An Adaptive Partition-based Approach for Solving Two-stage Stochastic Programs with Fixed Recourse

Yongjia Song[†], James Luedtke[‡]

[†]Virginia Commonwealth University, Richmond, VA, ysong3@vcu.edu
[‡]University of Wisconsin-Madison, Madison, WI, jrluedt1@wisc.edu

December 2, 2014

Abstract

We study an adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. A partition-based formulation is a relaxation of the original stochastic program, and we study a finitely converging algorithm in which the partition is adaptively adjusted until it yields an optimal solution. A solution guided refinement strategy is developed to refine the partition by exploiting the relaxation solution obtained from a partition. In addition to refinement, we show that in the case of stochastic linear programs, it is possible to merge some components in a partition, without weakening the corresponding relaxation bound, thus allowing the partition size to be kept small. We also show that for stochastic linear programs with simple recourse, there exists a small partition that yields an optimal solution. The size of this partition is independent of the number of scenarios used in the model. Our computational results show that the proposed adaptive partition-based approach converges very fast to a small partition for our test instances. In particular, on our test instances the proposed approach outperforms basic versions of Benders decomposition and is competitive with the state-of-art methods such as the level method for stochastic linear programs with fixed recourse.

1 Introduction

We study an adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. Specifically, we consider the following scenario-based formulation of a stochastic program:

$$\min_{x\in X} c^\top x + \sum_{k\in N} f_k(x), \tag{1}$$

where N is a given set of scenarios and for each $k \in N$:

$$f_k(x) := \min_{y^k\in\mathbb{R}^{n_2}_+} \{d^\top y^k \mid T^k x + W y^k \geq h^k\}. \tag{2}$$

$X \subseteq \mathbb{R}^{n_1}_+$ is a closed deterministic feasible set, $d \in \mathbb{R}^{n_2}$, and $T^k \in \mathbb{R}^{m_2\times n_1}, h^k \in \mathbb{R}^{m_2}$ for each scenario $k \in N$. We assume that the problem $\min\{c^\top x \mid x \in X\}$ and problem (1) are both feasible and bounded. The finite scenario stochastic program (1) is motivated by the sample average approximation (SAA) approach [7]. The number of scenarios required to obtain a good approximation may be very large in some cases. This motivates computational methods that can efficiently solve problems with a large scenario set. In (1), we assume the recourse matrix $W \in \mathbb{R}^{m_2 \times n_2}$ is the same for all scenarios (fixed recourse) and we also assume the cost vector d is fixed. The extensive formulation of (1) is given by:

$$z^* = \min c^\top x + \sum_{k \in N} d^\top y^k \tag{3a}$$

$$\text{s.t. } T^k x + W y^k \geq h^k, \ \forall k \in N \tag{3b}$$

$$x \in X, \ y^k \in \mathbb{R}^{n_2}_+, \ \forall k \in N. \tag{3c}$$

We study an adaptive partition-based approach for solving the two-stage stochastic program (3). A partition $\mathcal{N} = \{P_1, P_2, \cdots, P_L\}$ of the scenario set is a collection of subsets of the scenarios such that $P_1 \cup P_2 \cup \cdots \cup P_L = N$, and $P_i \cap P_j = \emptyset, \forall i, j \in \{1, 2, \ldots, L\}, i \neq j$. Given a partition $\mathcal{N}$, we obtain a relaxation of (3) by aggregating scenario constraints (3b) together, and replacing $\sum_{k \in P} y^k$ by y^P for each component $P \in \mathcal{N}$:

$$z_{\mathcal{N}} = \min c^\top x + \sum_{P \in \mathcal{N}} d^\top y^P \tag{4a}$$

$$\text{s.t. } \bar{T}^P x + W y^P \geq \bar{h}^P, \ \forall P \in \mathcal{N} \tag{4b}$$

$$x \in X, \ y^P \in \mathbb{R}^{n_2}_+, \ \forall P \in \mathcal{N}, \tag{4c}$$

where $\bar{T}^P := \sum_{k \in P} T^k, \bar{h}^P := \sum_{k \in P} h^k, \forall P \in \mathcal{N}$. We call (4) the *partition-based master problem* with respect to partition $\mathcal{N}$.

Lemma 1 *Given a partition $\mathcal{N} = \{P_1, P_2, \ldots, P_L\}$, the partition-based master problem* (4) *is a relaxation for* (3)*, and $z_{\mathcal{N}} \leq z^*$.*

Proof: Let $(\bar{x}, \bar{y})$ be an optimal solution of (3), and let $x = \bar{x}$, $y^P = \sum_{k \in P} \bar{y}^k$ for each $P \in \mathcal{N}$, then $c^\top x + \sum_{P \in \mathcal{N}} d^\top y^P = z^*$, and (x, y) is feasible to (4). □

We are interested in developing an adaptive partition-based approach that solves a sequence of problems of the form (4) with adaptively chosen partition $\mathcal{N}$, which either solves (3) exactly or finds a solution that has the corresponding objective value within ϵ of the optimal objective value z^*.

Definition 1 *A partition $\mathcal{N}$ is* ϵ-sufficient, *if $z_{\mathcal{N}} \geq z^* - \epsilon$. In particular, when $z_{\mathcal{N}} = z^*$, i.e., $\mathcal{N}$ is* 0*-sufficient, we say $\mathcal{N}$ is* completely sufficient.

The goal is to identify a completely sufficient partition or an ϵ-sufficient partition with a small ϵ, which has a much smaller size than the original scenario-based problem (3). Unless otherwise stated, let ϵ be a fixed parameter, and we refer to ϵ-sufficient partitions simply as sufficient partitions.

The idea of using aggregation for solving general large-scale optimization problems has a long history. Classically in [35, 36], constraint aggregation and variable aggregation are applied to obtain relaxations of linear programs (LP). This idea is then extended to solve a wider range of problems, including integer programs [15] and dynamic programs [2], etc. See [24] for a survey on some classic aggregation-based methods. In stochastic programming, aggregation has been studied to obtain either an optimality bound [6] or a hierarchy of bounds [28]. For multistage stochastic linear programs with a general probability distribution, [34] studies aggregation and disaggregation with respect to the underlying information structure of the probability distribution. A more general concept of aggregation is proposed in [25], which considers constraints aggregation across different scenarios as well as constraints within each scenario.

A natural extension to this aggregation approach is to identify a sufficient partition $\mathcal{N}$ through an iterative algorithm. Three questions arise when designing such an algorithm. The first question is how to determine if a partition $\mathcal{N}$ is sufficient, and when it is not sufficient, how to refine the partition. We provide a sufficient and necessary condition for a partition $\mathcal{N}$ to be completely sufficient. This condition provides guidance on how to refine a partition. The second question is to estimate the required size of a sufficient partition. There exists a trivial sufficient partition $\mathcal{N} = \{\{k\}_{k\in N}\}$, but it is not interesting because it does not reduce the size of the problem. For two-stage stochastic LPs with simple recourse, we show that there exists a completely sufficient partition whose size does not depend on the number of scenarios N. This suggests that the partition-based approach has the potential to solve a sequence of relatively small problems to achieve an optimal solution of (3). Another question is if it is possible to merge components of a partition back together dynamically, so that we are able to prevent the partition from growing too large during the procedure. When X is a polyhedron, we propose a merging strategy that guarantees that the relaxation is not weakened after merging. This analysis yields a new algorithmic framework for two-stage stochastic linear programs that is fundamentally different from Benders decomposition (or its L-shaped variant [32]) due to the difference in the master problem that is solved. We conduct extensive computational experiments on test instances having fixed recourse and technology matrices (i.e., $T^k \equiv T, k \in N$). We find that the adaptive partition-based algorithm outperforms basic versions of Benders decomposition (including the multi-cut implementation [7] and the single-cut L-shaped algorithm [32]). In addition, we found the adaptive partition-based approach to be often competitive, and occasionally superior, to the L-shaped algorithm regularized with the level method [13, 19] on our test instances.

Our work can be seen as an application of the adaptive partition-based algorithmic template proposed by Bienstock and Zuckerberg [5], which generalizes an algorithm for solving precedence constrained production scheduling problems. Our approach is also similar to the partition scheme proposed to solve the scenario-based formulation of a Conditional Value-at-Risk (CVaR) minimization stochastic program [12]. Computational experiments in [12] empirically demonstrate that the iterative partition scheme can keep the partition size small, and hence is computationally beneficial when the number of scenarios is large. We generalize this idea to two-stage stochastic programs with fixed recourse, with the CVaR minimization problem as a special case. We prove that there exists a small completely sufficient partition for the case of simple recourse, providing a theoretical explanation for the promising empirical results obtained in [12].

Besides [5] and [12], iterative algorithms based on adaptive aggregation have also been studied in other contexts. An adaptive clustering algorithm that converges in a finite number of iterations is proposed in [18] for the case of variable aggregation. In stochastic programming, a classic iterative solution scheme, called the sequential approximation method (SAM), is designed to improve the bounds for the expected second-stage objective value via partition refinements [11, 16, 17]. In that framework, the partition is performed on the support of the continuous distribution of the random variables in the model, rather than a scenario set. The corresponding partition-based problem is constructed using conditional expectation. This idea has been used in the level decomposition solution framework [13]. It has also been used for solving stochastic programs that arise in a variety of applications, e.g., stochastic network interdiction problems [9], and stochastic appointment scheduling problems [10]. Empirically, the size of the partition is not too large before a good approximation solution is obtained in these cases. However, even if a conditional expectation is used to reduce the computational effort, e.g., via stratified sampling [22], the partition-based problem is still hard to solve. Our approach uses sampling in a different way. We assume a (possibly very large) set of scenarios is given a priori, which is motivated by the SAA approach. In addition, we directly perform the partition on this given scenario set. In this context, an iterative refinement algorithm that converges finitely for the multi-stage setting is proposed in [8]. We propose an alternative approach for refining the partition using the information of dual optimal solutions, and we show how this refinement strategy can lead to small partitions in the case of simple recourse.

In the SAA framework, the idea of adaptive aggregation has been applied to manage the optimality cuts in the master problem to enhance the computational performance of Benders decomposition for solving two-stage stochastic programs [31] and multi-stage stochastic programs [33]. [31] introduces an adaptive multi-cut method that generalizes the single-cut (L-shaped) and multi-cut methods. The method dynamically adjusts the aggregation level of the optimality cuts in the master program. The key difference between the method in [31] and our proposed approach, is in the form of the master problem solved at each iteration. The master problem in [31] has the form of a Benders master problem, whereas our master problem is the partition problem (4), which introduces aggregate second-stage variables. Our methods also differ in the techniques used for updating partitions.

In Section 2 we describe a general adaptive partition-based algorithm for solving two-stage stochastic programs with fixed recourse. In Section 3 we analyze the case of simple recourse structure and show that a small completely sufficient partition exists. We show how the approach can be appied to problems with expected value constraints in Section 4, and in Section 5 we compare the computational performance of the proposed approach with alternative approaches.

2 The adaptive partition-based algorithm

A general adaptive partition-based algorithm works as follows. We start with an initial partition, e.g., $\mathcal{N} = \{N\}$. We solve the partition-based master problem (4) with respect to this partition $\mathcal{N}$ and let the first-stage solution be $\hat{x}$. The obtained optimal objective value $z_{\mathcal{N}}$ is then a lower bound for the optimal objective value z^* of the original scenario-based problem (3). Given a fixed first-stage solution $\hat{x}$, for each scenario k, we solve the second-stage problem (2), or its dual problem:

$$f_k(\hat{x}) = \max_{\lambda^k \in \mathbb{R}^{m_2}_+} \{(h^k - T^k\hat{x})^\top \lambda^k \mid W^\top \lambda^k \leq d\}. \tag{5}$$

If the second-stage problem (2) with this fixed $\hat{x}$ is infeasible, we solve the feasibility problem associated with the second-stage problem (2) by solving:

$$g_k(\hat{x}) := \max_{\mu^k \in \mathbb{R}^{m_2}_+} \{(h^k - T^k\hat{x})^\top \mu^k \mid W^\top \mu^k \leq 0, e^\top \mu^k \leq 1\}. \tag{6}$$

If $g_k(\hat{x}) > 0$, then the second-stage problem (2) with fixed $\hat{x}$ is infeasible.

If the second-stage problems (2) are feasible for all scenarios $k \in N$, then $z(\hat{x}) := \sum_{k\in N} f_k(\hat{x}) + c^\top \hat{x}$ is an upper bound for z^*. If the gap between the current best upper bound z^U and lower bound $z_{\mathcal{N}}$ is larger than the termination threshold ϵ, we can reduce the gap by performing a refinement on the current partition $\mathcal{N}$.

Definition 2 *$\mathcal{N}'$ is a refinement of $\mathcal{N}$, if $\forall P' \in \mathcal{N}'$, $P' \subseteq P$ for some $P \in \mathcal{N}$, and $|\mathcal{N}'| > |\mathcal{N}|$.*

It is clear that $z_{\mathcal{N}'} \geq z_{\mathcal{N}}$ if $\mathcal{N}'$ is a refinement of $\mathcal{N}$. If (2) are feasible for all scenarios $k \in N$ with a relaxation solution $\hat{x}$, we need to perform refinement on $\mathcal{N}$ when $\mathcal{N}$ is not sufficient. We also need to perform refinement if for some scenario $k \in N$, (2) is infeasible with $\hat{x}$. In the extreme case, partition $\{\{k\}_{k\in N}\}$ is a refinement of all possible partitions except itself. Therefore, an algorithm based on iterative partition refinement returns a sufficient partition in a finite number of steps. We describe this algorithm in Algorithm 1.

Algorithm 1 An iterative partition refinement algorithm for solving (3)

```
Suppose a stopping threshold $\epsilon \geq 0$, and an initial partition $\mathcal{N}_0$ are given.
Set $z^U := +\infty$, and $t := 0$.
repeat
    Solve (4) with respect to $\mathcal{N}_t$, obtain the optimal objective value $z_{\mathcal{N}_t}$ and the corresponding
    optimal solution $\hat{x}$.
    Solve (2) for each scenario $k \in N$ with $\hat{x}$.
    if $\exists k \in N$ that (2) is infeasible then
        Refine $\mathcal{N}_t$ to obtain $\mathcal{N}_{t+1}$.
    else
        Update $z^U \leftarrow \min\{z^U, z(\hat{x})\}$.
    end if
    Refine $\mathcal{N}_t$ if $z^U - z_{\mathcal{N}_t} > \epsilon$
    $t \leftarrow t + 1$.
until $z^U - z_{\mathcal{N}_t} \leq \epsilon$
```

Proposition 1 *Algorithm 1 converges in finitely many iterations.*

2.1 Construction of a completely sufficient partition from an optimal solution

A completely sufficient partition can be constructed from a dual optimal solution $(\hat{\pi}, \hat{\lambda})$ of (3), when the deterministic feasible region X is a polyhedral set, $X = \{x \in \mathbb{R}^{m_2}_+ \mid Ax = b\}$, where $A \in \mathbb{R}^{m_1 \times n_1}, b \in \mathbb{R}^{m_1}$. Without loss of generality, we assume $n_1 \geq m_1$. The dual of (3) is:

$$\max b^\top \pi + \sum_{k\in N} (h^k)^\top \lambda^k \tag{7a}$$

$$\text{s.t. } A^\top \pi + \sum_{k\in N} (T^k)^\top \lambda^k \leq c \tag{7b}$$

$$W^\top \lambda^k \leq d, \ \forall k \in N \tag{7c}$$

$$\pi \text{ free}, \ \lambda^k \in \mathbb{R}^{m_2}_+, \ \forall k \in N. \tag{7d}$$

Proposition 2 *Let $(\hat{\pi}, \hat{\lambda})$ be an optimal solution of* (7)*, and $\mathcal{N}$ be a partition of the scenario set N that satisfies $\hat{\lambda}^k = \hat{\lambda}^{k'}, \forall k, k' \in P, P \in \mathcal{N}$. Then partition $\mathcal{N}$ is completely sufficient.*

Proof: (This argument is inspired by arguments in [5].) Given a partition $\mathcal{N}$, the dual of the partition-based master problem (4) with $X = \{x \in \mathbb{R}^{n_2}_+ \mid Ax = b\}$ is equivalent to (7) plus the set of constraints: $\lambda^k = \lambda^{k'}, \forall k, k' \in P, P \in \mathcal{N}$, using a vector λ^P for each component $P \in \mathcal{N}$ to represent these common vectors $\lambda^k, \forall k \in P$. Moreover, adding the constraints $\lambda^k = \lambda^{k'}, \forall k, k' \in P, P \in \mathcal{N}$

to (7) does not change its optimal objective value, since an optimal solution $\hat{\lambda}$ satisfies these. □

Proposition 2 provides a completely sufficient partition $\mathcal{N}$ using an optimal dual solution, when X is a polyhedron. In Section 3, we derive an upper bound for the size of this partition in the special case of a two-stage stochastic program with simple recourse.

2.2 Solution guided refinement

Any refinement strategy can be used in Algorithm 1, although a refinement rule that converges to a small sufficient partition with a small number of iterations would be preferred. When X is a polyhedron, given an optimal solution of (7), we can construct a completely sufficient partition according to Proposition 2. However, during Algorithm 1, we only have a relaxation solution $\hat{x}$ that is optimal to the current partition-based master problem (4). The idea of solution guided refinement is to use $\hat{x}$ to guide the refining operation.

Given a first-stage solution $\hat{x}$ and a set of scenarios $P \subseteq N$, we define a second-stage problem with respect to $\hat{x}$ and P:

$$\bar{f}_P(\hat{x}) := \min_{y^P \in \mathbb{R}^{n_2}_+} \{d^\top y^P \mid W y^P \geq \bar{h}^P - \bar{T}^P \hat{x}\}. \tag{8}$$

If (2) is feasible for all $k \in P$, then $\bar{f}_P(\hat{x})$ is a lower bound for $\sum_{k\in P} f_k(\hat{x})$. In fact, any dual optimal solution $\bar{\lambda}^P$ of (8) is feasible to (5) for each scenario $k \in P$. Therefore, $\bar{f}_P(\hat{x}) = \sum_{k\in P}(h^k - T^k\hat{x})^\top \bar{\lambda}^P \leq \sum_{k\in P} f_k(\hat{x})$. Lemma 2 shows a sufficient and necessary condition when this lower bound is exact.

Lemma 2 *Let $\hat{x} \in X$ and $P \subseteq N$. Assume problem* (2) *is feasible for each $k \in P$. Then $\bar{f}_P(\hat{x}) = \sum_{k\in P} f_k(\hat{x})$ if and only if there exists a vector $\bar{\lambda} \in \mathbb{R}^{m_2}_+$ such that $\bar{\lambda}$ is an optimal solution of* (5) *for each scenario $k \in P$.*

Proof: We first prove sufficiency. Given a vector $\bar{\lambda} \in \mathbb{R}^{m_2}_+$ that is an optimal solution of (5) for each scenario $k \in P$, then $\bar{\lambda}$ is a feasible dual solution to (8). Therefore, $\bar{f}_P(\hat{x}) \geq (\bar{h}^P - \bar{T}^P\hat{x})^\top\bar{\lambda} = \sum_{k\in P}(h^k - T^k\hat{x})^\top\bar{\lambda} = \sum_{k\in P} f_k(\hat{x})$, and hence $\bar{f}_P(\hat{x}) = \sum_{k\in P} f_k(\hat{x})$.

We now prove necessity. Suppose there is no vector $\bar{\lambda} \in \mathbb{R}^{m_2}_+$ such that $\bar{\lambda}$ is an optimal solution of (5) for all $k \in P$, we show that $\bar{f}_P(\hat{x}) < \sum_{k\in P} f_k(\hat{x})$. In fact, for any dual optimal solution $\hat{\lambda}$ of (8), because of the assumption, there exists at least one scenario $\bar{k} \in P$ such that $\hat{\lambda}$ is not optimal to (5) for scenario $\bar{k}$. For this scenario $\bar{k}$, $f_{\bar{k}}(\hat{x}) > (h^{\bar{k}} - T^{\bar{k}}\hat{x})^\top\hat{\lambda}$, and for all other scenarios $k \in P, k \neq \bar{k}$, $f_k(\hat{x}) \geq (h^k - T^k\hat{x})^\top\hat{\lambda}$. Therefore, $\sum_{k\in P} f_k(\hat{x}) = f_{\bar{k}}(\hat{x}) + \sum_{k\in P, k\neq\bar{k}} f_k(\hat{x}) > \bar{f}_P(\hat{x})$.
□

If $\hat{x}$ is an optimal first-stage solution of (4) with respect to a partition $\mathcal{N}$, then $c^\top\hat{x} + \sum_{P\in\mathcal{N}} \bar{f}_P(\hat{x}) = z_{\mathcal{N}}$. Theorem 1 shows a sufficient and necessary condition when $z_{\mathcal{N}} = z^*$.

Theorem 1 *Let $\hat{x}$ be an optimal solution of the partition-based master problem* (4) *with partition $\mathcal{N}$, then $z_{\mathcal{N}} = z^*$ if and only if there exists a set of optimal solutions $\{\lambda^k\}_{k\in N}$ of* (5) *with respect to $\hat{x}$ that satisfies $\lambda^k = \lambda^{k'}, \forall k, k' \in P, P \in \mathcal{N}$.*

Proof: Since $\hat{x}$ is an optimal solution of the partition-based master problem (4) with partition $\mathcal{N}$, $z_{\mathcal{N}} = c^\top\hat{x} + \sum_{P\in\mathcal{N}} \bar{f}_P(\hat{x})$. According to Lemma 2, for each component $P \in \mathcal{N}$, $\bar{f}_P(\hat{x}) = \sum_{k\in P} f_k(\hat{x})$ if and only if there exists a vector $\bar{\lambda} \in \mathbb{R}^{m_2}_+$ such that $\bar{\lambda}$ is an optimal solution of (5) for each scenario

$k \in P$. Therefore, $z_{\mathcal{N}} = z(\hat{x}) = c^\top \hat{x} + \sum_{k \in N} f_k(\hat{x})$, if and only if $\sum_{P \in \mathcal{N}} \bar{f}_P(\hat{x}) = \sum_{k \in N} f_k(\hat{x})$, which holds if and only if there exists a set of optimal solutions $\{\lambda^k\}_{k \in N}$ of (5) that satisfies $\lambda^k = \lambda^{k'}, \forall k, k' \in P, P \in \mathcal{N}$. □

The consistency requirement that $\lambda^k = \lambda^{k'}, \forall k, k' \in P, P \in \mathcal{N}$ is restrictive, especially when λ^k is not a scalar. However, if the fixed recourse matrix W has a special structure such that there is only a small number of distinct values that an extreme point optimal solution λ^k of (5) could take, then it is hopeful that there exists a small sufficient partition.

Theorem 1 can be used as a guide for how to refine a partition given a solution $\hat{x}$. If there exists a component $P \in \mathcal{N}$ with $k \neq k' \in P$ such that $\lambda^k \neq \lambda^{k'}$, we can perform a refinement on partition $\mathcal{N}$ by splitting this component P according to the different values that $\lambda^k, k \in P$ take. In particular, if we split a partition P into subsets that correspond to all different values $\{\lambda^k\}_{k \in P}$, we call it a *complete refinement* with respect to $\{\lambda^k\}_{k \in P}$.

We perform refinement in a similar way when there exists some scenario $k \in P$ such that problem (2) is infeasible. In this case, we solve (6) for scenarios k that are infeasible, and perform refinements for these scenarios using dual solutions $\bar{\mu}^k$ to guide the refinement. Algorithm 2 summarizes our solution-guided refinement strategy. In our implementation of the algorithm, the check that $\lambda^k = \lambda^{k'}$ is replaced with the relaxed criterion $|(\lambda^k_j - \lambda^{k'}_j)/(\lambda^k_j + 10^{-5})| < \delta, \forall j$, where $\delta = 10^{-5}$ is a given threshold. A less restrictive heuristic strategy based on "quasi-collinearity" [21] could also be applied.

Algorithm 2 Complete refinement of a component P

```
Consider a partition component $P \in \mathcal{N}$, suppose a solution $\hat{x}$ of (4) is given. Let $P_1, P_2 = \emptyset$.
for scenario $k \in P$ do
  Solve the second-stage problem (5) with $\hat{x}$.
  if (5) is feasible then
    Obtain a dual optimal solution $\lambda^k$, and let $P_1 = P_1 \cup \{k\}$.
  else
    Solve the second-stage feasibility problem (6) with $\hat{x}$. Obtain a dual optimal solution $\mu^k$,
    and let $P_2 = P_2 \cup \{k\}$.
  end if
end for
Let $\{K_1^1, K_1^2, \ldots, K_1^{M_1}\}$ be a partition of $P_1$ that $\lambda^k = \lambda^{k'}, \forall k, k' \in K_1^m, m = 1, 2, \ldots, M_1$, and let
$\{K_2^1, K_2^2, \ldots, K_2^{M_2}\}$ be a partition of $P_2$ that $\mu^k = \mu^{k'}, \forall k, k' \in K_2^m, m = 1, 2, \ldots, M_2$.
Remove the component $P$ from partition $\mathcal{N}$.
Add components $K_1^1, K_1^2, \cdots, K_1^{M_1}$, and $K_2^1, K_2^2, \cdots, K_2^{M_2}$ to partition $\mathcal{N}$.
```

The complete refinement strategy fully exploits the information of $\{\lambda^k\}_{k \in P}$ based on the current relaxation $\hat{x}$. If (2) is feasible for all scenarios $k \in P$, after the complete refinement, the lower bound $\bar{f}_P(\hat{x})$ matches the true value $\sum_{k \in P} f_k(\hat{x})$ according to Lemma 2. If we perform a complete refinement for all the components, which we call a *fully complete refinement*, then the lower bound $c^\top \hat{x} + \sum_{P \in \mathcal{N}} \bar{f}_P(\hat{x})$ exactly matches the true objective $z(\hat{x})$.

2.3 Adaptive partition with merging

Iterative refinement on a partition leads to finite convergence of Algorithm 1. However, the size of the partition-based master problem increases as more refinements are performed. In this section, we consider putting some components in a partition back together without weakening the relaxation bound. Proposition 3 shows that if the deterministic feasible set X in (3) is a polyhedron, e.g., $X = \{x \in \mathbb{R}^{n_1}_{+} \mid Ax = b\}$, then we are able to merge components using the information of the dual optimal solution of (4).

Proposition 3 *Given a partition $\mathcal{N} = \{P_1, P_2, \ldots, P_L\}$, let $\{\bar{\lambda}^P\}_{P \in \mathcal{N}}$ be a dual optimal solution of* (4) *with this partition. Let $\{I_1, I_2, \ldots, I_{L'}\}$ be a partition over $\{1, 2 \ldots, L\}$ that is composed of sets of indices that correspond to the same $\{\bar{\lambda}^P\}_{P \in \mathcal{N}}$ values, and let $\bar{\mathcal{N}} = \{\bigcup_{l \in I_1} P_l, \bigcup_{l \in I_2} P_l, \ldots, \bigcup_{l \in I_{L'}} P_l\}$. Then $z_{\mathcal{N}} = z_{\bar{\mathcal{N}}}$.*

Proof: The conclusion is immediate by applying Proposition 2 on partition $\mathcal{N}$, considering each component $P \in \mathcal{N}$ as a scenario. □

Algorithm 3 Merging operation

```
Suppose a partition $\mathcal{N} = \{P_1, P_2, \cdots, P_L\}$, and an optimal dual solution $\{\bar{\lambda}^P\}_{P \in \mathcal{N}}$ of (4) with $\mathcal{N}$ are given.
Let $\{I_1, I_2, \ldots, I_{L'}\}$ be a partition over $\{1, 2, \ldots, L\}$ such that $\lambda^{P_l} = \lambda^{P_{l'}}, \forall l, l' \in I_t, t = 1, 2, \ldots, L'$.
Return a merged partition $\bar{\mathcal{N}} = \{\bigcup_{l \in I_1} P_l, \bigcup_{l \in I_2} P_l, \ldots, \bigcup_{l \in I_{L'}} P_l\}$.
```

The merging operation is given in Algorithm 3. The motivation of this merging operation is to use a smaller partition $\bar{\mathcal{N}}$ to replace the current partition $\mathcal{N}$, while the relaxation bound of the partition-based master problem with $\bar{\mathcal{N}}$ is the same as the one with $\mathcal{N}$. This is particularly true when the fixed recourse matrix W has some special structure that ensures that there exists a small number of distinct $\bar{\lambda}^P$ values. Let $\mathcal{N}_t$ be the current partition obtained after a refinement on the previous partition $\mathcal{N}_{t-1}$. If the lower bound is not improved after the refinement, i.e., $z_{\mathcal{N}_t} = z_{\mathcal{N}_{t-1}}$, then a cycle may be caused by the merging operation. Therefore, we only perform the merging operation when the relaxation bound is strictly improved from the previous iteration. We obtain a variant of Algorithm 1 by performing the merging operation before we refine the current partition $\mathcal{N}_t$, if the relaxation bound $z_{\mathcal{N}_t}$ is strictly better than $z_{\mathcal{N}_{t-1}}$. This merging strategy is also mentioned in [5].

The merging operation cannot be generalized to non-polyhedral set X, since we do not have the dual solutions that guide the merging operation as in the case of polyhedral X. Please refer to [29] for a heuristic extension for the special case when X is a finite set of integer vectors.

2.4 Partial refinement

The aforementioned refinement strategy is based on the current relaxation solution $\hat{x}$, which may not be the solution that gives the current best upper bound. We now propose a refinement strategy that only performs the merging operation if $\hat{x}$ is the current best solution, i.e., $z^U = z(\hat{x})$. If $z(\hat{x}) > z^U$, we can perform a *partial refinement* on $\mathcal{N}$. The motivation is that, since $\hat{x}$ is not the current best solution, it is unnecessary to perform a complete refinement to match the lower bound $\bar{f}_P(\hat{x})$ with the true value $\sum_{k \in P} f_k(\hat{x})$ for each component $P \in \mathcal{N}$. We only need to refine the partition just enough such that $c^\top \hat{x} + \sum_{P \in \mathcal{N}} \bar{f}_P(\hat{x}) > z^U$, which guarantees that $\hat{x}$ is suboptimal with respect to the refined partition. While a complete refinement also serves the purpose, it yields a larger refined partition. We perform partial refinement in a greedy manner described in Algorithm 4.

Algorithm 4 Partial refinement

```
Suppose a partition $\mathcal{N}$, an optimal solution $\hat{x}$ of (4) with respect to $\mathcal{N}$, the current best upper bound $z^U$, and a set of optimal solutions $\{\lambda^k\}_{k\in N}$ of (5) with $\hat{x}$ are given.
for each component $P \in \mathcal{N}$ do
    Calculate the true objective value $\sum_{k\in P} f_k(\hat{x})$.
    Calculate the lower bound $\bar{f}_P(\hat{x})$.
end for
Sort the gap $\sum_{k\in P} f_k(\hat{x}) - \bar{f}_P(\hat{x})$ in decreasing order.
Sequentially put components $P$ into a collection $\mathcal{P}$ according to this order until $z_{\mathcal{N}} + \sum_{P\in\mathcal{P}}(\sum_{k\in P} f_k(\hat{x}) - \bar{f}_P(\hat{x})) > z^U$.
Perform complete refinements for all components in $\mathcal{P}$ with respect to $\{\lambda^k\}_{k\in N}$.
```

In summary, we propose several different refinement and merging strategies. Given a solution $\hat{x}$ of the partition-based master problem (4) with partition $\mathcal{N}$:

1. No-Merge: Perform fully complete refinement on each component $P \in \mathcal{N}$ with respect to $\hat{x}$ according to Algorithm 5.

2. Merge-All: If the relaxation bound of the current partition improves the previous one, we perform a merging operation according to Algorithm 3 and then apply the fully complete refinement on the merged partition. If the relaxation bound is not improved, we skip the merging operation and perform the fully complete refinement directly on the current partition.

3. Merge-Partial: First, we determine if solution $\hat{x}$ is the current best solution according to the upper bound $z(\hat{x})$. If $\hat{x}$ is the current best solution, we perform a merging operation according to Algorithm 3 and then apply the fully complete refinement on the merged partition. Otherwise, we perform partial refinement according to Algorithm 4 with respect to the current $\hat{x}$, and the best bound so far.

We show their performance in our computational experiments in Section 5.

2.5 Relationship to Benders decomposition

We next discuss the relationship between Algorithm 1 and Benders decomposition (see, e.g., [7]) and the L-shaped algorithm [32] (i.e., the single-cut variant of Benders decomposition). The idea of Benders decomposition is to iteratively approximate the epigraph of function $f_k(x)$, $F_k := \{(x,\theta^k) \in \mathbb{R}^{n_1} \times \mathbb{R} \mid \theta^k \geq f_k(x)\}, \forall k \in N$, by constructing a convex relaxation defined by a set of valid inequalities that are generated during the algorithm. This relaxation is also called the *Benders master problem* as follows:

$$\min_{x\in X} c^\top x + \sum_{k\in N} \theta^k \tag{9a}$$

$$\text{s.t. } \theta^k \geq G^k x + g^k,\ (G^k, g^k) \in \mathcal{G}^k,\ \forall k \in N, \tag{9b}$$

$$L^k x \geq l^k,\ (L^k, l^k) \in \mathcal{L}^k,\ \forall k \in N, \tag{9c}$$

where $\mathcal{G}^k$ and $\mathcal{L}^k$ are collections of optimality cuts and feasibility cuts, respectively, which are valid inequalities that have been generated for each scenario $k \in N$ so far through the algorithm. The

Benders master problem is a relaxation of (3) in that it contains only a partial set of constraints that are necessary to describe the set F_k. Given an optimal solution $(\hat{x}, \{\hat{\theta}^k\}_{k\in N})$ of the Benders master problem (9), the second-stage problem (2) is solved for each $k \in N$. If (2) is feasible, we obtain an optimal dual solution $\hat{\lambda}^k$ from (5). The inequality $\theta^k \geq (h^k - T^k x)^\top \hat{\lambda}^k$ is then valid for F_k, and if it is violated by $(\hat{x}, \{\hat{\theta}^k\}_{k\in N})$, we call it a *Benders optimality cut*, and add it to the collection $\mathcal{G}^k$ in (9b). If the second-stage problem (2) is infeasible, we obtain an extreme direction $\bar{\mu}^k$ associated with the cone defined by $\{\mu^k \in \mathbb{R}^{m_2}_+ \mid W^\top \mu^k \leq 0\}$ by solving (6). Then $(h^k - T^k x)^\top \bar{\mu}^k \leq 0$ is a valid inequality that cuts off the current relaxation solution $\hat{x}$, which we call a *Benders feasibility cut*, and we add it to the collection $\mathcal{L}^k$ in (9c). The single-cut variant of Benders decomposition uses only a single variable Θ to approximate the epigraph of $\sum_{k\in N} f_k(x)$, and uses optimality cuts of the form $\Theta \geq \sum_{k\in N}(h^k - T^k x)^\top \hat{\lambda}^k$ (so that a single cut is obtained per iteration). Benders decomposition is similar to the adaptive partition-based approach in the sense that a subproblem (2) with fixed $\hat{x}$ is also solved for each scenario. The key difference is in the master problem solved by the two methods. Benders decomposition solves a master problem of the form (9), which is then updated by adding Benders cuts. The master problem in the proposed adaptive partition-based approach is the problem (4), which is updated by changing the partition, leading to a change in both the constraints and the variables (the aggregated second-stage variables) of the master problem. Improvements of Benders decomposition can be obtained by using regularization techniques, such as those in [26, 20, 19]. In Section 5, we compare our adaptively refined partition approach to basic variants of Benders decomposition and to the level method [19].

A crucial goal of the partition refinement approach is to obtain a small sufficient partition, and for the case of simple recourse, we show in Section 3 that there exists a sufficient partition whose size is independent of the number of scenarios. This result is similar to the result in [27] regarding critical scenarios in the context of Benders decomposition. Specifically, Ruszczyński and Świętanowski [27] define critical scenarios as being those scenarios whose cuts define more than one active constraint in the Benders master problem, and observe that the number of such scenarios is bounded by the number of first-stage decision variables. This observation is then used to significantly reduce the size of the master problem. The most significant difference between the concept of critical scenarios and our work is that the master problem being solved is different. Critical scenarios are used for reducing the master problem in Benders decomposition. Since the master problem we use involves both first-stage variables and (aggregated) second-stage variables, the results of [27] are not applicable. In particular, because our master problem includes these additional variables, the analysis to demonstrate existence of a small sufficient partition is different. Indeed, we are only able to show such a partition exists under the assumption of simple recourse, whereas the bound on critical scenarios does not require this assumption.

3 Existence of a small completely sufficient partition for stochastic programs with simple recourse

A two-stage stochastic LP with joint simple recourse can be written as follows.

$$\min c^\top x + \sum_{k\in N} y_k \tag{10a}$$
$$\text{s.t. } Ax = b \tag{10b}$$
$$T^k x + y_k e \geq h^k, \ \forall k \in N \tag{10c}$$
$$x \in \mathbb{R}^{n_1}_+, \ y_k \in \mathbb{R}_+, \ \forall k \in N, \tag{10d}$$

where $e = (1, 1, \dots, 1)^\top$, $c \in \mathbb{R}^{n_1}, A \in \mathbb{R}^{m_1 \times n_1}, b \in \mathbb{R}^{m_1}$, and $T^k \in \mathbb{R}^{m_2 \times n_1}, h^k \in \mathbb{R}^{m_2}$ for $k \in N$. This model penalizes the expected maximum violation among all the constraints in a scenario. However, a model in which a penalty is imposed separately for each individual constraint can be reduced to this case by considering each row of each scenario as a different scenario. Thus, if the original problem has N scenarios and m_2 rows in each scenario, the modified problem would have $N' = Nm_2$ scenarios and a single row in each scenario. Our results do not depend on the number of scenarios, so the increase in number of scenarios from this reduction is not a concern.

The penalty coefficient $d = 1$ for y_k variables in the objective of (10) is without loss of generality. Problems that have scenario-dependent coefficients $d_k \in \mathbb{R}_+$ can be transformed into (10). In these cases, we can introduce a new variable $\xi_k := d_k y_k$ for each scenario $k \in N$, and constraint (10c) is equivalent to $d_k T^k x + \xi_k e \geq d_k h^k$. We can then re-define $d_k T^k, d_k h^k$ as T^k, h^k, respectively, so that we have a formulation (10) with $d_k = 1, \forall k \in N$.

Let $\hat{x}$ be a first-stage optimal solution of the partition-based master problem for (10) with partition $\mathcal{N}$. With fixed $\hat{x}$, we solve a trivial second-stage problem for each scenario $k \in N$:

$$\min_{y_k \in \mathbb{R}_+} \{y_k \mid y_k e \geq h^k - T^k \hat{x}\}, \tag{11}$$

whose dual is:

$$\max_{\lambda^k \in \mathbb{R}^{m_2}_+} \{(h^k - T^k \hat{x})^\top \lambda^k \mid e^\top \lambda^k \leq 1\}. \tag{12}$$

Based on the structure of (12), there are at most $m_2 + 1$ possible values for an optimal solution $\hat{\lambda}^k$ of (12): if $h^k_i - T^k_i \hat{x} \leq 0, \forall i = 1, 2, \dots, m_2$, then $\hat{\lambda}^k_i = 0, \forall i$, otherwise, $\hat{\lambda}^k_{\bar{i}} = 1$, and $\hat{\lambda}^k_i = 0, \forall i \neq \bar{i}$, where $\bar{i} \in \arg\max_i \{h^k_i - T^k_i \hat{x}\}$.

Algorithm 5 Complete refinement for simple recourse

```
Suppose a partition $\mathcal{N}$ and a set of optimal solutions $\{\hat{\lambda}^k\}_{k \in N}$ of (12) with fixed $\hat{x}$ are given.
Let $K^0 = \{k \in N \mid \hat{\lambda}^k = \mathbf{0}\}$.
Let $K^i = \{k \in N \mid \hat{\lambda}^k = e^i\}, \forall i = 1, 2, \dots, m_2$.
for each $P \in \mathcal{N}$ do
    Remove component $P$ from partition $\mathcal{N}$.
    Add components $P \cap K^0$, $P \cap K^1$, $\cdots$, $P \cap K^{m_2}$ if they are nonempty.
end for
```

Algorithm 5 is an adaptation of Algorithm 2 in the case of simple recourse. We see that after a fully complete refinement, the size of the refined partition is at most $m_2 + 1$ times as large as the original partition.

We next show that because of simple recourse, there exists a small completely sufficient partition. Let $(\hat{\pi}, \hat{\lambda})$ be an extreme point optimal solution of the dual of (10):

$$\max\ b^\top \pi + \sum_{k \in N} (h^k)^\top \lambda^k \tag{13a}$$

$$\text{s.t. } A^\top \pi + \sum_{k \in N} (T^k)^\top \lambda^k \leq c \tag{13b}$$

$$e^\top \lambda^k \leq 1,\ \forall k \in N \tag{13c}$$

$$\pi \text{ free},\ \lambda^k \in \mathbb{R}^{m_2}_+,\ \forall k \in N. \tag{13d}$$

Define $K_1 = \{k \in N \mid \hat{\lambda}^k = \mathbf{0}\}, K_2(i) = \{k \in N \mid \hat{\lambda}^k = e^i\}, i = 1, 2, \dots, m_2$, and $K_3 = N \setminus (K_1 \bigcup (\bigcup_{i=1}^{m_2} K_2(i)))$.

Proposition 4 *Partition $\mathcal{N} = \{K_1, K_2(1), K_2(2), \dots, K_2(m_2), \{k\}_{k\in K_3}\}$ is completely sufficient, and $|K_3| \le n_1 - m_1$.*

Proof: According to Proposition 2, $\mathcal{N}$ is completely sufficient. We now show that $|K_3| \le n_1 - m_1$. According to the definition, K_3 is composed of the following two subsets: $K_3^= = \{k \in N \mid \sum_i \hat{\lambda}_i^k = 1\}$, i.e., there are at least two nonzero components in vector $\hat{\lambda}^k$ that sum up to 1, and $K_3^< = \{k \in N \mid \sum_i \hat{\lambda}_i^k < 1\}$. For any extreme point solution of formulation (13), the number of binding constraints is no less than the number of variables, $m_1 + |N|m_2$. Among these binding constraints, at most n_1 of them come from the side constraints (13b), thus there are at least $m_1 + |N|m_2 - n_1$ binding constraints in the system $\{\lambda_i^k \ge 0, \sum_{i=1}^{m_2} \lambda_i^k \le 1\}$. Let G be the set of (k,i) pairs for all the nonzero components λ_i^k in $K_3^=$. $|G| \ge 2|K_3^=|$ since we have at least two nonzero components in each scenario. Let T be the set of (k,i) pairs for all the fractional components in $K_3^<$. Since we have at least one nonzero component in each scenario, $|T| \ge |K_3^<|$. We summarize the number of variables and number of binding constraints for all cases as follows:

Index	Binding constraints	Size
$k \in K_1$	$\lambda_i^k = 0, \forall i$	$\|K_1\|m_2$
$k \in K_2$	$\sum_i \lambda_i^k = 1, \lambda_i^k = 0, \forall i \ne \bar{i}$	$\|K_2\| + \|K_2\|(m_2-1) = \|K_2\|m_2$
$k \in K_3^=$	$\sum_i \lambda_i^k = 1, \lambda_i^k = 0, \forall (k,i) \notin G$	$\|K_3^=\| + \|K_3^=\|m_2 - \|G\|$
$k \in K_3^<$	$\lambda_i^k = 0, \forall (k,i) \notin T$	$\|K_3^<\|m_2 - \|T\|$

We need at least $m_1 + |N|m_2 - n_1$ binding constraints out of all possible binding constraints shown in the above table. Thus

$$(|K_1|m_2) + (|K_2|m_2) + (|K_3^=| + |K_3^=|m_2 - |G|) + (|K_3^<|m_2 - |T|) \ge m_1 + |N|m_2 - n_1.$$

Recalling that $|K_1| + |K_2| + |K_3^=| + |K_3^<| = |N|$, we then have: $|G| + |T| \le |K_3^=| + (n_1 - m_1)$. Next, because $|G| \ge 2|K_3^=|$ and $|T| \ge |K_3^<|$, it follows that $2|K_3^=| + |K_3^<| \le |G| + |T| \le |K_3^=| + (n_1 - m_1)$, from which we conclude that $|K_3^=| + |K_3^<| = |K_3| \le n_1 - m_1$. □

The size of this completely sufficient partition, at most $n_1 - m_1 + m_2 + 1$, is independent of the number of scenarios $|N|$ in the model. We only need to solve a partition-based mster LP of a much smaller size than (10) to obtain an optimal solution if $n_1 - m_1 + m_2 + 1$ is very small compared to $|N|$.

We next provide an example that shows that the bound $n_1 - m_1 + m_2 + 1$ is tight for the case $m_2 = 1$ and $m_1 = 0$, in which case the bound becomes $n_1 + 2$. We let $N = \{1, \dots, n_1 + 2\}$ and show that the only completely sufficient partition is the trivial one that has all $n_1 + 2$ scenarios in different components, matching the bound. Consider the problem

$$\begin{aligned}
\min\ & \sum_{k=1}^{n_1} c_k x_k + \sum_{k=1}^{n_1+2} y_k \\
\text{s.t. } & x_k + y_k \ge 1,\ \forall k = 1, 2, \dots, n_1, && (14) \\
& y_{n_1+1} \ge 1, && (15) \\
& y_{n_1+2} \ge -1 && (16) \\
& x_k \ge 0,\ \forall k = 1, 2, \dots, n_1,\ y_k \ge 0,\ \forall k = 1, 2, \dots, n_1 + 2,
\end{aligned}$$

where we assume that $0 < c_1 < c_2 < \cdots < c_{n_1} < 1$. The optimal solution has $x^* = e$, $y^*_k = 0$ for $k \in N \setminus \{n_1+1\}$, and $y^*_{n_1+1} = 1$, with optimal objective value $z^* = \sum_{k=1}^{n_1} c_k + 1$. Let $1 \leq i < j \leq n_1 + 2$ be any two scenarios and $P = \{i, j\}$. Define the partition $\mathcal{N}_P = \{P, \{k\}_{k \in N \setminus P}\}$ which has size $n_1 + 1$. We show that any such partition yields $z_{\mathcal{N}_P} < z^*$, and hence is not completely sufficient. Any other partition $\mathcal{N}'$ would have $\mathcal{N}_P$ as a refinement of $\mathcal{N}'$ for some P, and so this establishes that the only completely sufficient partition is $\mathcal{N} = \{\{k\}_{k\in N}\}$. We consider all possible cases for $P = \{i, j\}$. First, suppose $i = n_1 + 1$ and $j = n_1 + 2$. Then the partition problem replaces (15) and (16) with $y_P \geq 0$, and replaces the terms $y_{n_1+1} + y_{n_1+2}$ in the objective with y_P. The optimal solution has $\hat{x} = e$, $\hat{y}_k = 0, \forall k = 1, 2, \ldots, n_1$, and $\hat{y}_P = 0$ yielding $z_P = \sum_{k=1}^{n_1} c_k < z^*$. Now suppose $j = n_1 + 2$ and $i \leq n_1$. Then the partition problem replaces constraint i of (14) and constraint (16) with $x_i + y_P \geq 0$. The optimal solution then has $\hat{x}_i = 0$, $\hat{x}_k = 1$ for $k \neq i$, $\hat{y}_P = 0$, $\hat{y}_{n_1+1} = 1$, and $\hat{y}_k = 0$ for all $k \neq i$, for an objective value of $\sum_{k \neq i} c_k + 1 < z^*$. Next suppose $j = n_1 + 1$ and $i \leq n_1$. Then constraint i of (14) and constraint (15) are replaced with $x_i + y_P \geq 2$. The optimal solution then has $\hat{x}_i = 2$, $\hat{x}_k = 1$ for $k \neq i$, $\hat{y}_P = 0$ and $\hat{y}_k = 0$ for all $k \neq i$, yielding objective value $\sum_{k=1}^{n_1} c_k + c_i < z^*$. Finally, suppose $1 \leq i < j \leq n_1$. Then constraints i and j in (14) are replaced with the constraint $x_i + x_j + y_P \geq 2$. Because $c_i < c_j$, the optimal solution is then $\hat{x}_i = 2$, $\hat{x}_j = 0$, $\hat{x}_k = 0$ for all $k \neq i, j$, $\hat{y}_P = 0$, $\hat{y}_{n_1+1} = 1$, and $\hat{y}_k = 0$ for all $k \neq i, j$, with objective value $\sum_{k=1}^{n_1} c_k - (c_j - c_i) + 1 < z^*$, since $c_j > c_i$.

Proposition 4 shows that the number of components in the constructed completely sufficient partition $\mathcal{N}$ can be small in the case of simple recourse. However, it does not guarantee that the partition-based algorithm will yield such a small partition, nor does it provide a limit on the number of iterations that the partition-based algorithm will take before convergence. [29] gives an example where the partition-based algorithm takes $|N| - 1$ iterations to converge to a completely sufficient partition, and the size of this completely sufficient partition is $|N|$, while there exists a completely sufficient partition of size 3. This undesirable phenomenon of having a large number of iterations is caused by the lack of information about the optimal solution x^* in the intermediate steps. However, according to Proposition 4, after the merging operation, a partition-based master problem with a partition of size at most 3 can be solved at each iteration for this example.

4 Adaptive partition-based approach for expected value constrained programs

In this section, we study how the adaptive partition-based approach can be applied for solving an expected value constrained two-stage stochastic program as follows:

$$\min_{x \in X} \{c^\top x \mid \sum_{k \in N} f_k(x) \leq B\}, \tag{17}$$

where B is a scalar that can be seen as the total available budget. An interesting special case of (17) is the LP relaxation of a chance-constrained LP.

Given a partition $\mathcal{N}$, the corresponding partition-based master problem is:

$$\min\ c^\top x \tag{18a}$$

$$\text{s.t. } \bar{T}^P x + W y^P \geq \bar{h}^P,\ \forall P \in \mathcal{N} \tag{18b}$$

$$\sum_{P \in \mathcal{N}} d^\top y^P \leq B \tag{18c}$$

$$x \in X, y^P \in \mathbb{R}^{n_2}_+,\ \forall P \in \mathcal{N}. \tag{18d}$$

Given a first-stage solution $\hat{x}$, (17) is a pure feasibility problem. We evaluate $f_k(\hat{x})$ separately for each scenario $k \in N$, and then check the violation value of the budget constraint: $v(\hat{x}) := \max\{0, \sum_{k\in N} f_k(\hat{x}) - B\}$. Similar to the definition of ϵ-sufficient partition, we define an ϵ-feasible partition as follows:

Definition 3 *A partition $\mathcal{N}$ is ϵ-feasible to* (17)*, if there exists an optimal solution $\hat{x}$ of the partition-based problem* (18) *with respect to $\mathcal{N}$ that satisfies $v(\hat{x}) \le \epsilon$.*

Given a threshold $\epsilon \ge 0$, we can modify Algorithm 1 to construct an ϵ-feasible partition, by changing the termination criterion into $v(\hat{x}) \le \epsilon$.

We next show that there exists a small completely sufficient partition for expected value constrained programs with simple recourse. An expected value constrained program can be written in extensive form as:

$$\min c^\top x \tag{19a}$$
$$\text{s.t. } T^k x + y_k e \ge h^k,\ \forall k \in N \tag{19b}$$
$$Ax = b,\ x \in \mathbb{R}^{n_1}_+ \tag{19c}$$
$$\sum_{k\in N} y_k \le B \tag{19d}$$
$$0 \le y_k \le u_k,\ \forall k \in N, \tag{19e}$$

where $u_k \ge 0, \forall k \in N$ and $B \ge 0$. Constraints (19e) mean that we may not have relatively complete recourse in this case.

We include the constraints (19e) so that the LP relaxation of a chance-constrained LP with finite scenarios fits this structure, as we now describe. Let $\tilde{T}$ be a random matrix, $\tilde{h}$ be a random vector, and ϵ be a given risk tolerance. Then a chance-constrained LP can be written as:

$$\min c^\top x \tag{20a}$$
$$\text{s.t. } \mathbb{P}(\tilde{T}x \ge \tilde{h}) \ge 1 - \epsilon \tag{20b}$$
$$Ax = b,\ x \in \mathbb{R}^{n_1}_+. \tag{20c}$$

We consider a finite scenario approximation of (20). Suppose a set of scenarios N is given, where each scenario $k \in N$ happens with probability p_k, and the corresponding realization of $\tilde{T}$ and $\tilde{h}$ in that scenario k is $\hat{T}^k$ and $\hat{h}^k$, respectively. Then the chance-constrained program (20) can be formulated as an MIP [4]. Introducing a binary variable z_k for each scenario $k \in N$, the chance-constrained LP (20) can be written as:

$$\min c^\top x \tag{21a}$$
$$\text{s.t. } \hat{T}^k x \ge \hat{h}^k - \hat{M}^k z_k,\ \forall k \in N \tag{21b}$$
$$\sum_{k\in N} p_k z_k \le \epsilon \tag{21c}$$
$$Ax = b,\ x \in \mathbb{R}^{n_1}_+, z \in \{0,1\}^{|N|}, \tag{21d}$$

where $\hat{M}^k \in \mathbb{R}^{m_2}$ is a chosen big-M vector such that when $z_k = 1$, inequalities $\hat{T}^k x \ge \hat{h}^k - \hat{M}^k$ in scenario k do not cut off any feasible solution. We assume that $\hat{M}^k_i > 0, \forall k \in N, i = 1, 2, \ldots, m_2$.

The LP relaxation of (21) is the engine of a branch-and-bound based MIP solver for solving chance-constrained LPs, and can be a bottleneck when the size of the scenario set $|N|$ is large. The partition-based approach can be applied to solve this LP relaxation by reformulating (21) into (19): first re-define variable z_k as $p_k z_k$, then define scenario data $T_i^k := \frac{\hat{T}_i^k}{p_k M_i^k}$, and $h_i^k := \frac{\hat{h}_i^k}{p_k M_i^k}$. Problem (19) can also be seen as a reformulation of a linear program with integrated chance constraints, as pointed out in [33].

We next show that given an optimal solution $\hat{x}$ of the partition-based expected value constrained master problem (18) with $\mathcal{N}$, if the consistency requirement that $\lambda^k = \lambda^{k'}, \forall k, k' \in P, P \in \mathcal{N}$ is satisfied, then partition $\mathcal{N}$ is completely sufficient for the expected value constrained problems (18).

Proposition 5 *Let $\hat{x}$ be an optimal solution of* (18) *with partition $\mathcal{N}$. If there exists a set of optimal solutions $\{\lambda^k\}_{k\in N}$ of* (5) *with $\hat{x}$ that satisfies $\lambda^k = \lambda^{k'}, \forall k, k' \in P, P \in \mathcal{N}$, then $\hat{x}$ is optimal to* (17).

Proof: We just need to show that $\hat{x}$ is feasible to (17). According to Lemma 2 and the assumption, $\sum_{P\in\mathcal{N}} \bar{f}_P(\hat{x}) = \sum_{k\in N} f_k(\hat{x})$. Since $\hat{x}$ is an optimal solution of (18) with partition $\mathcal{N}$, $\sum_{P\in\mathcal{N}} \bar{f}_P(\hat{x}) \le B$, hence $\sum_{k\in N} f_k(\hat{x}) \le B$, i.e., $\hat{x}$ is feasible to (17). □

The partial refinement strategy described in Algorithm 4 cannot be applied directly in the expected value constrained stochastic program, where there is no upper bound to keep track of. In this case, we use the violation value of the constraint in (17), $v(\hat{x}) = \max\{0, \sum_{k\in N} f_k(\hat{x}) - B\}$, as the metric for how feasible a solution $\hat{x}$ is. We then apply this metric as the criterion in Algorithm 4 to determine if $\hat{x}$ is the current best solution.

Similar to the case of two-stage stochastic LPs with simple recourse, the following result shows that there are at most $n_1 - m_1 + m_2 + 1$ distinct vectors in the set $\{(\hat{\lambda}^k, \hat{\gamma}^k), k \in N\}$ for any extreme point optimal solution $(\hat{\lambda}, \hat{\beta}, \hat{y}, \hat{\gamma})$ of the dual of (19):

$$\max b^\top \pi + \sum_{k\in N} (h^k)^\top \lambda^k - B\beta - \sum_{k\in N} u_k \gamma_k \tag{22a}$$

$$\text{s.t. } A^\top \pi + \sum_{k\in N} (T^k)^\top \lambda^k \le c \tag{22b}$$

$$\sum_{i=1}^{m_2} \lambda_i^k - \beta - \gamma_k \le 0, \ \forall k \in N \tag{22c}$$

$$\pi \text{ free}, \ \lambda^k \in \mathbb{R}_+^{m_2}, \ \beta \in \mathbb{R}_+, \ \gamma_k \in \mathbb{R}_+, \ \forall k \in N, \tag{22d}$$

where the dual variables $\lambda^k, \pi, \beta, \gamma_k$ correspond to constraint (19b), (19c), (19d) and (19e), respectively.

Proposition 6 *Let $(\hat{\lambda}, \hat{\beta}, \hat{\pi}, \hat{\gamma})$ be an extreme point optimal solution of* (22). *The number of distinct vectors in the set $\{(\hat{\lambda}^k, \hat{\gamma}^k), k \in N\}$ is at most $n_1 - m_1 + m_2 + 1$.*

Proof: We consider two different cases: $\hat{\beta} = 0$ and $\hat{\beta} > 0$. When $\hat{\beta} = 0$, $\sum_{i=1}^{m_2} \hat{\lambda}_i^k \le \hat{\gamma}_k$, and since

$u_k \geq 0, \hat{\gamma}_k = \sum_{i=1}^{m_2} \hat{\lambda}_i^k$. Problem (22) is then simplified as:

$$\max b^\top \pi + \sum_{k\in N} (h^k - u_k e)^\top \lambda^k \tag{23a}$$

$$\text{s.t. } A^\top \pi + \sum_{k\in N} (T^k)^\top \lambda^k \leq c \tag{23b}$$

$$\pi \text{ free}, \ \lambda^k \in \mathbb{R}_+^{m_2}, \ \forall k \in N. \tag{23c}$$

We consider the number of distinct vectors in the set $\{\hat{\lambda}^k\}_{k\in N}$. For any extreme point solution of formulation (23), the number of binding constraints is no less than the number of variables, which is $m_1 + |N|m_2$. We have at most n_1 binding constraints from (23b), thus at least $|N|m_2 + m_1 - n_1$ constraints are binding for $\lambda_i^k \geq 0, \forall k, i$. Therefore, there are at most $|N|m_2 - (|N|m_2 + m_1 - n_1) = n_1 - m_1$ nonzero values for $\hat{\lambda}_d^k$. The number of distinct vectors in the set $\{\hat{\lambda}^k\}_{k\in N}$ is then at most $n_1 - m_1 + 1$.

When $\hat{\beta} > 0$, let $K_1 := \{k \in N \mid \exists i = 1, 2, \ldots, m_2, \hat{\lambda}_i^k = \hat{\beta} e^i\}$, $K_2 := \{k \in N \mid \hat{\lambda}^k = \mathbf{0}\}$, $K_3 := \{k \in N \mid \exists i : 0 < \hat{\lambda}_i^k < \hat{\beta}$, and $\sum_{i=1}^{m_2} \hat{\lambda}_i^k \leq \hat{\beta}\}$, and $K_4 := \{k \in N \mid \sum_{i=1}^{m_2} \hat{\lambda}_i^k > \hat{\beta}\}$. We now show that $|K_3| + |K_4| \leq n_1 - m_1$.

Because $u_k \geq 0$, we may assume $\hat{\gamma}_k = 0, \forall k \in K_1 \cup K_2 \cup K_3$, and $\hat{\gamma}_k = \sum_{i=1}^{m_2} \hat{\lambda}_i^k - \hat{\beta}, \forall k \in K_4$. At an extreme point optimal solution, the number of binding constraints is no less than the number of variables in (22), which is $m_1 + 1 + |N|(1 + m_2)$. There are at most n_1 binding constraints from (22b), and $\hat{\beta} > 0$, thus there are at least $m_1 + |N|(1 + m_2) - n_1$ binding constraints in the other constraints. Let $T = \{(k, i) \mid k \in K_4, \hat{\lambda}_i^k > 0\}$, and $S(k) = \{i \in \{1, 2, \ldots, m_2\} \mid \hat{\lambda}_i^k > 0\}, \forall k \in K_3$. If $\sum_{i=1}^{m_2} \hat{\lambda}_i^k = \hat{\beta}$, then $|S(k)| \geq 2$, otherwise $|S(k)| \geq 1$. We summarize the possible binding constraints for each set K_1, K_2, K_3 and K_4 as follows:

Index	Binding constraints	Size
$k \in K_1$	$\lambda_{i'}^k = 0, \forall i' \neq i, \gamma = 0, \sum_{i=1}^{m_2} \lambda_i^k - \beta - \gamma_k = 0$	$(m_2 + 1)\|K_1\|$
$k \in K_2$	$\lambda_i^k = 0, \forall i = 1, 2, \ldots, m_2, \gamma_k = 0$	$(m_2 + 1)\|K_2\|$
$k \in K_3$	$\lambda_i^k = 0, \forall i \notin S(k), \gamma_k = 0$, if $\sum_{i=1}^{m_2} \hat{\lambda}_i^k = \hat{\beta}$, $\|S(k)\| \geq 2$	$\leq m_2\|K_3\|$
$k \in K_4$	$\lambda_i^k = 0, \forall (k, i) \notin T, \sum_{i=1}^{m_2} \lambda_i^k - \beta - \gamma_k = 0$	$\leq m_2\|K_4\|$

We need at least $m_1 + |N|(1 + m_2) - n_1$ binding constraints, and so

$$|K_1|(m_2 + 1) + |K_2|(m_2 + 1) + m_2|K_3| + m_2|K_4| \geq m_1 + |N|(1 + m_2) - n_1.$$

Since $|K_1| + |K_2| + |K_3| + |K_4| = |N|$, we have $|K_3| + |K_4| \leq n_1 - m_1$. Therefore, when $\hat{\beta} > 0$, the number of distinct $(\hat{\lambda}, \hat{\gamma})$ vectors is at most $n_1 - m_1 + m_2 + 1$. □

According to Proposition 2, a completely sufficient partition can be constructed by grouping scenarios that correspond to the same $(\hat{\lambda}^k, \hat{\gamma}^k)$ values, which has no more than $n_1 - m_1 + m_2 + 1$ components. Again, the size of this partition is independent of the number of scenarios $|N|$ in the model.

5 Computational experiments

We conduct computational experiments on the proposed partition-based approach for solving two-stage stochastic LPs with simple recourse, more general two-stage stochastic LPs with random right-hand side vectors, and LP relaxations of chance-constrained LPs.

5.1 Implementation Details

We implement all algorithms within the commercial MIP solver IBM Ilog CPLEX, version 12.4. We turn off the CPLEX Presolve and set the number of threads to one. When the number of scenarios is huge, e.g., $|N| > 10000$, we found that the computational effort for doing some necessary linear algebra operations became a bottleneck. We therefore use a numerical linear algebra library Eigen [14] for these operations. All tests are conducted on a Linux workstation with four 3.00GHz processors and 8Gb memory.

We report the average results over five replications for each instance and sample size. We use the following abbreviations throughout this section:

1. AvT: Average solution time.
2. AvI: Average number of iterations.
3. AvS: Average partition size.
4. AvC: Average number of Benders cuts added.

For all our tests on stochastic LPs with simple recourse, we use a time limit of 1200 seconds, and for general two-stage stochastic LPs with random right-hand side vectors, we use a time limit of 10800 seconds. We use "$>$" to denote the case when not all replications are solved within the time limit, since in this case we calculate the average time by using the time limit for replications that exceed the limit. We use "-" to denote the case when none of the replications are solved within the time limit.

We compare the performance of the proposed adaptive partition-based approach with the extensive formulation, two variants of Benders decomposition (multi-cut and single-cut), and the level method [13, 19]. For each class of instances, we found that one of the Benders variants dominated the other, and so when reporting results we only report results for the better of the two.

In our implementation of multi-cut Benders decomposition, we solve the Benders master problem (9) using dual simplex method by CPLEX. We add a Benders cut (9b) when the relaxation solution $(\hat{\theta}, \hat{x})$ violates the cut by more than a violation threshold. We set this threshold to be $\max\{1, |\hat{\theta}|\} \times 10^{-5}$ for instances on two-stage stochastic LPs with simple recourse, and we set it to be $\max\{1, |\hat{\theta}|\} \times 10^{-4}$ for general two-stage stochastic LP instances with random right-hand side vectors. We used the same settings for the single-cut variant of Benders decomposition, with the difference being in the master problem (only a single variable is used, and the aggregated single cuts are added). For our final set of test instances, the LP relaxation of chance-constrained LPs, we apply a specialized Benders decomposition, the projection cut formulation [30], which is similar to a specialization of the single-cut implementation of Benders to this problem class.

Our implementation of the level method is based on using aggregate cuts as in the single-cut version of Benders decomposition. The starting iterate solution is obtained by solving the mean-value problem. For two-stage stochastic LPs, we set the level parameter to $\lambda = 0.5$ (see, e.g., [13, 19]). For expected value constrained programs, we follow the implementation of the constrained level method proposed by [13]. We use parameters $\mu = 0.5$ and $\lambda = 0.5$ in the algorithm described in Section 3 of [13].

For two-stage stochastic LPs, for all methods that we test on, we terminate when the relative optimality gap is less than 10^{-4}. We calculate relative optimality gap as $(UB - LB)/UB < 10^{-4}$, where UB and LB are the best upper bound and lower bound obtained by the algorithm, respectively. For expected value constrained programs, we set the stopping criterion for both the partition-based approaches and the Benders formulation as $v(\hat{x}) < |N| \times 10^{-4}$, where $v(\hat{x})$ is the feasibility metric introduced in Section 4. We set the convergence threshold as $\epsilon = 10^{-4}$ for the constrained level method.

5.2 Two-stage stochastic LPs with simple recourse

We generate instances on two-stage stochastic programs with simple recourse based on deterministic instances from [3], including multi-dimensional knapsack instances cb7-1 and cb8-1 (we denote them as p1 and p2 for simplicity). Instances p1 have $n_1 = 100$ decision variables and instances p2 have $n_1 = 250$ decision variables. Both p1 and p2 have no first-stage constraints and $m_2 = 30$ second-stage constraints.

We randomly generate scenarios in the following way. First, each variable j fails to appear according to a Bernoulli distribution with the appearance probabilities equal to some random generated parameters $\mu_j, \forall j = 1, 2, ..., n$. These appearance probabilities μ_j are generated according to an exponential distribution with mean 0.1, and then truncated to be between 0 and 1. A variable that does not appear in a scenario has a zero coefficient in all the constraints for that scenario. If a variable appears, then its weight in each row is normally distributed with mean equal to its weight in the deterministic instance and standard deviation equal to 0.2 times the mean. Given this distribution, we take independent samples of different sizes. In our tests, we use two penalty coefficients, 0.01 and 0.002, denoted as "H" and "L" respectively.

Table 1 compares the results of the three different refinement options for the instances of two-stage stochastic LPs with simple recourse. We find that the No-Merge option yields the smallest number of iterations, but the partition size is the largest; Merge-All yields the largest number of iterations, but the partition size is the smallest. Merge-Partial takes slightly fewer iterations, and has slightly larger partition size than Merge-All. In most cases, Merge-Partial has the best performance. Thus, for these instances, it is a better idea to perform the merging operation only when $\hat{x}$ is the current best solution. We also see that as the number of scenarios increases, the average partition size does not increase much, which is expected according to Proposition 4. However, the number of iterations increases slightly with the number of scenarios.

Table 1: Two-stage stochastic LPs with simple recourse instances: average solution time, number of iterations and partition size, for three different partition refinement strategies.

Instances		No-Merge			Merge-All			Merge-Partial		
Ins	$\lvert N\rvert$	AvT	AvI	AvS	AvT	AvI	AvS	AvT	AvI	AvS
p1-H	5k	23.6	8	1278	3.9	19	139	4.3	21	139
	10k	46.7	9	2034	4.7	24	140	4.9	24	147
	20k	99.6	9	3438	5.8	26	145	5.9	25	153
p2-H	5k	39.1	9	1018	12.3	18	157	12.7	18	160
	10k	75.0	10	1543	14.8	21	162	15.4	22	168
	20k	135.2	10	2370	18.8	24	171	18.5	24	178
p1-L	5k	46.8	6	1429	11.2	22	200	11.9	21	215
	10k	100.9	7	2373	11.7	28	195	12.1	26	209
	20k	207.2	8	3803	13.3	32	195	14.0	30	214
p2-L	5k	78.3	7	1118	26.8	18	216	29.3	18	230
	10k	120.1	7	1645	29.2	23	216	30.1	22	234
	20k	240.8	9	2596	38.1	32	208	36.3	27	234

Table 2 compares the results of the extensive formulation (Ext), the multi-cut Benders decomposition algorithm (Multi-Benders), the level method (Level), and the best option of the partition-based approach (Merge-Partial). (For these instances, the multi-cut Benders decomposition con-

Table 2: Two-stage stochastic LPs with simple recourse instances: average time for the extended formulation; average time and number of iterations for the multi-cut Benders method; average time and number of iterations for the level method; and average time, number of iterations, and partition size for the best partition option Merge-Partial.

Instances		Ext	Multi-Benders			Level		Merge-Partial		
Ins	$\|N\|$	AvT	AvT	AvI	AvC	AvT	AvI	AvT	AvI	AvS
p1-H	5k	11.9	3.4	7	4278	2.9	89	4.3	21	139
	10k	25.4	4.7	7	4688	5.4	93	4.9	24	147
	20k	49.7	5.6	6	4840	10.5	104	5.9	25	153
p2-H	5k	26.2	7.6	7	4693	7.4	87	12.7	18	160
	10k	58.3	12.0	8	5588	13.7	96	15.4	22	168
	20k	121.6	16.6	8	5804	25.7	100	18.5	24	178
p1-L	5k	19.0	26.1	9	7171	2.9	63	11.9	21	215
	10k	44.6	41.0	10	8518	5.1	69	12.1	26	209
	20k	92.5	48.9	10	9136	9.6	75	14.0	30	214
p2-L	5k	42.9	44.4	9	7473	6.5	54	29.3	18	230
	10k	99.5	69.1	9	8227	11.8	63	30.1	22	234
	20k	214.2	106.3	11	9700	22.2	66	36.3	27	234

sistently outperformed the single-cut implementation.) We see from Table 2 that the extensive formulation takes longer to solve than the other three options. Multi-cut Benders works relatively well on instances with a larger penalty coefficient, in the sense that both the solution time and the number of Benders cuts do not increase much as the scenario size increases. The adaptive partition-based approach is competitive with the level method. In particular, for instances with a larger penalty coefficient, Merge-Partial outperforms the level method. For instances with a smaller penalty coefficient, Merge-Partial takes slightly more time than the level method. When the penalty coefficient is smaller, the number of iterations for the level method decreases and therefore the computational time is reduced. On the other hand, although the number of iterations by option Merge-Partial is not significantly increased, the sizes of partition significantly increase, which leads to more computational time.

5.3 General two-stage stochastic LPs with random right-hand side vectors

We next present computational results for general two-stage stochastic programming instances with random right-hand side vectors. These instances do not have simple recourse structure, and so there is no known theoretical guarantee that the proposed partition-based approach can yield a small sufficient partition. However, we are still interested in investigating the performance of this approach on this class of instances. Our test instances are taken from [1] and [20]. We generate samples with different sample sizes in our experiments, following the probability distributions specified in these instances. The sizes of these instances are described in Table 3.

Table 4 compares the results of the extensive formulation (Ext), the better between the two Benders variants (Best-Benders), the level method (Level) and the partition-based approach with the Merge-Partial option described above. The single-cut variant of Benders was better than the multi-cut variant for all of these instances except for ssn. We see from Table 4 that, the Merge-Partial partition-based approach yields the best performance among all the methods on the cargo, gbd and LandS instances. For the stormG2 instances the partition-based approach significantly outperforms the Ext and the best Benders option and performs similarly to the level method. For the ssn instances, the multi-cut Benders method is best, and the partition-based and level

Table 3: Description of the test instances from [1] and [20]. (n, m) means that the number of variables is n, and the number of constraints is m.

Instance	Original scenario size	First-stage size	Second-stage size
stormG2	6×10^{81}	(185,121)	(528,1259)
ssn	10^{70}	(1,89)	(175,706)
cargo	8192, 16384, 32768	(16,52)	(74,186)
gbd	684450	(4, 17)	(5, 10)
LandS	10^{6}	(2, 4)	(7, 12)

Table 4: Two-stage stochastic programs with random right-hand side vectors: average time for the extensive formulation; average time, number of iterations and number of cuts generated for the better version of Benders decomposition between single-cut and multi-cut; average time, number of iterations and number of cuts generated for the level method; and the average time, number of iterations, and partition size for the partition strategy Merge-Partial. Multi-cut Benders is reported for ssn, the single-cut Benders option is reported for all other instances.

Instances		Ext	Best-Benders			Level		Merge-Partial		
Instance	$\lvert N\rvert$	AvT	AvT	AvI	AvC	AvT	AvI	AvT	AvI	AvS
stormG2	1k	61.2	276.4	84	54k	62.8	22	72.9	3	336
	5k	514.5	1350.8	85	270k	348.6	24	379.3	3	1163
	10k	1363.9	2838.8	87	540k	703.1	24	764.7	3	1915
ssn	1k	57.5	99.3	17	9028	803.6	181	110.1	5	381
	2k	147.3	190.5	15	16k	1854.0	207	261.0	5	688
	5k	4312.2	478.5	14	34k	5335.9	239	5763.5	7	1501
cargo†	8k	43.5	431.6	250	250	67.2	40	10.0	3	344
	16k	-	749.5	219	219	155.5	48	14.1	3	265
	32k	-	1254.7	204	204	282.8	48	23.0	3	269
gbd	20k	12.3	32.2	27	27	27.6	22	5.0	5	135
	50k	110.0	79.5	27	27	71.1	23	12.3	5	142
	100k	139.2	164.5	27	27	144.4	23	24.7	5	143
LandS	20k	13.5	20.9	19	19	9.6	9	4.7	5	41
	50k	119.6	54.8	20	20	26.0	10	11.6	5	42
	100k	190.3	110.8	20	20	53.1	10	23.5	5	41

'†': Just one instance for each scenario size.

methods again perform similarly. For all of these instances, the partition-based approach yields a small number of iterations, and a small partition size. However, the performance of the partition-based approach is relatively poor on instance ssn. In that case, although the final partition size is not very large (due to merging), we observe that the partition sizes in the first few iterations are close to the entire scenario size $|N|$, which leads to a long computational time. The computational performance of the level method is also poor on the ssn instances because of a large number of iterations, and most of the time is spent on iteratively solving the subproblems. The existence of some instances (cargo, gbd, and LandS) where the partition-based approach yields the best performance motivates further study of specific structures of two-stage stochastic programs where the adaptive partition-based approach may work well.

5.4 The LP relaxation of chance-constrained LPs

We next compare performance of the methods for solving the continuous relaxation of chance-constrained LP instances. We generate stochastic covering LP instances c1 and c2 in the way suggested in [23]: each constraint coefficient a_j of a variable x_j is generated uniformly between 0.8 and 1.5, then the coefficients are divided by 1.1, and the right hand-side value is 1 in all scenarios. Instances c1 have $n_1 = 50$ decision variables and instances c2 have $n_1 = 100$ decision variables. Both c1 and c2 have a single inequality in the chance constraint. We use two different risk parameters $\epsilon = 0.05$, and $\epsilon = 0.01$, and denote them as "H" and "L" respectively.

Table 5: The LP relaxation of chance-constrained LPs with single-row covering instances: average solution time, number of iterations and partition size, for three different partition refinement strategies.

Instances		No-Merge			Merge-All			Merge-Partial		
Ins	$\lvert N\rvert$	AvT	AvI	AvS	AvT	AvI	AvS	AvT	AvI	AvS
c1-H	10k	2.8	13	369	0.8	246	31	0.6	137	47
	20k	5.9	13	485	1.3	328	36	1.0	169	52
	50k	26.6	14	914	3.5	497	40	2.1	230	63
c2-H	10k	3.8	12	335	0.7	152	36	0.8	133	50
	20k	8.7	13	452	1.4	227	40	1.6	181	57
	50k	43.1	14	866	4.2	364	44	3.3	206	68
c1-L	10k	9.1	14	770	5.7	968	51	1.9	181	84
	20k	30.5	15	1335	11.1	1557	53	2.9	226	94
	50k	171.8	16	2671	33.3	2955	55	6.2	311	107
c2-L	10k	27.3	14	821	18.3	797	92	9.1	205	133
	20k	107.4	15	1460	32.3	1207	95	13.5	255	145
	50k	547.1	16	2848	87.1	2169	10	24.8	303	173

Table 5 compares the results of the three different refinement options for our instances of the LP relaxation of chance-constrained LPs. We find that the behavior of these refinement strategies in this case is similar to the two-stage stochastic LPs with simple recourse. The No-Merge strategy yields much larger partition sizes, especially when we use a smaller risk parameter. Although it yields a smaller number of iterations, the No-Merge strategy is not competitive with the alternative strategies in terms of solution time. Comparing the two merging strategies, we see that Merge-Partial yields significantly better results than Merge-All in the single-row covering instances with a smaller risk parameter. The reason is that Merge-All requires many more iterations, while the partition size is similar for both options. Again, this shows the advantage of performing the merging operation based on the current best solution.

Table 6 compares the results of the extensive formulation (Ext), the specialized version of the single-cut Benders algorithm (Single-Benders), the constrained level method (CLM), and the best option of the partition-based approach (Merge-Partial). We see from Table 6 that the extensive formulation exceeds the time limit for all instances when the number of scenarios $|N|$ is large. The adaptive partition-based approach outperforms Single-Benders, especially when a smaller risk parameter is used. In these cases, a large number of Benders cuts are generated that slow down the solver (the number of iterations is the same as the number of Benders cuts generated, since we add the single most violated cut in each iteration). However, the increase in the number of iterations and the partition size does not lead to significant increase in solution time for the partition-based approach. We also see that the partition-based approach is competitive with the constrained level method on many of the instances. However, the constrained level method has a better performance for instances with a smaller risk parameter, due to a small number of iterations. This motivates further exploration on more sophisticated implementations of the proposed partition-based approach, for example, using regularization techniques.

Table 6: The LP relaxation for chance-constrained LPs with single-row covering instances: average time for the extensive formulation; average time and number of iterations for the specialized single-cut version of Benders decomposition, the projection cut formulation; average time and number of iterations for the constrained level method; and average time, number of iterations and partition size for the partition strategy Merge-Partial.

Instances		Ext	Single-Benders		CLM		Merge-Partial		
Ins	$\lvert N\rvert$	AvT	AvT	AvI	AvT	AvI	AvT	AvI	AvS
c1-H	10k	64.5	2.3	588	0.6	74	0.6	137	47
	20k	431.7	4.9	804	1.0	92	1.0	169	52
	50k	-	12.5	1043	1.2	82	2.1	230	63
c2-H	10k	145.6	7.9	884	1.1	96	0.8	133	50
	20k	>1170.3	16.2	1136	1.8	103	1.6	181	57
	50k	-	47.1	1625	2.5	97	3.3	206	68
c1-L	10k	33.0	5.9	948	0.8	74	1.9	181	84
	20k	171.1	7.8	966	0.5	68	2.9	226	94
	50k	-	9.8	1014	0.8	68	6.2	311	107
c2-L	10k	97.6	158.1	2888	1.6	115	9.1	205	133
	20k	689.4	184.0	3045	0.9	79	13.5	255	145
	50k	-	238.4	3319	2.0	93	24.8	303	173

6 Concluding remarks

We study an adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. We propose a general solution framework that converges in a finite number of iterations to a sufficient partition. A solution guided refinement strategy is developed to refine the partition. When the feasible set is polyhedral, we can further take advantage of the dual optimal solutions to put some components in a partition back together, without weakening the corresponding relaxation bound. For two-stage stochastic LPs and expected value constrained LPs with simple recourse, we show that there exists a small completely sufficient partition. The size of this particular partition is independent of the number of scenarios used in the model. Our preliminary computational results show that the proposed adaptive partition-based approach is competitive with the Benders decomposition and the level method in two-stage stochastic LPs with simple recourse, and empirically converges to a sufficient partition of a small size very fast. We also found that the proposed partition-based approach is competitive for two-stage stochastic programs with fixed, but not simple, recourse, even though our theory does not guarantee existence of a small sufficient partition for these instances. This motivates further investigation on specific structures where a small sufficient partition can be obtained.

There are several directions that can be explored to further enhance the adaptive partition-based approach. First, the partition-based approach can be integrated with decomposition approaches. For example, we may use Benders decomposition to solve the partition-based master problem in each step. Second, warm starting schemes may help when solving the partition-based master problem iteratively. Third, partition-based approach can be combined with inexact bundles. This may help solving two-stage stochastic programs in which exactly solving the second-stage problems is computationally demanding. Finally, it would be interesting to explore the use of regularization techniques with the partition-based problem as the master problem.

The solution framework can be extended to cases where the first stage feasible set X is convex but not necessarily polyhedral. In particular, we expect the small partition property for two-stage stochastic programs with simple recourse to hold for such problems under mild technical assumptions. The adaptive partition-based framework also extends to stochastic integer programs with integer variables only in the first stage. Although its performance is under further study, a limitation of the adaptive partition-based approach is that, aggregating coefficients may destroy structure that appears in the scenario-based formulation. For example, in a set packing problem, the constraint coefficients are all 0's and 1's. MIP solvers can take advantage of this structure and improve the problem formulation by generating valid inequalities that improve the LP relaxation. When this structure is destroyed in the partition-based master problem, the partition-based master problem may be much harder to solve than the scenario-based problems, despite being mroe compact.

Acknowledgments

This research has been supported in part by the National Science Foundation under grant CMMI-0952907 and by the Applied Mathematics activity, Advance Scientific Computing Research program within the Department of Energy Office of Science under a contract from Argonne National Laboratory.

References

[1] K.A. Ariyawansa and A.J. Felt, *On a new collection of stochastic linear programming test problems*, INFORMS J. Comput., 16 (2004), pp. 291–299.

[2] J.C. Bean, J. Birge, and R.L. Smith, *Aggregation in dynamic programming*, Oper. Res., 35 (1987), pp. 215–220.

[3] J.E. Beasley, *OR-library: Distributing test problems by electronic mail*, Journal of the Operational Research Society, 41 (1990), pp. 1069–1072.

[4] P. Beraldi and A. Ruszczyński, *A branch and bound method for stochastic integer programs under probabilistic constraints*, Optim. Methods Softw., 17 (2002), pp. 359–382.

[5] D. Bienstock and M. Zuckerberg, *Solving lp relaxations of large-scale precedence constrained problems*, IPCO, (2010), pp. 1–14.

[6] J. Birge, *Aggregation bounds in stochastic linear programming*, Math. Program., 31 (1985), pp. 25–41.

[7] J.R. Birge and F.V. Louveaux, *Introduction to stochastic programming, 2nd edition*, Springer, 2011.

[8] M.S. Casey and S. Sen, *The scenario generation algorithm for multistage stochastic linear programming*, Math. Oper. Res., 30 (2005), pp. 615–631.

[9] K.J. Cormican, D.P. Morton, and R.K. Wood, *Stochastic network interdiction*, Oper. Res., 46 (1998), pp. 184–197.

[10] B. Denton and D. Gupta, *A sequential bounding approach for optimal appointment scheduling*, IIE Transactions, 35 (2003), pp. 1003 – 1016.

[11] N.C.P. Edirisinghe and W.T. Ziembda, *Tight bounds for stochastic convex programs*, Oper. Res., 40 (1992), pp. 660–677.

[12] D. ESPINOZA AND E. MORENO, *A primal-dual aggregation algorithm for minimizing conditional-value-at-risk in linear programs*, Computational Optimization and Applications, 59 (2014), pp. 617–638.

[13] C. FÁBIÁN AND Z. SZÖKE, *Solving two-stage stochastic programming problems with level decomposition*, Computational Management Science, 4 (2007), pp. 313–353.

[14] G. GUENNEBAUD, B. JACOB, ET AL., *Eigen v3*. http://eigen.tuxfamily.org, 2010.

[15] A. HALLEFJORD AND S. STOROY, *Aggregation and disaggregation in integer programming problems*, Oper. Res., 38 (1990), pp. 619–623.

[16] D.B. HAUSCH AND W.T. ZIEMBA, *Bounds on the value of information in uncertain decision problems: two*, Stochastics, 10 (1983), pp. 181–217.

[17] C.C. HUANG, W.T. ZIEMBA, AND A. BEN-TAL, *Bounds on the expectation of a convex function of a random variable*, Oper. Res., 25 (1977), pp. 315–325.

[18] K. JORNSTEN, R. LEISTEN, AND S. STOROY, *Convergence aspects of adaptive clustering in variable aggregation*, Computers & Operations Research, 26 (1999), pp. 955–966.

[19] C. LEMARÉCHAL, A. NEMIROVSKII, AND Y. NESTEROV, *New variants of bundle methods*, Math. Program., 69 (1995), pp. 111–147.

[20] J. LINDEROTH, A. SHAPIRO, AND S. WRIGHT, *The empirical behavior of sampling methods for stochastic programming*, Ann. Oper. Res., 142 (2006), pp. 215–241.

[21] W. OLIVEIRA, C. SAGASTIZÁBAL, AND S. SCHEIMBERG, *Inexact bundle methods for two-stage stochastic programming*, SIAM J. Optim., 21 (2011), pp. 517–544.

[22] P. PIERRE-LOUIS, G. BAYRAKSAN, AND D.P. MORTON, *A combined deterministic and sampling-based sequential bounding method for stochastic programming*, in Winter Simulation Conference, S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds., WSC, 2011, pp. 4172–4183.

[23] F. QIU, S. AHMED, S.S. DEY, AND L.A. WOLSEY, *Covering linear programming with violations*, INFORMS Journal on Computing.

[24] D.F. ROGERS, R.D. PLANTE, R.T. WONG, AND J.R. EVANS, *Aggregation and disaggregation techniques and methodology in optimization*, Oper. Res., 39 (1991), pp. 553–582.

[25] C.H. ROSA AND S. TAKRITI, *Improving aggregation bounds for two-stage stochastic programs*, Oper. Res. Lett., 24 (1999), pp. 127–137.

[26] A. RUSZCZYŃSKI, *A regularized decomposition method for minimizing a sum of polyhedral functions*, Math. Program., 35 (1986), pp. 309–333.

[27] A. RUSZCZYŃSKI AND A. ŚWIĘTANOWSKI, *Accelerating the regularized decomposition method for two stage stochastic linear problems*, European J. Oper. Res., 101 (1997), pp. 328 – 342.

[28] B. SANDIKÇI, N. KONG, AND A.J. SCHAEFER, *A hierarchy of bounds for stochastic mixed-integer programs*, Math. Program., 138 (2013), pp. 253–272.

[29] Y. SONG, *Structure-exploiting algorithms for chance-constrained and integer stochastic programs*, PhD thesis, University of Wisconsin-Madison, 2013.

[30] Y. Song, J. Luedtke, and S. Küçükyavuz, *Chance-constrained binary packing problems*, (2014). To appear in INFORMS Journal on Computing.

[31] S. Trukhanov, L. Ntaimo, and A. Schaefer, *Adaptive multicut aggregation for two-stage stochastic linear programs with recourse*, European J. Oper. Res., 206 (2010), pp. 395 – 406.

[32] R. Van Slyke and R. J.-B. Wets, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math, 17 (1969), pp. 638–663.

[33] C. Wolf and A. Koberstein, *Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested l-shaped method*, European J. Oper. Res., 230 (2013), pp. 143 – 156.

[34] S. E. Wright, *Primal-dual aggregation and disaggregation for stochastic linear programs*, Math. Oper. Res., 19 (1994), pp. 893–908.

[35] P.H. Zipkin, *Bounds for row-aggregation in linear programming*, Oper. Res., 28 (1980), pp. 903–916.

[36] ——, *Bounds on the effect of aggregating variables in linear programming*, Oper. Res., 28 (1980), pp. 403–418.