \in S$.*

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

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

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:

$$\begin{aligned} &= \sum_{s\in S} p_s \Big\{ \text{minimize } z^s \mid q^{i\top} d^s - q^{i\top} B^s \bar{y} \leq z^s, i \in I \Big\} \\ &= \sum_{s\in S} p_s \Big\{ \arg\max_{i\in I} q^{i\top} d^s - q^{i\top} B^s \bar{y} \Big\} \\ &= \sum_{s\in S} p_s \Big\{ q^{i_s^*\top} d^s - q^{i_s^*\top} B^s \bar{y} \Big\} \quad \text{ with } i_s^* \in \arg\max_{i\in I} q^{i\top} d^s - q^{i\top} B^s \bar{y},\ \forall s \in S \\ &= \sum_{s\in S} p_s \bar{z}^s. \textbf{Q.E.D.} \end{aligned}$$

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 oder 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 sub-problem s' in the master problem. We further assume that the artificial scenario s' is defined using a convex combination such that $\alpha_s^{s'} > 0$, $\forall s \in S$. 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 the optimal solutions to $BP(\bar{y},I,J)$ and $BP(\bar{y},\bar{S},I,J)$, respectively; then, $\bar{z}_A^s \geq \bar{z}^s$, $\forall s \in S$.*

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

$$\text{minimize} \quad \sum_{s\in S} p_s z^s \tag{24}$$

$$\text{subject to} \quad c^\top \bar{x}_A^{s'} = \sum_{s\in S} \alpha_s^{s'} z^s \tag{25}$$

$$q^{i\top} d^s - q^{i\top} B^s \bar{y} \leq z^s, \quad \forall i \in I, s \in S. \tag{26}$$

Given $\alpha_s^{s'} > 0$, $s \in S$, and constraint (25), model (24)-(26) does not decompose via the scenarios. Constraints (26) can, however, be reformulated as

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

By recalling that $i_s^* \in \arg\max_{i \in I} q^{ik\top} d^s - q^{ik\top} B^s \bar{y}^k$, $\forall s \in S$, the last inequalities are a direct consequence of Lemma 2. Given that $\bar{z}^s \leq z^s$, $\forall s \in S$, are enforced for all feasible solutions to (24)-(26), we have that the optimal values $\bar{z}_A^s$ satisfy $\bar{z}_A^s \geq \bar{z}^s$, $\forall s \in S$.**Q.E.D.**

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 sub-problem $s \in S$ improves on the one obtained using $BP(I, J)$. We next investigate the actual value of the opportunity loss of not using the scenario sub-problem 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 sub-problem 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 the optimality cuts associated to set I. 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 $c^\top \bar{x}_A^{s'} \neq \sum_{s \in S} \alpha_s^{s'} \bar{z}^s$.

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 $\Delta_A(\bar{y}, I, J) = \frac{p^{\tilde{s}}}{\alpha_{\tilde{s}}^{s'}} \Big(c^\top \bar{x}_A^{s'} - \sum_{s \in S} \alpha_s^{s'} \bar{z}^s \Big)$, where $\tilde{s} \in \arg\min_{s \in S} \frac{p^s}{\alpha_s^{s'}}$.*

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

$$\begin{aligned} c^\top \bar{x}_A^{s'} &\neq \sum_{s \in S} \alpha_s^{s'} \bar{z}^s \\ &\Updownarrow \\ c^\top \bar{x}_A^{s'} &\neq \sum_{s \in S} \alpha_s^{s'} (q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y}). \end{aligned}$$

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 be set to their respective lower bounds, which are specified by the optimality cuts present in $BP(\bar{y}, \bar{S}, I, J)$. As a direct consequence, $c^\top \bar{x}_A^{s'} > \sum_{s \in S} \alpha_s^{s'} (q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y})$ and in the optimal solution to $BP(\bar{y}, \bar{S}, I, J)$, at least of one variable z^s will take on a value that is strictly higher than the lower bound provided by the optimality cuts. 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 $\sigma^s = q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y}$, $\forall s \in S$ (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:

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

We further define the following variables:

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

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

$$\begin{aligned}
\min \quad & \sum_{s\in S} p^s \left(\frac{\beta}{\alpha_s^{s'}} v^s + \sigma^s\right) \\
\text{s.t.} \quad & \sum_{s\in S} \alpha_s^{s'} \left(\frac{\beta}{\alpha_s^{s'}} v^s + \sigma^s\right) = \beta \\
& 0 \le v^s \le 1, \forall s \in S.
\end{aligned}$$

$$\Updownarrow$$

$$\min \quad \sum_{s\in S} \frac{p^s}{\alpha_s^{s'}} v^s \tag{27}$$

$$\text{s.t.} \quad \sum_{s\in S} v^s = \frac{\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s}{\beta} \tag{28}$$

$$0 \le v^s \le 1, \forall s \in S. \tag{29}$$

Model (27)-(29) represents a continuous knapsack problem. Furthermore, considering that $0 < (\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s)/\beta < 1$, then the optimal solution to (27)-(29) is expressed as follows:

$$v^{\tilde{s}} = \frac{\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s}{\beta} \quad , \quad \tilde{s} \in \arg\min_{s\in S} \frac{p^s}{\alpha_s^{s'}} \tag{30}$$

$$v^s = 0 \quad , \quad \forall s \in S \setminus \{\tilde{s}\}. \tag{31}$$

By replacing (30)-(31) in $z^s = (v^a/\alpha_s^{s'})\beta + \sigma^s$, $\forall s \in S$, the optimal solution to $BP(\bar{y},\bar{S},I,J)$ is expressed as:

$$\bar{z}_A^{\tilde{s}} = \frac{\beta - \sum_{s\in S} \alpha_s^{s'} \sigma^s}{\alpha_{\tilde{s}}^{s'}} + \sigma^{\tilde{s}} \quad , \quad \tilde{s} \in \arg\min_{s\in S} \frac{p^s}{\alpha_s^{s'}} \tag{32}$$

$$\bar{z}_A^s = \sigma^s \quad , \quad \forall s \in S \setminus \{\tilde{s}\}. \tag{33}$$

Considering that $\bar{z}^s = q^{i_s^\star \top} d^s - q^{i_s^\star \top} B^s \bar{y} \Rightarrow \bar{z}^s = \sigma^s$, $\forall s \in S$, and given the optimal solution detailed by (32)-(33), one obtains $\Delta_A(\bar{y}, I, J) = \sum_{s \in S} p_s(\bar{z}_A^s - \bar{z}^s) = \frac{p^{\tilde{s}}}{\alpha_{\tilde{s}}^{s'}} \left(c^\top \bar{x}_A^{s'} - \sum_{s \in S} \alpha_s^{s'} \bar{z}^s \right)$. **Q.E.D.**

An important implication of this result is that when creating an artificial scenario, $s^{'}$, assigning weights ($\alpha_s^{s'}$) equal to the probabilities associated with each scenario (e.g. $\alpha_s^{s'} = p_s$) will yield the strongest bound.

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.

To describe these hybrids, we label the **Row Covering** strategy as **RC**, the **Convex Hull** strategy as **CH**, and creating an artificial scenario as **C**. With this notation, we have the following hybrid strategies: **RC+C(**$|\bar{S}|$**)**, wherein scenarios from S are retained via the **Row Covering** strategy and an artificial scenario is created, and **CH+C(**$|\bar{S}|$**)**, which differs in that scenarios are retained via the **Convex Hull** strategy. In both cases, $\bar{S}$ includes one artificial scenario and $|\bar{S}| - 1$ scenarios from S. Similarly, we have the pure strategies **RC(**$|\bar{S}|$**)** and **CH(**$|\bar{S}|$**)**, wherein $\bar{S}$ only contains scenarios retained from S. 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 mixed integer programs. The first will support both pure and hybrid strategies for when scenarios are to be retained through the **Row Covering** strategy; the second 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, i.e., $p_s = p_{\tilde{s}} = p, \forall s, \tilde{s} \in S$. And, given Proposition 4, this implies that the artificial scenario should be created by assigning each non-retained scenario the weight $p/[p(|S| - R)] = 1/(|S| - R)$.

Row Covering-based strategies: **RC($|\bar{S}|$), RC+C($|\bar{S}|$)**

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 do 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 non-retained 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 $l = 1, \ldots, m$, 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 $l = 1, \ldots, m, i = 1, \ldots, n$, 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 $\delta_l^{\tilde{s}s}, \gamma_{li}^{\tilde{s}s}, \; i = 1, \ldots, n$, 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 $d_l^{s'}, l = 1, \ldots, m$, and $B_{li}^{s'}, l = 1, \ldots, m; \; i = 1, \ldots, n$, 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 $l = 1, \ldots, m$. Variable b_l^s indicates whether or not 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 non-retained scenarios are covered. Specifically, we define the binary variables c_{li}^s and $\bar{c}_{li}^s$, $s \in S, l = 1, \ldots, m, i = 1, \ldots, n$, 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 non-retained scenarios. Specifically, we define the continuous variables $e_l^s,\ \ s \in S, l = 1, \dots, m$, to measure the degree to which the artificial scenario does not cover the value d_l^s of a non-retained scenario. We also define the continuous variables $f_{li}^s,\ \ s \in S, l = 1, \dots, m, i = 1, \dots, n$, to measure the degree to which the artificial scenario does not cover the value B_{li}^s of a non-retained scenario.

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

$$\text{maximize} \sum_{s \in S} \sum_{l=1}^{m} b_l^s + \sum_{s \in S} \sum_{l=1}^{m} \sum_{i=1}^{n} c_{li}^s$$

subject to

$$b_l^s \le \sum_{\tilde{s} \in S} \delta_l^{\tilde{s}s} r_{\tilde{s}} + \bar{b}_l^s,\ l = 1, \dots, m, \forall s \in S, \tag{34}$$

$$c_{li}^s \le \sum_{\tilde{s} \in S} \gamma_{li}^{\tilde{s}s} r_{\tilde{s}} + \bar{c}_{li}^s,\ l = 1, \dots, m,\ i = 1, \dots, n, \forall s \in S, \tag{35}$$

$$d_l^{s'} = \sum_{s \in S} d_l^s \alpha_s^{s'},\quad l = 1, \dots, m, \tag{36}$$

$$d_l^s \le d_l^{s'} + e_l^s,\quad l = 1, \dots, m,\ \forall s \in S, \tag{37}$$

$$\bar{b}_l^s \le 1 - \frac{e_l^s}{d_l^s},\quad l = 1, \dots, m,\ \forall s \in S \mid d_l^s > 0, \tag{38}$$

$$B_{li}^{s'} = \sum_{s \in S} B_{li}^s \alpha_s^{s'},\quad l = 1, \dots, m,\ i = 1, \dots, n, \tag{39}$$

$$B_{li}^{s'} - f_{li}^s \le B_{li}^s,\quad l = 1, \dots, m,\ i = 1, \dots, n, \forall s \in S, \tag{40}$$

$$\bar{c}_{li}^s \le 1 - \frac{f_{li}^s}{B_{li}^s},\quad l = 1, \dots, m,\ i = 1, \dots, n,\ \forall s \in S \mid B_{li}^s > 0, \tag{41}$$

$$\sum_{s \in S} r_s \le R, \tag{42}$$

$$\alpha_s^{s'} \le C \frac{p}{p(|S| - R)} (1 - r_s),\quad \forall s \in S, \tag{43}$$

$$\alpha_s^{s'} \ge C\left(\frac{p}{p(|S| - R)} - r_s\right),\quad \forall s \in S, \tag{44}$$

$$\sum_{s \in S} \alpha_s^{s'} = C, \tag{45}$$

$$\alpha_s^{s'} \ge 0, r_s \in \{0, 1\}, \forall s \in S, \tag{46}$$

$$d_l^{s'} \ge 0,\quad l = 1, \dots, m, \tag{47}$$

$$b_l^s \in \{0, 1\}, \bar{b}_l^s \in \{0, 1\}, 0 \le e_l^s \le d_l^s,\quad l = 1, \dots, m, \forall s \in S, \tag{48}$$

$$B_{li}^{s'} \geq 0, \quad l = 1, \ldots, m, \; i = 1, \ldots, n, \tag{49}$$

$$c_{li}^{s} \in \{0,1\}, \bar{c}_{li}^{s} \in \{0,1\}, 0 \leq f_{li}^{s} \leq B_{li}^{s}, \quad l = 1, \ldots, m, \; i = 1, \ldots, n, \forall s \in S. \tag{50}$$

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 (34) 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 $d_l^{\tilde{s}} \geq d_l^s$ or the artificial scenario is constructed in such a way that $d_l^{s'} \geq d_l^s$. Constraints (35) play a similar role albeit with respect to element B_{li}^s being covered by a retained or artificial scenario.

Constraints (36) calculate the d vector for the artificial scenario, whereas constraints (37) calculate the amount (represented by e_l^s) by which the artificial scenario does not cover d_l^s. Note that without sacrifying the optimal solution to this MIP we can presume that $e_l^s \leq d_l^s$. As such, constraint (38) ensures that when we do not have $d_l^{s'} \geq d_l^s$ (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 (39), (40), and (41) serve a similar purpose, albeit for determining $\bar{c}_{li}^s$.

Constraint (42) limits the number of scenarios from S that are retained to be at most R. Constraints (43) and (44) ensure that when an artificial scenario is to be created, it is constructed from scenarios that are not retained. Finally, constraint (45) 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.

Convex Hull-based strategies: **CH**($|\bar{S}|$)**, CH+C**($|\bar{S}|$)

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 can be expressed as a convex combination of scenarios in $\bar{S}$. Like the previous MIP, this MIP makes these choices jointly because Proposition 2 does not require that only scenarios in S be used to construct the convex hull.

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 the convex combination that approximates the vector $d^{\tilde{s}}$ and 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 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 the approximation of scenario $\tilde{s}$, we define the continuous variables $u_l^{\tilde{s}}$, $\tilde{s} \in S$, $l = 1, \ldots, m$ to represent how element l of vector $d^{\tilde{s}}$ is approximated by a convex combination of the scenarios $s \in \bar{S}$. We then define the continuous variables $e_l^{\tilde{s}}$, $\tilde{s} \in S, l = 1, \ldots, m$ to measure the error in that representation. Similarly, we define continuous variables $v_{li}^{\tilde{s}}$ and

$f_{li}^{\tilde{s}}, \tilde{s} \in S,\ l = 1,\ldots,m,\ i = 1,\ldots,n$ to represent the approximation of the matrix element $B_{li}^{\tilde{s}}$ and the error associated with that approximation. As such, the MIP we solve to construct the set $\bar{S}$ is as follows:

$$\text{minimize} \sum_{s\in S}\sum_{l=1}^{m} e_l^s + \sum_{s\in S}\sum_{l=1}^{m}\sum_{i=1}^{n} f_{li}^s$$

subject to

$$u_l^{\tilde{s}} = \sum_{s\in S} \alpha_s^{\tilde{s}} d_l^s,\ l = 1,\ldots,m,\ \forall \tilde{s} \in S, \tag{51}$$

$$v_{li}^{\tilde{s}} = \sum_{s\in S} \alpha_s^{\tilde{s}} B_{li}^s,\ l = 1,\ldots,m,\ i = 1,\ldots,n,\ \forall \tilde{s} \in S, \tag{52}$$

$$\alpha_s^{\tilde{s}} \le \frac{\beta_{s'}^{\tilde{s}}}{|S|-R} + r_s,\ \ \forall s,\tilde{s} \in S, \tag{53}$$

$$\alpha_s^{\tilde{s}} \ge \frac{\beta_{s'}^{\tilde{s}}}{|S|-R} - r_s,\ \ \forall s,\tilde{s} \in S, \tag{54}$$

$$\sum_{s\in S} \alpha_s^{\tilde{s}} = 1,\ \forall \tilde{s} \in S, \tag{55}$$

$$\sum_{s\in S} r_s \le R, \tag{56}$$

$$e_l^{\tilde{s}} \ge u_l^{\tilde{s}} - d_l^{\tilde{s}},\ \ l = 1,\ldots,m,\ \forall \tilde{s} \in S, \tag{57}$$

$$e_l^{\tilde{s}} \ge d_l^{\tilde{s}} - u_l^{\tilde{s}},\ \ l = 1,\ldots,m,\ \forall \tilde{s} \in S, \tag{58}$$

$$f_{li}^{\tilde{s}} \ge v_{li}^{\tilde{s}} - B_{li}^{\tilde{s}},\ \ l = 1,\ldots,m,\ i = 1,\ldots,n,\ \forall \tilde{s} \in S, \tag{59}$$

$$f_{li}^{\tilde{s}} \ge B_{li}^{\tilde{s}} - v_{li}^{\tilde{s}},\ \ l = 1,\ldots,m,\ i = 1,\ldots,n,\ \forall \tilde{s} \in S, \tag{60}$$

$$r_s \in \{0,1\},\ \forall s \in S, \tag{61}$$

$$1 \ge \alpha_s^{\tilde{s}} \ge 0,\ \ \forall s,\tilde{s} \in S, \tag{62}$$

$$C \ge \beta_{s'}^{\tilde{s}} \ge 0,\ \forall \tilde{s} \in S, \tag{63}$$

$$e_l^{\tilde{s}} \ge 0,\ u_l^{\tilde{s}} \in \Re,\ l = 1,\ldots,m,\ \forall \tilde{s} \in S, \tag{64}$$

$$f_{li}^{\tilde{s}} \ge 0,\ v_{li}^{\tilde{s}} \in \Re,\ l = 1,\ldots,m,\ i = 1,\ldots,n,\ \forall \tilde{s} \in S. \tag{65}$$

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 (51) and (52), respsectively. Constraints (53), and (54) ensure that a scenario is assigned the appropriate weight based upon whether it is retained or used to create the artificial scenario. Constraints (55) ensure that each $\tilde{s} \in S$ is approximated by a convex combination. Constraint (56) limits the number of scenarios that are included in $\bar{S}$ to be at most R. Constraints (57) and (58), coupled with the objective, define that $e_l^{s'} = |u_l^{s'} - d_l^{s'}|$. Similarly, Constraints (59) and (60), coupled with the objective, define that $f_{li}^{s'} = |v_{li}^{s'} - B_{li}^{s'}|$. Finally, Constraints (61), (62), (63), (64), and (65) define the decision variables and their domains.

5 Experimental Design

To resolve the issues of whether a PBD should be performed and, if so, how it should be performed, we complement the analysis presented in the previous section with results from an extensive computational study. In this section, we describe the specific problem used to perform our numerical analysis, how the algorithms were implemented, and the characteristics of the test instances.

5.1 Problem studied

We study the effectiveness of the PBD strategies presented on a variant of the fixed charge multi-commodity network design problem wherein there are (potentially) 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 (e.g., Klibi et al. 2010, Klibi and Martel 2012), and they are notoriously hard to solve (e.g., Crainic et al. 2011, 2001).

We consider a directed network with node set N, arc set A, commodity set K, and scenario set S. Each commodity k must be routed from an origin node, o_k, to a destination node, d_k. We consider three settings in which parameters are not known *a priori*: (1) only demands are not known, (2) only capacities are not known, and, (3) both capacities and demands are not known. We only present a model for the last setting here; the others are analogous. In such a model, as both demands and capacity are not known *a priori*, neither are revealed until the second stage of the problem and are both indexed by scenario. The formulation of the stochastic fixed charge multi-commodity network design problem, *CMND(S)*, is

$$\text{minimize} \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{s\in S} p_s\Big(\sum_{k\in K}\sum_{(i,j)\in A} c^k_{ij}x^{ks}_{ij}\Big) \tag{66}$$

subject to

$$\sum_{j\in N^+(i)} x^{ks}_{ij} - \sum_{j\in N^-(i)} x^{ks}_{ji} = d^{ks}_i, \quad \forall\, i\in N, k\in K, s\in S, \tag{67}$$

$$\sum_{k\in K} x^{ks}_{ij} \le u^s_{ij}y_{ij}, \quad \forall\,(i,j)\in A, s\in S, \tag{68}$$

$$y_{ij}\in\{0,1\}, \quad \forall\,(i,j)\in A, \tag{69}$$

$$x^{ks}_{ij} \ge 0, \quad \forall\,(i,j)\in A, k\in K, s\in S, \tag{70}$$

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_{ij}^k being the cost per unit of demand k flowed on arc (i, j). Constraints (67) 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_i^{ks} is either set to 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 (68) guarantee that the same design is used in each scenario, and that the arc capacity (u_{ij}^s) is not violated in any scenario. Finally, Constraints (69) and (70) impose the necessary integrality and non-negativity 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_{ij}^s, and commodity demands, d_i^{ks}, in the *CMND(S)*. Thus, in the *CMND(S)*, scenario $\tilde{s}$ covers scenario s when it models smaller arc capacities ($u^{\tilde{s}} \leq u^s$) and larger commodity demands ($d^{\tilde{s}} \geq d^s$). 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 non-negative), 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.

Finally, we note that for settings where there is uncertainty in B, we add to the network, for each commodity k, an "auxiliary arc" of the form (o_k, d_k) that can only be taken by that commodity. These arcs guarantee feasibility of the instance, regardless of the values of the variables y_{ij}, and model the opportunity for outsourcing a specific shipment. We associate with these arcs a zero fixed cost ($f_{o_k d_k} = 0$) and a high variable cost, $c_{o_k d_k}^k$.

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 *Multi-cut* version of the L-Shaped method (Birge and Louveaux 1988), as previously presented in Algorithm 1 in Section 2. Preliminary tests showed that this version of the method outperformed the original *Single-cut* version (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. Therefore, we also opted for the two phase Benders solution approach developed in McDaniel and Devine (1977). As such, the first phase involves the solution of the linear relaxation of problem (66)-(70). 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 (69) are reintroduced and the Benders algorithm is again applied to produce an optimal solution to the $CMND(S)$.

Regarding the second phase, the master problem is solved via CPLEX, with optimality and feasibility cuts added at nodes of the branch-and-bound tree whenever integer feasible solutions are found. This approach is inspired by 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 by 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 non-dominated cuts are added to the master problem. Also, 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 $\sum_{j \in N^+(i)} y_{ij} \geq 1$, where node i is the origin for some commodity k. Similar inequalities can be added for destination nodes for commodities. 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 $d^{ks} < u_{ij}$, adding constraints of the form $x_{ij}^{ks} \leq d^{ks} y_{ij}$ to (66)-(70) 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 some of the variables x_{ij}^{ks}, we therefore dynamically add these inequalities when solving the linear relaxation of the $CMND(S)$. The collected inequalities $x_{ij}^{ks} \leq d^{ks} y_{ij}$ 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 use 7 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 et al. (2001).

We then generate three sets of scenario for each given instance class. The first set consists of 64 scenarios wherein uncertainty in the demand vector, d^s, is modeled. The second set consists of 64 scenarios wherein uncertainty in the capacity vector, u^s, is modeled. Finally,

Class	$\|N\|$	$\|A\|$	$\|K\|$
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

Table 1: Instance Class Characteristics

the last set consists of 96 scenarios, wherein uncertainty in both vectors is modeled. We used the algorithm presented in Høyland et al. (2003) to generate these scenarios. For the scenarios just described we presume no correlation between the random variables. However, to study whether correlations can impact the performance of the Partial Benders Decomposition strategies proposed, we generate four more sets of scenarios wherein there is uncertainty in demands only. These four sets differ with respect to the percentage (20%, 40%, 60%, and 80%) of commodities that are positively correlated. In summary, we test our strategies on 245 instances of varying structure, stochasticity, and correlation.

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 computational benefits (if any) associated with performing a PBD. We first present results for all strategies wherein Benders (with or without a partial decomposition) is executed for two hours. We then analyze the performance of a few of the strategies when executed for ten hours.

In all experiments, we executed our implementations of Benders decomposition on a cluster of machines with 8 Intel Xeon CPUs running at 2.66 GHz with 32 GB RAM. All linear and mixed integer programs were solved with CPLEX 12. We solved $CMND(S)$ instances with either CPLEX or a Benders-based algorithm, which uses either a partial or a full decomposition. All algorithms were executed with an optimality gap tolerance of 1%. All computation times reported are in seconds.

6.1 Benchmarking the use of partial decomposition

We first study the impact of using PBD on instances wherein there is uncertainty in values in both the matrix, B, and the right-hand-side vector, d. To do so, we benchmark the partial decomposition strategies against both a Benders implementation wherein scenarios are not retained (which we label as **Trad** for "Traditional") and when scenarios are retained, but chosen at random (which we label **Rand**($|\bar{S}|$)) Similarly, we also consider hybrid strategies wherein the scenarios retained are chosen randomly (labeled **Rand+C**($|\bar{S}|$)).

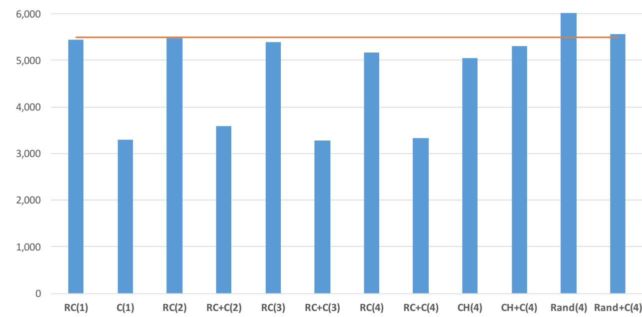
We present in Figures 1(a), 1(b), and 1(c) three performance metrics for diferent strategies. In Figure 1(c), we illustrate the percentage of instances solved to within an optimality tolerance of 1% when executing a strategy. In Figure 1(b), we illustrate the average optimality gap at termination when executing a strategy. In Figure 1(a), we represent the average amount of time until termination when executing a strategy. Note for this last figure, that when an instance is not solved, the time to termination is two hours (or 7,200 seconds).

The figures clearly indicate that performing a partial decomposition, regardless of the strategy employed, improves computational performance. Specifically, all PBD strategies are able to report a significantly smaller optimality gap than when a partial decomposition is not performed. Regarding the pure strategies, we see that **RC**($|\bar{S}|$) and **CH**($|\bar{S}|$) strategies exhibit similar performance, and both perform better on all three performance metrics than the **Rand**($|\bar{S}|$) strategy. Finally, we see that the pure "Creation" strategy, **C(1)**, outperforms the pure "Retention" strategies, **RC**($|\bar{S}|$), **CH**($|\bar{S}|$), and **Rand**($|\bar{S}|$), on all metrics.

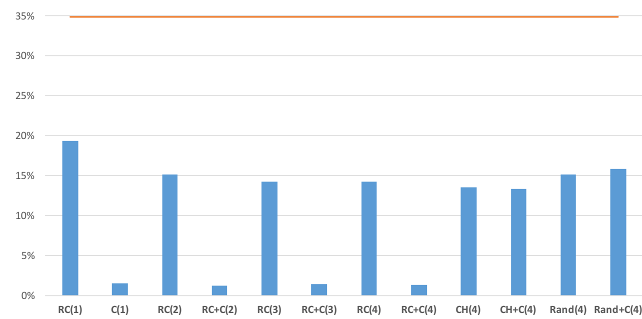
We next turn our attention to instances wherein there is uncertainty in B or d, but not both. We report in Table 2 the same performance metrics (% instances solved, optimality gap at termination, and total time to termination) as in the previous figures and for the same strategies. Here we see that the best strategy depends upon which parameter values are uncertain. When there is only uncertainty in the matrix B, the pure strategy **C(1)** performs the best. However, when there is only uncertainty in the right-hand-side vector d, the hybrid strategies **RC+C**($|\bar{S}|$) perform the best. We again see that the hybrid strategies **RC+C**($|\bar{S}|$) significantly outperform the **CH+C**($|\bar{S}|$) and **Rand+C**($|\bar{S}|$) strategies. As such, we do not include these strategies in the remaining discussion. We also note that the instances wherein there is uncertainty in the right-hand-side vector, d, appear to be easier to solve than those where there is uncertainty in the matrix B.

Finally, to further understand the potential of these PBD strategies, we again execute the algorithm for the hybrid strategies **RC+C**($|\bar{S}|$) for $|\bar{S}| = 2,3,4$ and the pure **C(1)** strategy, but for ten hours. We illustrate the results of these experiments for all three settings (uncertainty in B, uncertainty in d, and uncertainty in both B and d) in Figures 2(a), 2(b), and 2(c).

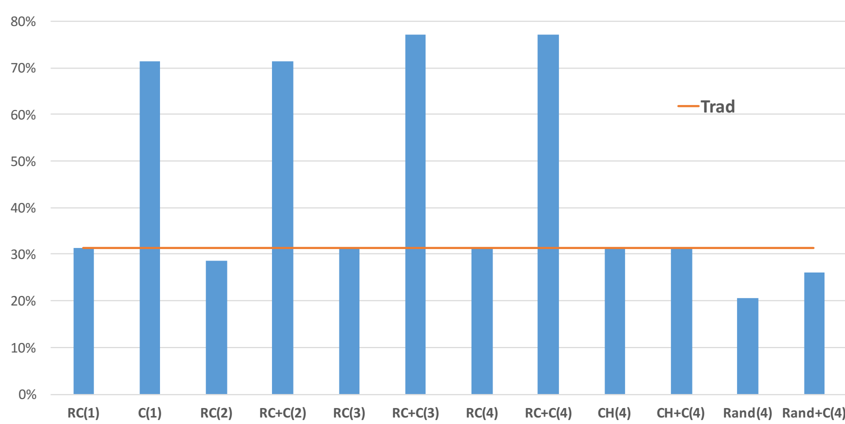
Here, while all four strategies perform well, we see a clearer distinction between the hybrid strategies and the pure strategy. For all three uncertainty settings, the **RC+C(3)** strategy is able



(a) Time to termination (sec.)



(b) Optimality gap at termination

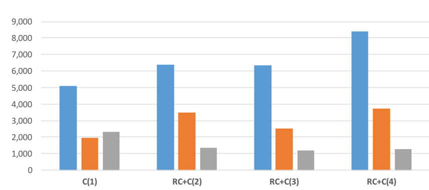


(c) % Instances solved

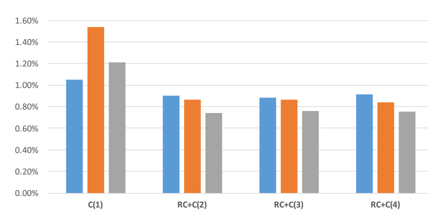
Figure 1: Performance by strategy when uncertainty in both B and d, $|S| = 96$

	Uncertainty in B only			Uncertainty in d only		
	% Solved	Opt. gap	Time (sec.)	% Solved	Opt. gap	Time (sec.)
RC(1)	31.43%	13.86%	5,087.57	37.14%	7.25%	4,610.63
C(1)	94.29%	0.89%	2,299.49	91.43%	3.44%	1,896.77
RC(2)	31.43%	11.88%	5,017.77	40.00%	6.10%	4,977.49
RC+C(2)	88.57%	1.01%	2,264.97	94.29%	0.75%	1,476.80
RC(3)	31.43%	11.31%	5,124.54	40.00%	6.11%	4,852.57
RC+C(3)	82.86%	1.04%	2,399.71	94.29%	0.78%	1,509.74
RC(4)	31.43%	10.11%	5,136.80	40.00%	5.98%	4,711.74
RC+C(4)	91.43%	0.99%	2,133.20	94.29%	0.78%	1,226.03
CH(4)	31.43%	10.12%	5,127.46	40.00%	5.58%	4,913.86
CH+C(4)	31.43%	11.17%	5,136.20	34.29%	6.55%	4,921.51
Rand(4)	28.67%	11.74%	5,266.37	31.33%	31.33%	5,307.61
Rand+C(4)	26.67%	12.31%	5,284.86	27.33%	7.20%	5,627.57
Trad	31.43%	29.33%	5,378.63	40.00%	20.83%	4,713.09

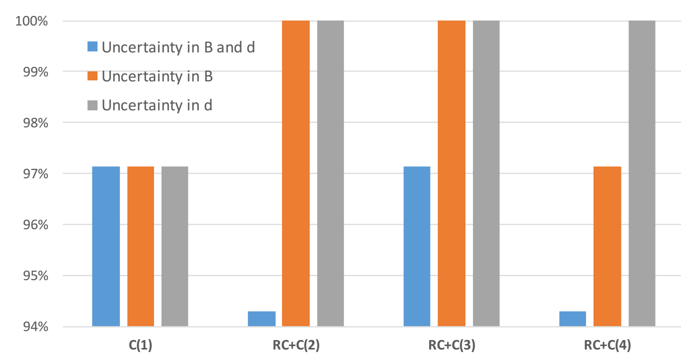
Table 2: Performance by strategy when uncertainty in B or d, $|S| = 64$, 2 hour runtime



(a) Time to termination (sec.)



(b) Optimality gap at termination



(c) % Instances solved

Figure 2: Performance by strategy, 10 hour runtime.

to solve more instances and produce a smaller optimality gap, on average. The **RC+C($|\bar{S}|$)** strategies do require more execution time (again on average), however.

Finally, recalling that the instances wherein there is uncertainty in the B matrix contain an "auxiliary" arc that guarantees their feasibility, we run one more set of experiments on these instances wherein that auxiliary arc is removed. We report the results of the hybrid **RC+C($|\bar{S}|$)** strategies as well as the **C(1)** strategy and the traditional benders implementation (**Trad**) in Table 3. We see that while the **RC+C(2)** and **C(1)** strategies perform comparably on the instances wherein there is uncertainty in both B and d, the hybrid strategy performs best on the instances wherein there is only uncertainty in B.

	Uncertainty in B and d			Uncertainty in B		
	% Solved	Opt. gap	Time (sec.)	% Solved	Opt. gap	Time (sec.)
RC+C(2)	91.43%	0.86%	2,171.97	100.00%	0.74%	1,607.71
RC+C(3)	88.57%	0.90%	2,416.83	94.29%	0.80%	1,430.54
RC+C(4)	85.71%	1.27%	2,363.97	97.14%	2.15%	1,209.37
Trad	37.14%	20.07%	4,868.74	51.43%	19.49%	6,122.11
C(1)	91.43%	0.83%	1,957.20	97.14%	0.99%	2,086.23

Table 3: Performance with no Auxiliary arc, 2 hour runtime

Having established that performing a proposed partial decomposition strategy yields superior computational performance, we next turn our attention to understanding why it (particularly when employing a hybrid strategy) performs so well.

6.2 Analyzing the impact of partial decomposition

We next seek a fuller picture of the performance seen when performing partial decomposition. In this section, we only consider the hybrid strategies, **RC+C($|\bar{S}|$)**. Recall that, in our Benders implementation, we first solve the linear programming relaxation of P with Algorithm 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. Notice that we define an iteration of Algorithm 1 as one execution of the while loop, wherein a relaxation of the problem is solved and violated cuts are generated.

Then, to measure the impact of partial decomposition on convergence, we illustrate in Figure 3 a comparison of the number of iterations required (on average) to solve the linear programming relaxation of P for each strategy. We see a clear trend in that performing a PBD significantly decreases the number of iterations needed to converge. We also see that while creating a scenario speeds up convergence significantly, the hybrid strategy speeds up convergence even more. However, there are diminishing marginal returns with respect to convergence as the convergence rate for **RC+C(4)** is not much faster than that of **RC+C(3)**.

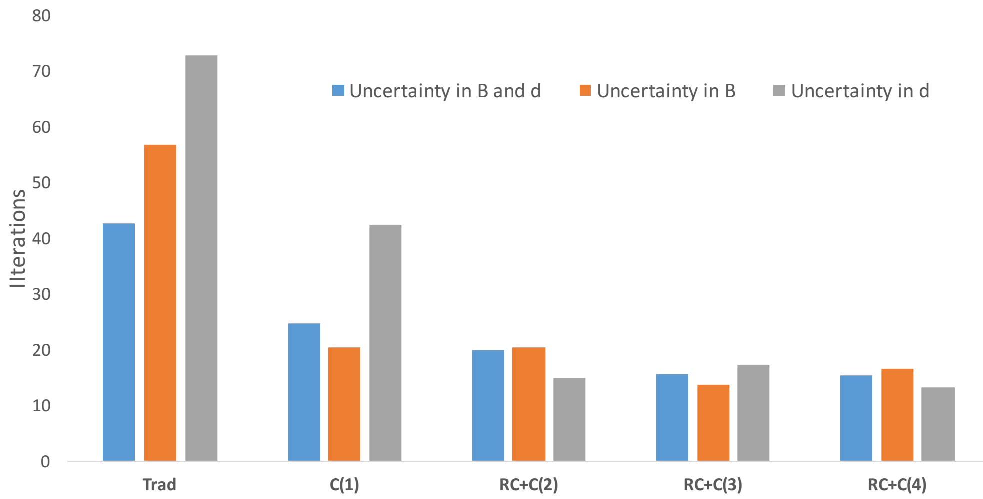


Figure 3: Convergence rate by strategy, all uncertainty settings

One statistic related to convergence is the number of Benders cuts generated. We next report in Figure 4 the number of optimality cuts generated for different strategies when the algorithm was allowed to execute for ten hours. Recalling that only the instances where there is uncertainty in d can be infeasible (due to the presence of arcs that guarantee delivery), we present the number of feasibility cuts generated by different strategies in Table 4. In both cases, we report the average number of cuts generated for each strategy, considering both the cuts generated when solving the linear programming relaxation of P with Algorithm 1 and when solving P via a branch-and-bound-based Benders implementation.

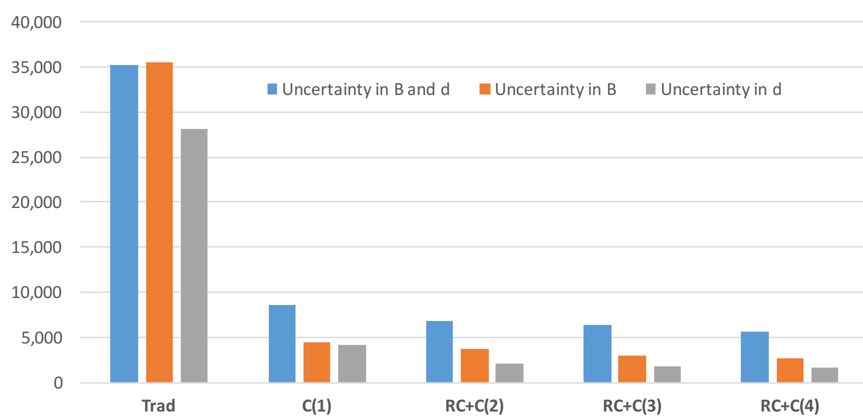


Figure 4: # Optimality cuts by strategy, all uncertainty settings

We see that performing a PDB 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 cuts, the hybrid strategies do so even more. We also note that for the instances where there is only uncertainty in d, there are again diminishing marginal returns

	Trad	C(1)	RC+C(2)	RC+C(3)	RC+C(4)
Feasibility cuts	20,555.31	780.20	31.14	8.00	9.03

Table 4: Feasibility cuts by strategy (uncertainty in d)

with respect to how many scenarios are retained; the decrease in both the number of optimality and feasibility cuts is small (negative for feasibility cuts) when you retain three scenarios as opposed to two. Regardless, these results clearly correlate with the speed-up in convergence that we observed in Figure 3. Continuing our analysis of the number of cuts generated by partial decomposition strategy used, we recall that for some of the instances wherein there is only uncertainty in d there are correlations in the volumes. However, our computational results indicate that the level of correlation has almost no effect on the number of cuts generated for any of the decomposition strategies studied. Tables 12 and 11 in the Appendix report the number of optimality and feasibility cuts by strategy and correlation.

To understand why performing partial decomposition can have such an impact on the number of optimality cuts generated, we next measure, by strategy, the gap between the expected recourse term, $\sum_{s\in S} p_s(\sum_{k\in K}\sum_{(i,j)\in A} c_{ij}^k x_{ij}^{ks})$, at the first iteration of solving the linear programming relaxation (LPR) of P and when the LPR has been solved. We report these gaps in Table 5 and see that while simply retaining a scenario (**RC(1)**) reduces the gap, it is the artificial scenario that has the greatest impact. We note that these results are for Instance 5 from Class 7 and a setting wherein both B and d are uncertain.

	Trad	C(1)	RC(1)	RC+C(2)	RC+C(3)	RC+C(4)
Recourse gap at 1st iteration	100.00%	0.62%	94.63%	0.08%	0.30%	0.48%

Table 5: Impact of strategy on expected recourse, uncertainty in B and d

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 PBD mitigates this issue by measuring the Hamming distance between successive vectors of first stage variable values (Δ *next*), which represent a network design in our application. We report averages of these distances in Table 6 along with the Hamming distance from each design to the final design produced (Δ *final*). Comparing the values of Δ *next* we see that the partial decomposition strategies lead to a more stable search than a traditional Benders decomposition wherein no scenario information is retained in the master. Comparing the values of Δ *final*, we see that the PBD strategies also direct the search towards better solutions. Interestingly, we note that the pure strategy, **C(1)**, yields the most stable search (the smallest value of Δ *next*), yet the largest value for Δ *final* amongst the PBD strategies. We hypothesize that while the single artificial scenario may stabilize the search, it may do so too much, and thus the search is often far from the portion of the solution space that contains the optimal design.

Finally, one of our motivations for the use of a Partial Benders Decomposition was that it should yield a higher quality primal solution earlier in the search. To evaluate whether

Method	Uncertainty in B and d Δ next	Δ final	Uncertainty in B Δ next	Δ final	Uncertainty in d Δ next	Δ final
Trad	35.57	21.02	34.51	20.96	27.51	17.66
C(1)	17.02	19.90	14.08	20.26	5.11	12.79
RC+C(2)	22.91	20.69	21.48	21.29	10.33	10.70
RC+C(3)	21.50	18.62	21.03	19.46	10.15	10.49
RC+C(4)	21.91	18.03	21.57	18.65	10.06	10.23

Table 6: Stability (measured by Hamming distance) by strategy

performing a PBD does in fact speed up the search for a high quality primal solution, we again focus on solving the linear relaxation of our stochastic program, P. We then present in Figure 5, for each decomposition strategy used, the gap at each iteration of our Benders implementation between the objective function value of the primal solution produced and the dual bound. We note that these results are for Instance 5 from Class 7 and a setting wherein both B and d are uncertain. We have limited the data illustrated to just the first 25 iterations. We see that while the pure **C(1)** strategy produces an optimal primal solution much faster than **Trad**, the hybrid strategies do so even more quickly, with the more scenarios retained, the more quickly the high quality primal solution is found. From our results, we conclude that retaining a scenario and creating an artificial scenario serve different purposes with respect to speeding up a Benders implementation: retaining a scenario speeds up the search for a high-quality primal solution, while creating an artificial scenario strengthens the dual bound.

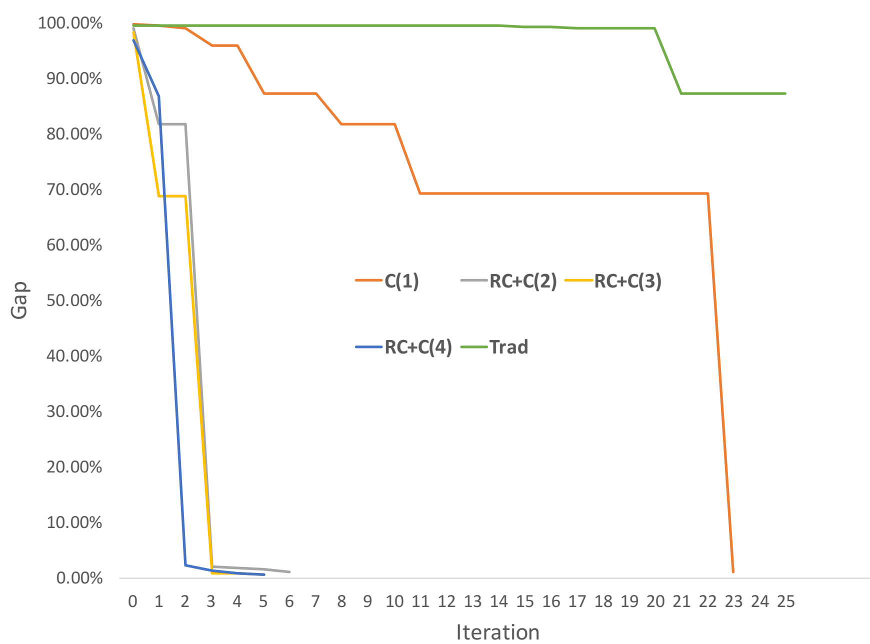


Figure 5: Optimality gap by iteration

7 Conclusions and future work

We have established, both theoretically and numerically, the benefits associated with performing a Partial Benders Decomposition (PBD) strategy. Namely, retaining scenario sub-problem 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. 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 reserach 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. Finally, 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.

Lastly, we have examined the effectiveness of a PBD strategy on one class of stochastic programs. Namely, two-stage programs that have continuous recourse. Yet the strategy can also be applied to problems with multiple stages and/or integer recourse. Such problems would require both theoretical and computational development, which we intend to pursue in future research papers.

Acknowledgments

Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery Grant program. We also gratefully acknowledge the support of Fonds de recherche du Québec through their infrastructure grants and of Calcul Québec and Compute Canada for access to their high-performance computing infrastructure. The third author expresses her acknowledgments to the Research Council of the University of Bergamo for the visiting professor research grant support 2015. While working on this project, the first author was Adjunct Professor with the Department of Computer Science and Operations Research of the Université de Montréal.

References

Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

S. Batun, B. T. Denton, T. R. Huschka, and A. J. Schaefer. Operating room pooling and parallel surgery processing under uncertainty. *INFORMS Journal on Computing*, 23(2):220–237, 2011.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

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

J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, 2011.

J. R. Birge and F. V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34:384–392, 1988.

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

A. M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers and operations research*, 32(6):1429–1450, 2005.

G. Côté and M. A. Laughton. Large-scale mixed integer programs: Benders-type heuristic. *European Journal of Operational Research*, 16(3):327–333, 1984.

T. G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1):73–99, 2001.

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

R. S. de Camargo, G. de Miranda Jr., and H. P. L. Luna. Benders decomposition for hub location problems with economies of scale. *Transportation Science*, 43(1):86–97, 2009.

M. Fischetti, D. Salvagnin, and A. Zanette. A note on the selection of benders' cuts. *Mathematical Programming*, 124:175–182, 2010.

M. Fischetti, I. Ljubicn, and M. Sinni. Redesigning benders decomposition for large-scale facility location. *Management Science*, 2016.

A. M. Geoffrion. Elements of large-scale mathematical programming part 1: Concepts. *Management Science*, 11:652–675, 1970a.

A. M. Geoffrion. Elements of large-scale mathematical programming part 2: Synthesis of algorithms and bibliography. *Management Science*, 11:676–691, 1970b.

A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(2):237–260, 1972.

A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by benders decomposition. *Management Science*, 20(5):822–844, 1974.

K. Holmberg. On using approximations of the benders master problem. *European Journal of Operational Research*, 77(1):111–125, 1994.

J. N. Hooker and G. Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96: 33–60, 2003.

K. Høyland, M. Kaut, and S. W. Wallace. A heuristic for moment-matching scenario generation. *Computational optimization and applications*, 24(2-3):169–185, 2003.

W. Klibi and A. Martel. Scenario-based supply chain network risk modeling. *European Journal of Operational Research*, 223:644–658, 2012.

W. Klibi, A. Martel, and A. Guitouni. The design of robust value-creating supply chain networks: A critical review. *European Journal of Operational Research*, 203:283–293, 2010.

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

T. L. Magnanti and R. T. Wong. Accelerating benders decomposition algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

D. McDaniel and M. Devine. A modified benders partitioning algorithm for mixed integer programming. *Management Science*, 24:312–319, 1977.

S. A. MirHassani, C. Lucas, G. Mitra, E. Messina, and C. A. Poojari. Computational solution of capacity planning models under uncertainty. *Parallel Comput.*, 26(5):511–538, April 2000a. ISSN 0167-8191. doi: 10.1016/S0167-8191(99)00118-0. URL `http://dx.doi.org/10.1016/S0167-8191(99)00118-0`.

S. A. MirHassani, C. Lucas, G. Mitra, E. Messina, and C. A. Poojari. Computational solution of capacity planning models under uncertainty. *Parallel Computing*, 26:511–538, 2000b.

G. Nemhauser and L. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

N. Papadakos. Practical enhancements to the magnanti-wong method. *Operations Research Letters*, 36: 444–449, 2008.

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

C. A. Poojari and J. E. Beasley. Improving benders decomposition using a genetic algorithm. *European Journal of Operational Research*, 199:89–97, 2009.

R. Rahmaniani, T.G. Crainic, M. Gendreau, and W. Rei. The Benders Decomposition Algorithm: A Literature Review. Publication CIRRELT-2016-30, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada, 2016.

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

R. T. Rockafellar and S. Uryasev. The fundamental risk quadrangle in risk management, optimization and statistical estimation. *Surveys in Operations Research and Management Science*, 18:33–53, 2013.

T. J. V. Roy. Cross decomposition for mixed integer programming. *Mathematical Programming*, 25(1): 46–63, 1983.

H. D. Sherali and B. J. Lunday. On generating maximal nondominated benders cuts. *Annals of Operations Research*, 210:57–72, 2013.

R. V. Slyke and R. J.-B. Wets. L-shaped programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

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

G. Zakeri, A. D. Philpott, and D. M. Ryan. Inexact cuts in benders decomposition. *SIAM Journal on Optimization*, 10(3):643–657, 2000.

Appendix - Tables Supporting Figures

In this appendix, we present tables of results supporting the claims and figures presented in Section 6.

Strategy	Solved	Opt. Gap	Time to termination
RC(1)	31.43%	19.36%	5,439.69
C(1)	71.43%	1.51%	3,302.54
RC(2)	28.57%	15.17%	5,472.97
RC+C(2)	71.43%	1.22%	3,594.40
RC(3)	31.43%	14.24%	5,390.29
RC+C(3)	77.14%	1.45%	3,283.94
RC(4)	31.43%	14.21%	5,175.60
RC+C(4)	77.14%	1.31%	3,329.94
CH(4)	31.43%	13.54%	5,054.40
CH+C(4)	31.43%	13.32%	5,297.51
Rand(4)	20.67%	15.08%	6,032.75
Rand+C(4)	26.00%	15.80%	5,567.05
Trad	31.43%	31.43%	5,501.17

Table 7: Results supporting Figures 1(c), 1(b), and 1(a); Performance metrics when uncertainty in B and d, by strategy.

Solved	Uncertainty in B and d	Uncertainty in B	Uncertainty in d
RC+C(2)	94.29%	100.00%	100.00%
RC+C(3)	97.14%	100.00%	100.00%
RC+C(4)	94.29%	97.14%	100.00%
C(1)	97.14%	97.14%	97.14%
Opt. Gap	Uncertainty in B and d	Uncertainty in B	Uncertainty in d
RC+C(2)	0.90%	0.87%	0.74%
RC+C(3)	0.88%	0.86%	0.76%
RC+C(4)	0.91%	0.84%	0.75%
C(1)	1.05%	1.54%	1.21%
Time to termination	Uncertainty in B and d	Uncertainty in B	Uncertainty in d
RC+C(2)	6,371.63	3,499.09	1,342.66
RC+C(3)	6,345.09	2,537.91	1,177.43
RC+C(4)	8,399.97	3,737.80	1,290.40
C(1)	5,079.29	1,961.57	2,337.31

Table 8: Results supporting Figures 2(c), 2(b), and 2(a); Performance metrics with a 10 hour runtime, by strategy.

Strategy	Uncertainty in B and d	Uncertainty in B	Uncertainty in d
RC+C(2)	20.03	20.49	14.89
RC+C(3)	15.57	13.71	17.26
RC+C(4)	15.43	16.63	13.31
C(1)	24.83	20.49	42.40
Trad	42.63	56.86	72.74

Table 9: Results supporting Figure 3 - # Iterations to converge, by strategy

Strategy	Uncertainty in B and d	Uncertainty in B	Uncertainty in d
RC+C(2)	6,787.83	3,635.37	2,051.09
RC+C(3)	6,393.80	2,983.51	1,780.37
RC+C(4)	5,607.63	2,610.89	1,700.46
C(1)	8,639.06	4,435.71	4,185.23
Trad	35,242.69	35,455.94	28,183.80

Table 10: Results supporting Figure 4, # Optimality cuts, by strategy

	Correlation			
Strategy	0.20	0.40	0.60	0.80
C(1)	4,064.63	4,158.91	4,100.57	4,085.26
RC+C(2)	2,019.86	2,035.69	2,016.66	2,039.77
RC+C(3)	1,848.66	1,826.74	1,831.49	1,852.37
RC+C(4)	1,765.23	1,792.66	1,746.29	1,815.06

Table 11: Optimality cuts by correlation and strategy, uncertainty in d

	Correlation			
Strategy	0.20	0.40	0.60	0.80
C(1)	810.17	823.86	809.51	811.46
RC+C(2)	23.54	22.46	18.83	25.77
RC+C(3)	15.69	14.54	13.57	19.60
RC+C(4)	11.09	10.83	11.11	14.66

Table 12: Feasibility cuts by correlation and strategy, uncertainty in d