Sample Average Approximation for Stochastic Nonconvex Mixed Integer Nonlinear Programming via Outer Approximation

Can Li[1], David E. Bernal[1], Kevin C. Furman[2], and Ignacio E. Grossmann[1]

[1]Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA.
{canl1,bernalde,grossmann}@cmu.edu
[2]ExxonMobil Upstream Research Company, 22777 Springwoods Village Parkway, Spring, TX 77389
kevin.c.furman@exxonmobil.com

December 12, 2019

Abstract

Stochastic mixed-integer nonlinear programming (MINLP) is a very challenging type of problem. Although there have been recent advances in developing decomposition algorithms to solve stochastic MINLPs [5, 9, 18, 20], none of the existing algorithms can address stochastic MINLPs with continuous distributions. We propose a sample average approximation-based outer approximation algorithm (SAAOA) that can address nonconvex two-stage stochastic programs (SP) with any continuous or discrete probability distributions. The SAAOA algorithm does internal sampling within a nonconvex outer approximation algorithm where we iterate between a mixed-integer linear programming (MILP) master problem and a nonconvex nonlinear programming (NLP) subproblem. We prove that the optimal solutions and optimal value obtained by the SAAOA algorithm converge to the optimal solutions and the optimal value of the true SP problem as the sample size goes to infinity. The convergence rate is also given to estimate the sample size. However, the theoretical sample size estimate is too conservative in practice. Therefore, we propose an SAAOA algorithm with confidence intervals for the upper bound and the lower bound at each iteration of the SAAOA algorithm. Two policies are proposed to update the sample sizes dynamically within the SAAOA algorithm with confidence intervals. The proposed algorithm works well for the special case of pure binary first stage variables and continuous stage two variables since in this case the nonconvex NLPs can be solved for each scenario independently. The proposed algorithm is tested with a stochastic pooling problem and is shown to outperform the external sampling approach where large scale MINLPs need to be solved. Stochastic programming Sample Average Approximation Mixed-integer Nonlinear Programming Outer Approximation

1 Introduction

Mixed-integer nonlinear programming (MINLP) is a framework to model optimization problems that involve discrete and continuous variables and nonlinear constraints. Many applications can be modeled with MINLP, such as the pooling problem [25], batch plant design [17], and water network synthesis [10]. Although there have been significant advances to solve deterministic MINLPs [15], fewer works have been proposed to solve MINLP problems under uncertainty.

Two-stage stochastic programming (SP) is a framework to model decision-making problems under uncertainty [30]. Specifically, stage 1 decisions are made 'here and now' and are then followed by the resolution of uncertainty. Stage 2 decisions, or recourse decisions, are taken 'wait and see' as corrective actions. The objective of SP is to optimize the expected value of an objective function over a known probability distribution. The probability distribution is usually assumed to be discrete so that the two-stage SP problem can be modeled using a scenario-based approach.

Decomposition algorithms, such as Benders decomposition [4], are used to solve SP problems. Recently, some advances in decomposition algorithms for scenario-based two-stage stochastic MINLP problems have been proposed. For convex stochastic MINLP, where the nonlinear feasible region of the continuous relaxation is convex, Mijangos [24] proposes an algorithm based on Branch-and-Fix Coordination method [2] for convex problems with mixed-binary variables in the first stage and only continuous variables in the second stage. Atakan and Sen [3] propose a progressive hedging-based branch-and-bound algorithm for convex stochastic MINLP. An improved L-shaped method where the Benders subproblems are convexified by rank-one lift-and-project, and Lagrangean cuts are added to tighten the Benders master problem is proposed by Li and Grossmann [17]. Li and Grossmann further propose a generalized Benders decomposition-based branch and bound algorithm [16] with finite ϵ-convergence for convex stochastic MINLPs with mixed-binary first and second stage variables.

For nonconvex stochastic MINLP, where the nonlinear functions in the stochastic MINLPs can be nonconvex, Li et al. [20] propose a nonconvex generalized Benders decomposition algorithm which can solve two-stage nonconvex MINLPs with pure binary variables in a finite number of iterations. For the more general case where the first stage variables can be mixed-integer, Ogbe and Li [27] propose a joint decomposition algorithm. A perfect information-based branch and bound algorithm that solves non-separable nonconvex stochastic MINLPs to global optimality is proposed by Cao and Zavala [5]. Kannan and Barton [9] propose a modified Lagrangean relaxation-based (MLR) branch and bound algorithm, and they prove that MLR has finite ϵ-convergence. A generalized Benders decomposition-based branch and cut algorithm for nonconvex stochastic MINLPs with mixed-binary first and second stage variables was proposed by Li and Grossmann [18].

All the decomposition algorithms mentioned above for stochastic MINLPs assume that the SP is scenario-based, i.e., the probability distribution is discrete. For SP with continuous distributions, the 'true' problem can be solved using sample average approximation (SAA) [14, 30]. SAA approximates the 'true' value of SP with the sample average of N i.i.d samples. There are two classes of sample average methods: 'external sampling' and 'internal sampling'.

'External sampling' means that the sampling is performed external to (prior to) the solution procedure. Mak et al. [23] prove bounding properties for external sampling. Kleywegt et al. [14] study a Monte Carlo simulation-based approach for SP with only discrete variables in the first stage. Linderoth et al. [22] computationally study the quality of solutions obtained from sample-average approximations to two-stage stochastic linear programs.

'Internal sampling' means that the sampling is done within the solution procedure. Norkin et al. [26] propose a stochastic branch and bound method for solving stochastic global optimization problems. Higle and Sen [8] solve two-stage stochastic linear programs with stochastic decomposition, where they construct statistical estimates of supports of the objective function using Benders-like cutting planes. Wei and Realff [32] perform Monte Carlo sampling within the outer approximation (OA) algorithm [6] to solve stochastic convex MINLPs.

In this paper, we address two-stage stochastic nonconvex MINLPs with any probability distribution $\mathbb{P}$ using SAA. The sampling is performed in an 'internal sampling' manner within a nonconvex OA procedure similar to the one proposed by Kesavan et al. [11]. The contributions of this paper are outlined as follows,

1. We propose an internal sampling-based outer approximation (OA) algorithm for two-stage stochastic nonconvex MINLP with continuous distribution.

2. We prove that the proposed algorithm converges to the 'true' value and optimal solution of the stochastic program (SP) as the sample sizes in all the internal sampling steps go to infinity.

3. We provide sample size estimates and convergence rates for the proposed algorithm.

4. We propose an efficient way to implement the sample average-based OA algorithm using confidence interval estimates.

5. We propose two update policies for updating the sample size within SAAOA.

6. Case studies for a stochastic pooling problem are used to demonstrate different update policies for the sample sizes in the OA algorithm.

In section 2, we define the problem addressed in this paper. Section 3 presents an overview of the SAAOA algorithm and defines the subproblems in each step of the OA procedure. We then provide the steps of the SAAOA algorithm. In section 4, we prove that the proposed algorithm converges as the sample size goes to infinity and find its convergence rate. Section 5 describes an efficient implementation of the algorithm using confidence interval estimates. In section 6, we demonstrate the effectiveness of the proposed algorithm with a stochastic pooling problem. We draw the conclusion in section 7.

2 Problem Statement

The problem that we address in this paper is defined in Eq. (1),

$$\text{(SP-MINLP)} \quad \min_{x\in X, y\in\{0,1\}^m} \mathop{\mathbb{E}}_{\theta\sim\mathbb{P}} \min_{z\in Z} f\left(x,y,z,\theta\right) \tag{1a}$$

$$\text{s.t.} \quad g(x,z,y,\theta) \leq 0 \quad \forall\theta\in\Theta \tag{1b}$$

where, x and y are vectors of continuous and discrete decision variables, respectively. Notice that any bounded general integer variable can be modeled using binary variables. z is a vector of continuous control variables, which can vary depending on the realization of uncertain parameter θ. The uncertain parameter $\theta \in \Theta$ follows a probability distribution $\mathbb{P}$, which can be continuous. Functions f and g are smooth functions, which can be nonconvex and sets X and Z are compact. Problem (1) is a two-stage SP where variables x and y represent the first stage decisions, variable z represents the second stage decisions. The expectation is taken over the probability distribution of uncertainty parameter θ. We make the following assumption about problem (SP-MINLP).

Assumption 1 *Problem (SP-MINLP) has relatively complete recourse, i.e., any solution (x, y) that satisfies the first stage constraints has feasible recourse decisions in the second stage.*

Solving (SP-MINLP) with continuous probability distribution directly involves integrating over the distribution, which is usually computationally intractable. Instead of minimizing the 'true' expectation, one can generate N i.i.d. samples for the uncertain parameter θ and minimize the empirical risk. The empirical risk minimization problem described in Eq. (2) is called sample average approximation (SAA) [30] in SP literature.

$$\text{(SAA-MINLP)} \quad \min_{x\in X, z_i\in Z, y\in\{0,1\}^m} \frac{1}{N}\sum_{i=1}^{N} f\left(x,y,z_i,\theta_i\right) \tag{2a}$$

$$\text{s.t.} \quad g(x, y, z_i, \theta_i) \leq 0 \quad \forall i \in [N] \tag{2b}$$

where $[N]$ represents the set $\{1, 2, \cdots, N\}$, θ_i, $i \in [N]$, are N i.i.d. samples of uncertain parameter, z_i is the stage 2 variable corresponding to θ_i. One option to solve (SP-MINLP) is to approximate it with (SAA-MINLP), which can be regarded as 'external sampling'. The convergence properties of (SAA-MINLP) have been studied widely [30]. In this paper, we take an internal sampling approach to solve (SP-MINLP) by using a nonconvex OA algorithm.

3 Sampling Average Approximation within the Outer Approximation Algorithm (SAAOA)

We first give a high-level overview of the SAAOA algorithm before we go to the details. Let us take a step back and consider a nonconvex OA algorithm to solve deterministic MINLP. We have an MILP master problem where the nonconvex functions in the original MINLP are replaced by their polyhedral relaxations. The MILP master problem provides a lower bound (LB) of the original MINLPs optimal objective function value. After solving the MILP master problem, we fix the binary variables in the original MINLP to the optimal solution of the master problem and solve the resulting nonconvex NLP. If the nonconvex NLP provides a feasible solution, it is feasible to the original problem and is an upper bound (UB) of the optimal objective function value.

Additionally, we include a 'no-good' cut that removes from the feasible set all the previously found binary variable combinations at iteration k,

$$\sum_{i=1}^{m} a_i^{k'} y_i \leq b^{k'}, \text{where } a_i^{k'} = \begin{cases} 1 \text{ if } y_i^{k'} = 1 \\ -1 \text{ if } y_i^{k'} = 0 \end{cases} \quad \text{and } b^{k'} = \sum_{i=1}^{m} y_i^{k'} - 1, \forall k' < k \tag{3}$$

Finally, we keep iterating between the MILP master problem and the NLP subproblem until the upper and lower bounds are within certain tolerance.

The deterministic nonconvex OA algorithm can be extended to (SP-MINLP) by generating i.i.d. samples for both the master problem and the nonconvex subproblem using an internal sampling approach. We define all the subproblems in the SAAOA algorithm in the next subsection.

3.1 Subproblems definition

The (SP-OA-MILP) master problem at iteration k is defined as

$$\text{(SP-OA-MILP)} \quad LB^k = \min_{x \in X, y \in \{0,1\}^m} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \min_{z \in Z} \hat{f}\left(x, y, z, \theta\right) \tag{4a}$$

$$\text{s.t.} \quad \hat{g}(x, y, z, \theta) \le 0; \quad \text{Eq. (3)} \tag{4b}$$

where $\hat{f}$ and $\hat{g}$ are polyhedral relaxations for function f, and g, respectively. LB^k represents the LB of (SP-MINLP) after $k-1$ iterations, where $k-1$ 'no-good' cuts (3) have been added. Parameter $y_i^{k'}$ is the optimal value of the ith binary variable at iteration k'.

Problem (SP-OA-MILP) can be approximated by generating N i.i.d. samples of θ_i from probability distribution $\mathbb{P}$ and solving the following (SAA-OA-MILP) master problem.

$$\text{(SAA-OA-MILP)} \quad \hat{w}_N^k = \min_{x \in X, z_i \in Z, y \in \{0,1\}^m} \frac{1}{N} \sum_{i=1}^{N} \hat{f}\left(x, y, z_i, \theta_i\right) \tag{5a}$$

$$\text{s.t.} \quad \hat{g}(x, y, z_i, \theta_i) \le 0 \quad \forall i \in [N]; \quad \text{Eq. (3)} \tag{5b}$$

An UB to (SP-MINLP) can be found by fixing the binary variables at a value $\bar{y}$ and solving problem (SP-nonconvex-NLP),

$$\text{(SP-nonconvex-NLP)} \quad UB(\bar{y}) = \min_{x \in X} \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \min_{z \in Z} f\left(x, \bar{y}, z, \theta_i\right) \tag{6a}$$

$$\text{s.t.} \quad g(x, \bar{y}, z, \theta) \le 0 \tag{6b}$$

The nonconvex NLP at fixed binary variable $\bar{y}$ by SAA is defined as,

$$\text{(SAA-nonconvex-NLP)} \quad \hat{u}_N(\bar{y}) = \min_{x \in X, z_i \in Z} \frac{1}{N} \sum_{i=1}^{N} f\left(x, \bar{y}, z_i, \theta_i\right) \tag{7a}$$

$$\text{s.t.} \quad g(x, \bar{y}, z_i, \theta_i) \le 0 \quad \forall i \in [N] \tag{7b}$$

where N i.i.d. samples of $\theta_i \in \Theta$ are generated from probability distribution $\mathbb{P}$.

3.2 Internal sampling using outer approximation

The OA algorithm iterates between the (SAA-OA-MILP) master problem and the (SAA-nonconvex-NLP) subproblem. N i.i.d. samples are generated for both the master and the subproblem at each iteration. A 'no-good' cut (3) is added to the (SAA-OA-MILP) master problem to eliminate the current integer solution. The steps of the SAAOA algorithm is described in Algorithm 1.

Algorithm 1

Initialization: Iteration counter $k = 1$; upper bound $UB = +\infty$; sample size N

Step 1 Convexify the (SAA-nonconvex-MINLP) with samples of size N and generate (SAA-OA-MILP)

Step 2 Solve the (SAA-OA-MILP). Denote the optimal objective value as $\hat{w}_N^k$ and the integer variable value as $\bar{y}^k$. Set $LB = \hat{w}_N^k$

If $LB \geq UB - \epsilon$, then go to step 5; otherwise go to step 3.

Step 3 Fix the binary variables in the (SAA-nonconvex-NLP) to $\bar{y}^k$, i.e., set $\bar{y} = \bar{y}^k$

Solve the (SAA-nonconvex-NLP) to global optimality. Denote the objective value as $\hat{u}_N^k$ and the optimal solution as $\tilde{x}^k, \bar{y}^k$.

If $\hat{u}_N^k \leq UB - \epsilon$, then let $UB = \hat{u}_N^k$, $x_p^* = \tilde{x}^k$, $y_p^* = \bar{y}^k$.

If $LB \geq UB - \epsilon$, then go to step 5; otherwise go to step 2.

Step 4 Let $k = k + 1$ and return to step 2

Step 5 Stop, the optimal solution is x_p^*, y_p^*, and the optimal objective value is UB.

---

4 Convergence results

4.1 Convergence of objective values and solutions

In this subsection, we prove that the optimal value and solutions obtained the proposed Algorithm 1 converge to the optimal value and solutions of (SP-MINLP) with probability (w.p.) one as the sample size $N \to \infty$. Note that by optimal solutions, we mean the optimal first-stage decisions.

4.1.1 Convergence of the upper bound.

In order to establish the results, we first prove that the convergence of the UB (SAA-nonconvex-NLP) to (SP-nonconvex-NLP). Note that the optimal objective values of (SAA-nonconvex-NLP) and (SP-nonconvex-NLP) when y is fixed at $\bar{y}$ are defined as $UB(\bar{y})$ and $\hat{u}_N(\bar{y})$, respectively. We further define the set of optimal solutions of (SP-nonconvex-NLP) as $S^*(\bar{y})$; the set of optimal solutions of (SAA-nonconvex-NLP) as $\hat{S}_N(\bar{y})$. We want to prove that $\lim_{N\to\infty} \hat{u}_N(\bar{y}) = UB(\bar{y})$ and the event $\{S^*(\bar{y}) = \hat{S}_N(\bar{y})\}$ happens w.p. 1 as $N \to \infty$.

We define function $UB(\bar{y}, x)$ as

$$UB(\bar{y}, x) = \mathop{\mathbb{E}}_{\theta\sim\mathbb{P}} \min_{z\in Z} f\left(x, \bar{y}, z, \theta_i\right) \tag{8a}$$

$$\text{s.t.} \quad g(x, \bar{y}, z, \theta) \leq 0 \tag{8b}$$

Function $\hat{u}_N(\bar{y}, x)$ is defined as,

$$\hat{u}_N(\bar{y}, x) = \frac{1}{N} \sum_{i=1}^{N} \min_{z_i\in Z} f\left(x, \bar{y}, z_i, \theta_i\right) \tag{9a}$$

$$\text{s.t.} \quad g(x, \bar{y}, z_i, \theta_i) \leq 0 \quad \forall i \in [N] \tag{9b}$$

Note that $\hat{u}_N(\bar{y}, x)$ contains the summation of N i.i.d samples. According to the Law of Large Numbers, $\hat{u}_N(\bar{y}, x)$ converges pointwise w.p. 1 to $UB(\bar{y}, x)$ as $N \to \infty$. Moreover, under some mild conditions ([13, 28, 29, 30]), the convergence is uniform. To prove the convergence of the UB, we make the following assumptions, which are similar to the assumptions in Theorem 5.3 of Shapiro et al. [30].

Assumption 2 *All the samples θ_is in all the SAA problems are i.i.d. from a distribution $\mathbb{P}$.*

Assumption 3 *Set X is compact.*

Assumption 4 *The set $S^*(\bar{y})$ of optimal solutions of the problem (SP-nonconvex-NLP) is nonempty, for any vector $\bar{y} \in \{0, 1\}^m$.*

Assumption 5 *The function $UB(\bar{y}, x)$ is finite valued and continuous in x on X, for any vector $\bar{y} \in \{0, 1\}^m$.*

Assumption 6 *The set $\hat{S}_N(\bar{y})$ is nonempty w.p. 1 for $N \to \infty$, for any vector $\bar{y} \in \{0,1\}^m$.*

Assumption 7 *The function $\hat{u}_N(\bar{y}, x)$ converges to $UB(\bar{y}, x)$ w.p. 1, as $N \to \infty$, uniformly on X, for any vector $\bar{y} \in \{0,1\}^m$.*

If Assumptions 3-7 hold true, $\lim_{N\to\infty} \hat{u}_N(\bar{y}) = UB(\bar{y})$ and the event $\{S^*(\bar{y}) = \hat{S}_N(\bar{y})\}$ happens w.p. 1 as $N \to \infty$, for any vector $\bar{y} \in \{0,1\}^m$ (Theorem 5.3 of Shapiro et al. [30]).

According to Assumption 7,

$$\max_{x\in X} |\hat{u}_N(\bar{y}, x) - UB(\bar{y}, x)| \to 0, \text{ w.p. 1 as } N \to \infty$$

We can bound $|\hat{u}_N(\bar{y}) - UB(\bar{y})|$ by,

$$\begin{aligned}
|\hat{u}_N(\bar{y}) - UB(\bar{y})| &= |\min_{x\in X} \hat{u}_N(\bar{y}, x) - \min_{x\in X} UB(\bar{y}, x)| \\
&= \max\left\{\min_{x\in X} \hat{u}_N(\bar{y}, x) - \min_{x\in X} UB(\bar{y}, x), \min_{x\in X} UB(\bar{y}, x) - \min_{x\in X} \hat{u}_N(\bar{y}, x)\right\} \\
&\leq \max\left\{\hat{u}_N(\bar{y}, x^*_{UB}(\bar{y})) - UB(\bar{y}, x^*_{UB}(\bar{y})), UB(\bar{y}, x^*_u(\bar{y})) - \hat{u}_N(\bar{y}, x^*_u(\bar{y}))\right\} \\
&\leq \max_{x\in X} |\hat{u}_N(\bar{y}, x) - UB(\bar{y}, x)|
\end{aligned}$$

where $x^*_{UB}(\bar{y}) = \text{argmin}_{x\in X} UB(\bar{y}, x)$, $x^*_u(\bar{y}) = \text{argmin}_{x\in X} \hat{u}_N(\bar{y}, x)$. Therefore, we have $\lim_{N\to\infty} \hat{u}_N(\bar{y}) = UB(\bar{y})$.

Now we need to prove that the event $\{S^*(\bar{y}) = \hat{S}_N(\bar{y})\}$ happens w.p. 1 as $N \to \infty$, for any vector $\bar{y} \in \{0,1\}^m$. We prove this by contradiction. Due to the compactness of X, we can assume that there exists $\hat{x}_N(\bar{y}) \in \hat{S}_N(\bar{y})$ such that $\text{dist}\big(\hat{x}_N(\bar{y}), S^*(\bar{y})\big) \geq \epsilon$, for some $\epsilon > 0$, and that $\hat{x}_N(\bar{y})$ tends to a point $x^*(\bar{y}) \in X$. Since $x^*(\bar{y}) \notin S^*(\bar{y})$, we have $UB(\bar{y}, x^*(\bar{y})) > UB(\bar{y})$.

$$\begin{aligned}
\hat{u}_N\big(\bar{y}, \hat{x}_N(\bar{y})\big) - UB\big(\bar{y}, x^*(\bar{y})\big) =& \Big[\hat{u}_N\big(\bar{y}, \hat{x}_N(\bar{y})\big) - UB(\bar{y}, \hat{x}_N(\bar{y}))\Big] \\
&+ \Big[UB\big(\bar{y}, \hat{x}_N(\bar{y})\big) - UB\big(\bar{y}, x^*(\bar{y})\big)\Big]
\end{aligned}$$

The first term on the right hand side goes to zero by the uniform convergence of $\hat{u}_N(\bar{y}, x)$ (Assumption 7). The second term goes to zero by the continuity of function $UB(\bar{y}, x)$ (Assumption 5). Then we have $\lim_{N\to\infty} \hat{u}_N\big(\bar{y}, \hat{x}_N(\bar{y})\big) = UB\big(\bar{y}, x^*(\bar{y})\big) > UB(\bar{y})$, which contradicts with the convergence of the objective value $\hat{u}_N(\bar{y})$ to $UB(\bar{y})$.

Under assumptions 3-7, the following lemma holds.

4.1.2 Convergence of the upper estimator of the upper bound

Note that in each iteration of the proposed algorithm, the upper estimator of the UB is defined by solving N' i.i.d samples of (single-nonconvex-NLP) described in Eq. (14) where the $\bar{x}$ and the $\bar{y}$ are fixed at $\tilde{x}^k$ and $\bar{y}^k$, respectively. The value of $\tilde{x}^k$ is determined by heuristics. The average of N' i.i.d samples of problem (single-nonconvex-NLP) at iteration k is defined as $\bar{u}^k_{N'}$. Here, we prove that by using a particular heuristic for fixing $\tilde{x}^k$, the optimal value $\bar{u}^k_{N'}$ converges to the optimal value of (SP-nonconvex-NLP), i.e., $UB(\bar{y}^k)$ as $N \to \infty$. The heuristic is defined as 'Optimal Solution Heuristic'.

4.1.3 Optimal Solution Heuristic

The value of $\tilde{x}^k$ is fixed at one of the optimal solutions of problem (SAA-nonconvex-NLP). As $N \to \infty$ and $N' \to \infty$, by applying the Optimal Solution Heuristic, the average of the optimal values of N' (single-nonconvex-NLP) at any iteration k, $\bar{u}^k_{N'}$, tends to $UB(\bar{y}^k)$ w.p. 1. From Lemma 4.1.1, as $N \to \infty$, any optimal solution of problem (SAA-nonconvex-NLP) is optimal for problem (SP-nonconvex-NLP), i.e., $\forall \hat{S}_N(\bar{y})$, we have $x \in S^*(\bar{y})$. If Optimal Solution Heuristic is applied, the N' (single-nonconvex-NLP) have the x variables fixed at the optimal solution of (SP-nonconvex-NLP). According to the Law of Large Numbers, the average of the optimal values of N' (single-nonconvex-NLP) goes to $UB(\bar{y}^k)$ w.p. 1 as $N' \to \infty$.

4.1.4 Convergence of the optimal values and solutions of the lower bound.

Now, we prove the convergence of the LB estimators. Similar to the analysis of the UB, we define function, $LB^k(x,y)$,

$$LB^k(x,y) = \mathop{\mathbb{E}}_{\theta \sim \mathbb{P}} \min_{z \in Z} \hat{f}\,(x,y,z,\theta) \tag{10a}$$

$$\text{s.t.} \quad \hat{g}(x,y,z,\theta) \leq 0; \quad \text{Eq. (3)} \tag{10b}$$

Function $\hat{w}^k_N(x,y)$ is defined as,

$$\hat{w}^k_N(x,y) = \frac{1}{N}\sum_{i=1}^{N} \min_{z_i \in Z} \hat{f}\,(x,y,z_i,\theta_i) \tag{11a}$$

$$\text{s.t.} \quad \hat{g}(x,y,z_i,\theta_i) \leq 0; \quad \text{Eq. (3)} \tag{11b}$$

The optimal value of (SP-OA-MILP) is defined as LB^k. The optimal value of (SAA-OA-MILP) is defined as $\hat{w}_N^k$. We define the set of optimal solution for (SP-OA-MILP) at iteration k as T_k^*, and the set of optimal solutions for problem (SAA-OA-MILP) at iteration k with sample size N as $\hat{T}_k^N$. We want to prove that $\lim_{N\to\infty} \hat{w}_N^k = LB^k$ and the event $\{T_k^* = \hat{T}_k^N\}$ happens w.p. 1 as $N \to \infty$. The complication compared with the previous subsection is the presence of binary variables in (SP-OA-MILP). Note that in the proof for the convergence of (SP-nonconvex-NLP) the binary variables are fixed. However, if we consider some fixed $\bar{y}$ that is feasible for (SP-OA-MILP) at iteration k, by making similar assumptions to Assumptions 3-5, we can prove the convergence of the optimal values and solutions of the LB estimators in (SAA-OA-MILP) to (SP-OA-MILP) with the y variables fixed at $\bar{y}$. Let the optimal value of (SP-OA-MILP) with y fixed at $\bar{y}$ be $LB^k(\bar{y})$. The optimal value of (SP-OA-MILP) can be seen as taking the minimum over all the possible $LB^k(\bar{y})$. Since the combinations of binary variables are finite, the convergence of both the optimal values and optimal solutions to (SP-OA-MILP) can be established. We state the following lemma whose major steps of proofs are similar to Lemma 4.1.1, therefore, we omit its proof. Under mild conditions similar to Assumptions 3-7, the optimal value and the optimal solutions of (SAA-OA-MILP) converges to those of (SP-OA-MILP) w.p. 1 as $N \to \infty$, i.e., $\lim_{N\to\infty} \hat{w}_N^k = LB^k$, the event $\{T_k^* = \hat{T}_k^N\}$ happens w.p. 1 as $N \to \infty$.

4.1.5 Convergence of optimal objective values and solutions of the proposed algorithm.

The convergence of the optimal values and solutions of the LB and the UB is proved setting up the main theorem of this section. The proposed Algorithm 1 returns the optimal value and the set of optimal solutions (x^*, y^*) of (SP-MINLP) w.p. 1 as $N \to \infty$. Note that the Algorithm 1 can only miss finding the optimal solution y^* in 1) the comparison of the optimal objective value of (SAA-nonconvex-NLP), $\hat{u}_N^k$, and the current upper bound UB, and 2) the comparison of the objective value of the (SAA-OA-MILP) with the current UB. We need to prove that neither of these two cases can happen as $N \to \infty$. Note that Lemmas 4.1.1 and 4.1.4 show that all the estimators are 'exact' in the sense they converge to the optimal value of the corresponding true stochastic programming problems (SP-nonconvex-NLP) and (SP-OA-MILP). Therefore, the optimal solution cannot be missed in either of the two cases. Since there are only finite combinations of binary variables, it takes the proposed algorithm finite number of iterations to converge. The optimal value returned is the optimal value of the true (SP-MINLP) and the optimal solution comes from the y^* that yields the best UB.

4.2 Estimates of sample sizes

Now that we have proved that Algorithm 1 converges in the limit, it is desirable to estimate the sample size to achieve finite error ϵ with high probability.

Ahmed and Shapiro [1] give the sample size estimator for two-stage SP. An SAA estimator, which is solved to δ-optimality, gives the ϵ-optimal solution to the corresponding true problem with probability at least $1-\alpha$ if the sample size

$$N \geq \frac{12\sigma^2}{(\epsilon-\delta)^2}\Big(n_1 \ln \frac{2DL}{\epsilon-\delta} - \ln \alpha\Big) \tag{12}$$

where D is the diameter of set X, the objective function of the SP problem is assumed to be L-Lipschitz continuous on X, n_1 is the dimension of the first stage variables, and σ^2 is the maximal variance of certain differences between values of the objective function of the SAA problem (see [14]). Eq. (12) gives the rate of convergence for any SAA estimators of any two-stage SP. Theoretically, it can be used the calculate the sample sizes of the MILP master problem and the nonconvex NLP subproblem, respectively, if we have the upper bounds of the diameter of the feasible set X and the Lipschitz constant L. However, for stochastic MINLP problems, the Lipschitz constant L is difficult to estimate. Moreover, the sample size estimates from (12) is usually too conservative in practice [14]. Therefore, we need to design an algorithm based on Algorithm 1 and this observation that is more efficient in practice.

5 Algorithm Design

Here we extend Algorithm 1 to a more practical algorithm based on the ideas of Kleywegt et al. [14] where it is proposed an empirical method to construct confidence intervals for the optimal objective value of the 'true' SP problem.

For a finite sample size N, Mak et al. [23] prove that the expectation of the SAA problems provides lower bounds for the corresponding true SP problems. More specifically, for the objective of (SP-OA-MILP), LB^k, and the objective of (SAA-OA-MILP), $\hat{w}_N^k$, at iteration k, we have,

$$\mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\hat{w}_N^k \right] \leq LB^k$$

For the objective of (SP-nonconvex-NLP), $UB(\bar{y})$, and the objective of the corresponding (SAA-nonconvex-NLP), $\hat{u}_N(\bar{y})$, we have,

$$\mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\hat{u}_N(\bar{y}) \right] \leq UB(\bar{y})$$

Therefore, the SAA estimators, $\hat{w}_N^k$ and $\hat{u}_N(\bar{y})$, can be regarded as the lower estimators of (SP-OA-MILP) and (SP-nonconvex-NLP), respectively. In order to construct the confidence intervals for (SP-OA-MILP) and (SP-nonconvex-NLP), we need to provide their upper estimators.

5.1 Upper estimators

Mak et al. [23] prove that the upper estimator of a SP can be obtained by evaluating the sample mean at a given feasible first stage decision. In our case, the upper estimator of the (SP-OA-MILP) can be obtained by solving the following problem for random sample θ_i, $i = 1, ..., N_l'$

$$\text{(single-OA-LP)} \quad \hat{w}_1^{(i),k} = \min_{z_i \in Z} \hat{f}\left(\tilde{x}, \tilde{y}, z_i, \theta_i\right) \tag{13a}$$

$$\text{s.t.} \quad \hat{g}(\tilde{x}, \tilde{y}, z_i, \theta_i) \leq 0 \tag{13b}$$

where the first stage decisions (x, y) are fixed at $(\tilde{x}, \tilde{y})$. The values of $(\tilde{x}, \tilde{y})$ can come from a good estimate of the optimal solution of (SP-OA-MILP), for example, the optimal solution of (SAA-OA-MILP). The upper estimator of the (SP-OA-MILP) can be the average of the N_l' (single-OA-LP) problems,

$$\bar{w}_{N_l'}^k = \frac{1}{N_l'} \sum_{i=1}^{N_l'} \hat{w}_1^{(i),k}$$

Similarly, the upper estimator of (SP-nonconvex-NLP) can be derived by fixing the first stage decisions (x, y) to $(\bar{x}, \bar{y})$ and solve the rest of the nonconvex NLP for N_u' samples of θ_i. The values for $(\bar{x}, \bar{y})$ can come from the optimal solution of (SAA-OA-MILP) $(\bar{x}^k, \bar{y}^k)$. The ith single size nonconvex NLP at iteration k is defined as,

$$\text{(single-nonconvex-NLP)} \quad \hat{u}_1^{(i),k} = \min_{z_i \in Z} f\left(\bar{x}, \bar{y}, z_i, \theta_i\right) \tag{14a}$$

$$\text{s.t.} \quad g(\bar{x}, \bar{y}, z_i, \theta_i) \le 0 \tag{14b}$$

The upper estimator of the (SP-nonconvex-NLP) can be the average of the N'_u (single-nonconvex-NLP) problems,

$$\bar{u}^k_{N'_u} = \frac{1}{N'_u} \sum_{i=1}^{N'_u} \hat{u}_1^{(i),k}$$

5.2 Confidence intervals for the upper and lower bound

We show how the confidence intervals of the upper and the lower bounds at every iteration k of the SAAOA algorithm can be constructed.

We first show how the confidence interval of (SP-nonconvex-NLP) can be constructed at each iteration. Recall that the expectation of (SAA-nonconvex-NLP) provides a LB of (SP-nonconvex-NLP), i.e., $\mathbb{E}[\hat{u}_N(\bar{y})] \le UB(\bar{y})$. To have a good estimate for the expectation, we solve M_u batches of (SAA-nonconvex-NLP). We make the following assumption about the batches of the SAA problems,

Assumption 8 *All the M_u (M_l) batches of samples in the (SAA-nonconvex-NLP) ((SAA-OA-MILP)) problems are i.i.d..*

At each iteration k, (SAA-nonconvex-NLP) with N_u samples is solved M_u times to obtain a lower estimator of the upper bound $UB(\bar{y}^k)$. We use random variable $\hat{u}^{(m),k}_{N_u}$ to denote the optimal objective value of the mth batch of (SAA-nonconvex-NLP) at iteration k. From Mak et al. [23], random variable $\hat{u}^{(m),k}_{N_u}$ is a biased estimator of $UB(\bar{y}^k)$, i.e.,

$$\mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\hat{u}^{(m),k}_{N_u} \right] \le UB(\bar{y}^k) \tag{15}$$

The mean of the M_u batches (SAA-nonconvex-NLP) is defined as $\bar{u}^k_{N_u,M_u} = \frac{1}{M_u} \sum_{m=1}^{M_u} \hat{u}^{(m),k}_{N_u}$. By central limit theorem,

$$\sqrt{M_u} \Big(\bar{u}^k_{N_u,M_u} - \mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\bar{u}^k_{N_u,M_u} \right] \Big) \Rightarrow \mathcal{N}(0, \sigma^2_{u,k}) \tag{16}$$

where '$\Rightarrow$' denotes convergence in distribution, and $\mathcal{N}(0, \sigma^2)$ denotes a normal distribution with mean zero and variance σ^2. $(\hat{S}^{u,k}_{M_u})^2$ is the standard sample variance estimator of $\sigma^2_{u,k}$ is defined by

$$\frac{(\hat{S}^{u,k}_{M_u})^2}{M_u} = \frac{1}{M_u(M_u-1)} \sum_{m=1}^{M_u} (\hat{u}^{(m),k}_{N_u} - \bar{u}^k_{N_u,M_u})^2 \tag{17}$$

Therefore, the $(1-\alpha)$ confidence interval of the lower estimator of the UB can be approximated by,

$$\left(\bar{u}^k_{N_u,M_u} - t^{\alpha/2}_{M_u-1} \frac{\hat{S}^{u,k}_{M_u}}{\sqrt{M_u}}, \quad \bar{u}^k_{N_u,M_u} + t^{\alpha/2}_{M_u-1} \frac{\hat{S}^{u,k}_{M_u}}{\sqrt{M_u}}\right) \tag{18}$$

where $t^{\alpha/2}_{M_u-1}$ is the $1-\alpha/2$ quantile of t-distribution with $M_u - 1$ degrees of freedom.

At each iteration k, N'_u (single-nonconvex-NLP)s are solved to estimate the UB of (SP-nonconvex-NLP). We use $\hat{u}^{(i),k}_1$ to denote the optimal objective value of the ith sample of (single-nonconvex-NLP) at iteration k. By definition, we also have the mean of the N'_u samples, $\bar{u}^k_{N'_u} = \frac{1}{N'_u} \sum_{i=1}^{N'_u} \hat{u}^{(i),k}_1$. From Mak et al. [23], we have

$$\mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\bar{u}^k_{N'_u}\right] \leq UB(\bar{y}^k) \tag{19}$$

In Eqs. (16) and (17) By central limit theorem,

$$\sqrt{N'_u}\Big(\bar{u}^k_{N'_u} - \mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\bar{u}^k_{N'_u}\right]\Big) \Rightarrow N(0, \sigma^2_{u',k}) \tag{20}$$

The standard sample variance estimator $(\hat{S}^{u,k}_{N'_u})^2$ of $\sigma^2_{u',k}$ is defined by,

$$\frac{(\hat{S}^{u,k}_{N'_u})^2}{N'_u} = \frac{1}{N'_u(N'_u-1)} \sum_{i=1}^{N'_u} \left(\hat{u}^{(i),k}_1 - \bar{u}^k_{N'_u}\right)^2 \tag{21}$$

Therefore, the $(1-\alpha)$ confidence interval of the upper estimator of the UB can be approximated by,

$$\left(\bar{u}^k_{N'_u} - t^{\alpha/2}_{N'_u-1} \frac{\hat{S}^{u,k}_{N'_u}}{\sqrt{N'_u}}, \quad \bar{u}^k_{N'_u} + t^{\alpha/2}_{N'_u-1} \frac{\hat{S}^{u,k}_{N'_u}}{\sqrt{N'_u}}\right) \tag{22}$$

where $t^{\alpha/2}_{N'_u-1}$ is the $1-\alpha/2$ quantile of the t-distribution with $N'_u - 1$ degrees of freedom. By combining (22) and (18), we have with probability at least

$(1-\alpha)$, the UB obtained at iteration k, $UB(\bar{y}^k)$, lies within the interval,

$$(\overline{UB}, \underline{UB}) = \left(\bar{u}^k_{N_u,M_u} - t^{\alpha/2}_{M_u-1}\frac{\hat{S}^{u,k}_{M_u}}{\sqrt{M_u}}, \quad \bar{u}^k_{N'_u} + t^{\alpha/2}_{N'_u-1}\frac{\hat{S}^{u,k}_{N'_u}}{\sqrt{N'_u}}\right) \tag{23}$$

where $(\hat{S}^{u,k}_{N'_u})^2$ is the standard sample variance estimator of $\sigma^2_{u',k}$, the variance of the scaled normal distribution to what $\bar{u}^k_{N'_u}$ minus its expected value converges in distribution.

Similarly, to construct the confidence interval for (SP-OA-MILP) at each iteration k, we can solve the (SAA-OA-MILP) with N_l i.i.d. samples for M_l i.i.d. batches. The optimal objective of the mth batch is denoted as $\hat{w}^{(m),k}_{N_l}$. We compute the mean and variance of the M_l batches by,

$$\bar{w}^k_{N_l,M_l} = \tfrac{1}{M_l}\textstyle\sum_{m=1}^{M_l} \hat{w}^{(m),k}_{N_l} \text{ and } \tfrac{(\hat{S}^{w,k}_{M_l})^2}{M_l} = \tfrac{1}{M_l(M_l-1)}\textstyle\sum_{m=1}^{M_l}(\hat{w}^{(m),k}_{N_l} - \bar{w}^k_{N_l,M_l})^2$$

Similar to Eq. (18), the $(1-\alpha)$ confidence interval of the lower estimator of the LB is approximately,

$$\left(\bar{w}^k_{N_l,M_l} - t^{\alpha/2}_{M_l-1}\frac{\hat{S}^{w,k}_{M_l}}{\sqrt{M_l}}, \quad \bar{w}^k_{N_l,M_l} + t^{\alpha/2}_{M_l-1}\frac{\hat{S}^{w,k}_{M_l}}{\sqrt{M_l}}\right) \tag{24}$$

For the upper estimator of (SP-OA-MILP), we can solve (single-OA-LP) N'_l times with N'_l i.i.d. samples of θ_i. The optimal objective value of the ith (single-OA-LP) is denoted as $\hat{w}^{(i),k}_1$. We can compute the mean and variance of the objective values with,

$$\bar{w}^k_{N'_l} = \tfrac{1}{N'_l}\textstyle\sum_{i=1}^{N'_l} \hat{w}^{(i),k}_1 \text{ and } \tfrac{(\hat{S}^{w,k}_{N'_l})^2}{N'_l} = \tfrac{1}{N'_l(N'_l-1)}\textstyle\sum_{i=1}^{N'_l}(\hat{w}^{(i),k}_1 - \bar{w}^k_{N'_l})^2$$

As in Eq. (22), the $(1-\alpha)$ confidence interval of the for the upper estimator of the LB can be approximated by,

$$\left(\bar{w}^k_{N'_l} - t^{\alpha/2}_{N'_l-1}\frac{\hat{S}^{w,k}_{N'_l}}{\sqrt{N'_l}}, \quad \bar{w}^k_{N'_l} + t^{\alpha/2}_{N'_l-1}\frac{\hat{S}^{w,k}_{N'_l}}{\sqrt{N'_l}}\right) \tag{25}$$

With Eqs. (24) and (25), the $(1-\alpha)$ confidence interval of the LB can be approximated by,

$$(\overline{LB}, \underline{LB}) = \left(\bar{w}^k_{N_l,M_l} - t^{\alpha/2}_{M_l-1}\frac{\hat{S}^{w,k}_{N_l,M_l}}{\sqrt{M_l}}, \quad \bar{w}^k_{N'_l} + t^{\alpha/2}_{N'_l-1}\frac{\hat{S}^{w,k}_{N'_l}}{\sqrt{N'_l}}\right) \tag{26}$$

5.3 SAAOA with confidence intervals

With the confidence interval results from Eqs. (23) and (26), we can approximate the values of the upper and lower bounds of the 'true' SP at each iteration of the OA algorithm with high probability. The high-level overview of the SAAOA with confidence intervals is shown in Figure 1. At each iteration, we solve M_l (SAA-OA-MILP) problems each with size N_l. Then we fix the first stage binary and continuous variables and solve N_l' (single-OA-LP) to construct the upper estimator of the LB. After that, we only fix the first stage binary variables y and solve M_u batches of (SAA-nonconvex-NLP) each with sample size N_u to construct the lower estimator of the UB. Then we solve N_u (single-nonconvex-NLP) to construct the upper estimator of the UB. At the end of each iteration, we check if the algorithm converges. The steps of the SAAOA method with confidence interval estimators are described in Algorithm 2.

It is difficult to estimate the desired sample sizes for the estimators *a priori*. Therefore, we may need to update the sample sizes, N_u, M_u, N_u', N_l, M_l, N_l', if the confidence intervals are not tight enough. The update policies for those parameters are not unique. We discuss the update policies in the next subsection.

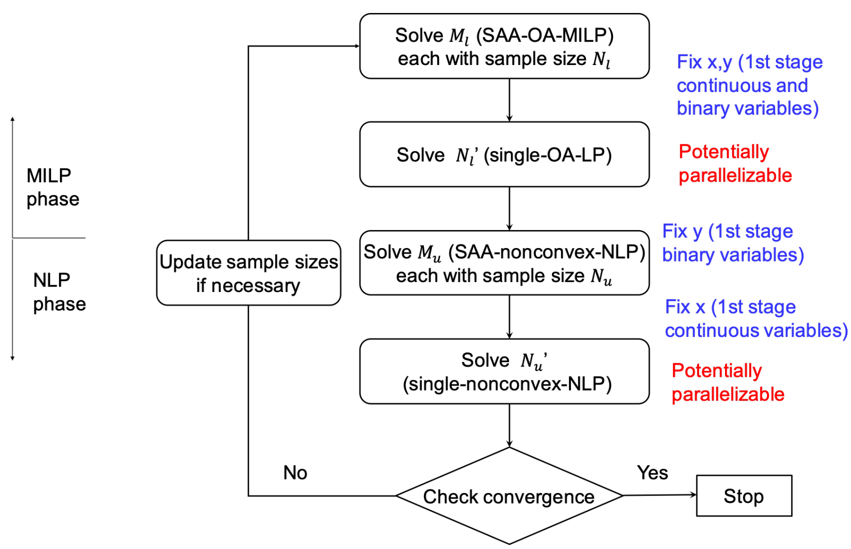

Figure 1: Flowchart of the SAAOA algorithm with confidence interval estimators

5.4 Update policies

As we have discussed, estimating the proper sample size before we solve any of the SAA problems is challenging. Small sample sizes may yield large confidence intervals and provide poor estimates for the upper and lower bounds. On the other hand, large sample sizes becomes too conservative and increases the computational time. Therefore, choosing the update policies on N_u, M_u, N'_u, N_l, M_l, N'_l, is crucial to the performance of the algorithm.

For the upper estimators, we are solving N'_u or N'_l single sample size problems. The number N'_u, N'_l affect the accuracy of the estimates of feasible solutions $(\tilde{x}, \tilde{y})$. The sample sizes N'_u, N'_l should increase if the confidence intervals for the upper estimators in Eqs. (22) and (25) are too large.

For the lower estimators, we have two types of parameters to tune, i.e., the batch sizes M_l and M_u, and the sample size of each batch N_l, N_u. Increasing the value of the batch sizes M_l and M_u make the estimators for the expected value more accurate, i.e., the length of the confidence intervals for the lower estimators in Eqs. (18) and (24) will decrease. The impact of increasing the value of sample size N_l, N_u is two-fold. First, the expectation of the lower estimators becomes tighter with the increase in the sample size, which is proved by Mak et al. [23].

$$\mathbb{E}_{\theta_i \sim \mathbb{P}}\left[\hat{u}_{N_u}^{(m),k}\right] \leq \mathbb{E}_{\theta_i \sim \mathbb{P}}\left[\hat{u}_{N_u+1}^{(m),k}\right] \leq UB(\bar{y}^k), \quad \forall N_u \tag{27}$$

Therefore, increasing the sample size makes the lower estimator tighter and Algorithm 2 may converge in fewer iterations. Second, the variance of the lower estimator decreases with the increase in the number of samples. With the same number of batches, tighter confidence intervals can be obtained with larger sample size.

We describe two update policies for choosing the sample and batch sizes.

Policy 1 (P1): Increase if loose confidence interval policy At each iteration, check if the confidence interval is tight enough. If not, multiply the number of samples by a fixed ratio. For example, if $t_{M_l-1}^{\alpha/2} \frac{\hat{S}_{N_l,M_l}^{w,k}}{\sqrt{M_l}}$ is large, increase N_l by setting $N_l^{\text{new}} = \beta N_l^{\text{old}}$, increase M_l by setting $M_l^{\text{new}} = \gamma M_l^{\text{old}}$, where β and γ are parameters greater than 1. Note that the 'no-good' cut is only added when the confidence interval for the UB is tight. Upper limits for N_u, M_u, N'_u, N_l, M_l, N'_l are set to avoid intractability, which could sacrifice the tightness of the confidence intervals.

Policy 2 (P2): Increase until overlap policy Do not increase the initial sample sizes until the confidence interval of the LB and the confidence interval of the UB overlap. After the overlap occurs, increase the sample sizes of (SAA-OA-MILP) and/or (single-OA-LP) at the current iteration if the confidence interval of the LB is not tight and increase the sample sizes of (SAA-nonconvex-NLP) and/or (single-nonconvex-NLP) corresponding to the best UB found so far if the confidence interval for the UB is not tight. The increase strategy could be multiplying fixed ratios similar to P1. Keep increasing the sample sizes until one of the three cases occur: 1) the confidence intervals of the lower and the upper bound become tight; 2) the confidence intervals no longer overlap but the LB is still lower than the current best UB; 3) the confidence intervals no longer overlap but the LB is greater than the best UB. In case 1), check if the UB is less than the LB, if not, keep iterating. Otherwise, terminate. In case 2), keep iterating until the bounds overlap again. In case 3), terminate. The 'no-good' cut is added in a given iteration if the confidence intervals do not overlap or the UB confidence interval is tight. Note that if keep iterating, then the same strategy is used recursively until one of the termination criteria is satisfied. Since the confidence intervals of the UB are not necessarily tight when the algorithm terminates, reevaluating all the integer solutions $\bar{y}^k$s may be needed to find the best feasible solution.

We give the following remark to end this section. The algorithm can be applied to two-stage SP with both continuous and binary first stage variables in the first stage. However, it works better for problems with pure binary variables in the first stage. In this case, we do not have to solve (SAA-nonconvex-NLP), which can be a large scale nonconvex NLP. Instead, we need to solve (single-nonconvex-NLP), potentially in parallel.

6 Computational Results

Algorithm 2 with the two proposed update policies is implemented in Pyomo / Python [7]. The proposed algorithm is implemented in a python package `saaoa.py` and can take a two-stage model in the data structure of PySP [31].

6.1 Stochastic Pooling problem

The stochastic pooling problem has been studied by Li et al. [19, 21]. The first stage decisions are investment decisions on sources, pools, and pipelines, which are represented as binary variables. The second decisions are the mass flows and the split fractions, which are represented as continuous variables. The constraints include mass balance, investment capacity, and quality specifications. The objective is to minimize the expected cost. We assume that the uncertainty comes from the quality of one source whose deviation from the nominal value follows a truncated distribution $\mathcal{N}(0, \sigma)$ where the parameter values less than -2σ or greater 2σ are truncated .

We apply the SAAOA algorithm described in Algorithm 2 to solve the problem for $\sigma = 0.0031$ and $\sigma = 0.004$ using both update policies P1 and P2. Since the problem only has pure binary first stage variables, we do not need to solve (SAA-nonconvex-NLP). The sample sizes and batch sizes that need to be updated are N_l, M_l, N'_l, N'_u. We multiply the sizes by fixed ratios, i.e., $N_l^{\text{new}} = \beta N_l^{\text{old}}$, $M_l^{\text{new}} = \gamma M_l^{\text{old}}$, ${N'}_l^{\text{new}} = \beta {N'}_l^{\text{old}}$, ${N'}_u^{\text{new}} = \beta {N'}_u^{\text{old}}$, where β and γ are constants greater than 1. In the SAAOA algorithm, we start with $N_l = 50$, $M_l = 10$, $N'_l = 50$, $N'_u = 50$ and update the sample sizes using P1 and P2 respectively. We calculate the 95% confidence intervals for both the upper and lower bound, i.e., $\alpha = 5\%$. A confidence interval is considered tight if the relative gap between its upper and lower estimators is less than 5%.

The convex relaxation for (SAA-MINLP) is obtained using a special version of BARON [12] which provides the root node polyhedral relaxation. (SAA-OA-MILP) and (single-OA-LP) are solved using CPLEX v.12.9 and the (single-nonconvex-NLP) problems are solved with BARON v.19.3.24.

To compare with the SAAOA algorithm, we use BARON to solve 10 batches of (SAA-MINLP) problems each with a sample size of 50. The time limit for each (SAA-MINLP) problem is set to 10,000 seconds. The expectation of (SAA-MINLP) provides a lower bound for the original (SP-MINLP). The lower estimator of (SP-MINLP) can be obtained similar to Eq. (26). Once we fix the binary variables from the optimal solution of (SAA-MINLP), we solve 100 individual (single-nonconvex-NLP) to estimate the expected value of the optimal solution. The best solution for the 10 batches is reported. All the problems are solved using one processor of an Intel Xeon (2.67GHz) machine with 64 GB RAM.

The computational results for $\sigma = 0.0031$ and $\sigma = 0.004$ are shown in Table 1. In the case of $\sigma = 0.0031$, P1 and P2 give the same results. We use different constants β and γ for updating the sample sizes. The upper and lower bound estimators when the SAAOA algorithm terminates are shown in the table. Note that we allow the algorithm to terminate after the confidence

Table 1: Computational results.

Variance	$\sigma = 0.0031$					$\sigma = 0.004$								
Policy	P1 and P2				DE	P1				P2				DE
β	2	2	1.5	1.5	-	2	2	1.5	1.5	2	2	1.5	1.5	-
γ	2	1.5	2	1.5	-	2	1.5	2	1.5	2	1.5	2	1.5	-
Wall time (s)	833	712	612	**541**	100,412	19,397	16,162	16,871	11,610	12,578	13,444	**7,676**	9,829	100,471
Iteration	9	9	9	9	-	22	24	20	20	21	22	**10**	12	-
$\overline{UB}$	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0
$\underline{UB}$	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0	-115.0
$\overline{LB}$	-105.0	-115.0	-105.0	-105.0	-	-107.2	-105.5	-109.6	-111.4	-108.7	-105.0	-103.5	-109.5	-
$\underline{LB}$	-113.5	-109.8	-110.9	-109.1	-172.4	-114.5	-114.8	-114.6	-114.5	-114.2	-113.8	-114.8	-113.9	-171.3
N_l	100	100	**75**	**75**	50	200	200	**168**	**168**	200	200	**168**	252	50
M_l	20	**15**	20	**15**	10	40	**22**	80	33	40	**22**	80	49	10
N_l'	100	100	**75**	**75**	-	3,200	2,000	2,868	2,868	1,600	3,200	**75**	168	-
N_u'	50	50	50	50	100	2,000	3,200	2,000	2,000	2,000	2,000	**50**	75	100

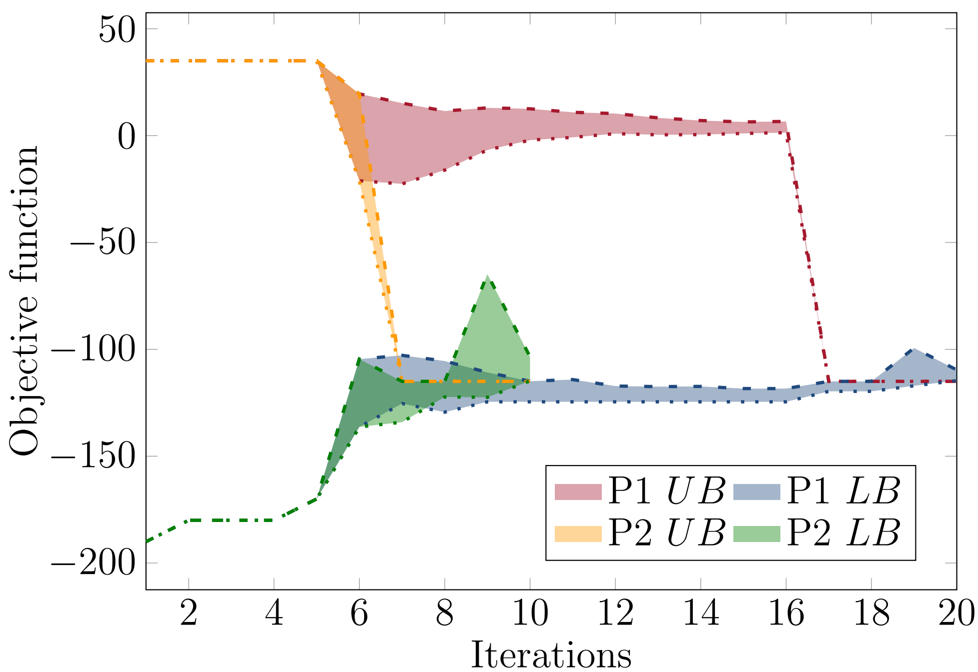

Figure 2: Bounds convergence for test case with $\sigma = 0.004$, $\beta = 1.5$, $\gamma = 2$

interval of the LB is strictly greater than the confidence interval of the UB. In all the four cases, the SAAOA algorithm returns the same optimal solution, which gives an optimal value of -115.0. In both policies, the parameters M_l, N_l, N_l' are updated once at the last iteration to tighten the LB. After the update, the algorithm terminates. Therefore, the smallest update ratio,

$\beta = 1.5$, $\gamma = 1.5$ gives the least computational time. The column 'DE' (deterministic equivalent) represents using BARON to solve (SAA-MINLP) directly. Here, we slightly abuse notation to report the UB estimator and the LB estimator returned by external sampling. It is easy to see that the SAAOA algorithm outperforms external sampling given that it avoids solving the large scale nonconvex MINLP problem directly.

Compared with the low variance case, in the case with $\sigma = 0.004$ the algorithms need more iterations to converge especially for P1. Recall that in P1, the sample sizes are updated whenever the confidence intervals are not tight. Therefore, if the confidence interval of the UB in a given iteration is not tight enough, we need to run another iteration with an increased sample size without adding a 'no-good' cut to cut off the current binary solution. In P2, the algorithm keeps adding one 'no-good' cut to the master problem at each iteration until the confidence intervals of the upper and the lower bound overlap. In general, P2 takes less computational time and fewer iterations to converge than P1.

To show how P1 and P2 perform differently, we show the convergence of the confidence intervals of $\sigma = 0.004$, $\beta = 1.5$, $\gamma = 2$ in Figure 2. In the beginning, the confidence intervals for both the upper and lower bounds are tight. When the confidence intervals become loose, P1 increases the sample sizes immediately to get tighter confidence intervals. As a result, P1 spends several iterations just to tighten the confidence intervals while the gap between bounds does not reduce significantly. On the other hand, P2 keeps adding 'no-good' cuts until the upper and the lower confidence intervals overlap. Therefore, P2 converges in a fewer number of iterations and saves computational time. In general, P2 outperforms P1 in terms of the number of iterations and computational time. In the test cases, they yield the same optimal solution. P1 is more conservative in adding 'no-good' cuts to the master problem, i.e., a 'no-good' cut can only be added if the confidence interval corresponding to the integer solution is tight. Computational time can be wasted in constructing a tight estimate for suboptimal solutions. However, if the user prefers to have a tight estimate for each solution so that we can be more certain that one solution is suboptimal to the optimal solution, P1 is preferable.

7 Conclusion

In this paper, we propose a sample average approximation based outer approximation (SAAOA) algorithm for solving two-stage nonconvex stochastic MINLPs. The SAAOA algorithm iterates between an MILP master problem (SAA-OA-MILP) and nonconvex NLP subproblems (SAA-nonconvex-NLP). We prove that the SAAOA algorithm converges as the sample size goes to infinity and provides a theoretical estimate for the sample size. Since the sample size estimates are too conservative in practice. We design an SAAOA algorithm with confidence interval estimates for the upper bound and the lower bound at each iteration of the OA algorithm. To construct the confidence intervals, we define (single-OA-LP) and (single-nonconvex-NLP), which are proved to provide upper estimators for the lower bound and upper bound, respectively. The sample sizes are updated dynamically using some update polices. We propose two update policies, namely, *P1: Increase if loose confidence interval policy* and *P2: Increase until overlap policy*. The algorithm is suitable for solving two-stage stochastic MINLPs with pure binary variables where the nonconvex NLP subproblems can be solved for each scenario separately. Computational results are shown for a stochastic pooling problem. The SAAOA algorithm with confidence interval estimates is shown to perform better than solving the deterministic equivalent (SAA-MINLP) directly in terms of computational time and optimality gap. We provide some criteria for selecting between update Policy 1 and Policy 2 in Remark 6.1.

Future work can be focused on improving the update policies. To find an update policy that works well in general, we need to have more test cases to benchmark different update policies and tune the parameters like the fixed ratio in the two update policies proposed in this paper.

Acknowledgements

The authors gratefully acknowledge financial support from the Center of Advanced Process Decision-making at Carnegie Mellon University.

References

[1] Ahmed, S., Shapiro, A., Shapiro, E.: The sample average approximation method for stochastic programs with integer recourse. Submitted for publication pp. 1–24 (2002)

[2] Alonso-Ayuso, A., Escudero, L.F., Ortuno, M.T.: BFC, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0–1 programs. European Journal of Operational Research **151**(3), 503–519 (2003)

[3] Atakan, S., Sen, S.: A progressive hedging based branch-and-bound algorithm for mixed-integer stochastic programs. Computational Management Science pp. 1–40 (2018)

[4] Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numerische mathematik **4**(1), 238–252 (1962)

[5] Cao, Y., Zavala, V.M.: A scalable global optimization algorithm for stochastic nonlinear programs. Journal of Global Optimization **75**(2), 393–416 (2019)

[6] Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical programming **36**(3), 307–339 (1986)

[7] Hart, W.E., Laird, C.D., Watson, J.P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., Siirola, J.D.: Pyomo-optimization modeling in python, vol. 67. Springer (2017)

[8] Higle, J.L., Sen, S.: Stochastic decomposition: An algorithm for two-stage linear programs with recourse. Mathematics of operations research **16**(3), 650–669 (1991)

[9] Kannan, R.: Algorithms, analysis and software for the global optimization of two-stage stochastic programs. Ph.D. thesis, Massachusetts Institute of Technology (2018)

[10] Karuppiah, R., Grossmann, I.E.: Global optimization of multiscenario mixed integer nonlinear programming models arising in the synthesis of

integrated water networks under uncertainty. Computers & Chemical Engineering **32**(1-2), 145–160 (2008)

[11] Kesavan, P., Allgor, R.J., Gatzke, E.P., Barton, P.I.: Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs. Mathematical Programming **100**(3), 517–535 (2004)

[12] Kılınç, M.R., Sahinidis, N.V.: Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with baron. Optimization Methods and Software **33**(3), 540–562 (2018)

[13] King, A.J., Rockafellar, R.T.: Asymptotic theory for solutions in statistical estimation and stochastic programming. Mathematics of Operations Research **18**(1), 148–162 (1993)

[14] Kleywegt, A.J., Shapiro, A., Homem-de Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization **12**(2), 479–502 (2002)

[15] Kronqvist, J., Bernal, D.E., Lundell, A., Grossmann, I.E.: A review and comparison of solvers for convex minlp. Optimization and Engineering **20**(2), 397–455 (2019)

[16] Li, C., Grossmann, I.E.: A finite \ϵ -convergence algorithm for two-stage stochastic convex nonlinear programs with mixed-binary first and second-stage variables. Journal of Global Optimization pp. 1–27

[17] Li, C., Grossmann, I.E.: An improved l-shaped method for two-stage convex 0-1 mixed integer nonlinear stochastic programs. Computers & Chemical Engineering **112**, 165–179 (2018)

[18] Li, C., Grossmann, I.E.: A generalized benders decomposition-based branch and cut algorithm for two-stage stochastic programs with nonconvex constraints and mixed-binary first and second stage variables. Journal of Global Optimization pp. 1–26 (2019)

[19] Li, X., Armagan, E., Tomasgard, A., Barton, P.I.: Stochastic pooling problem for natural gas production network design and operation under uncertainty. AIChE Journal **57**(8), 2120–2135 (2011)

[20] Li, X., Tomasgard, A., Barton, P.I.: Nonconvex generalized benders decomposition for stochastic separable mixed-integer nonlinear programs. Journal of optimization theory and applications **151**(3), 425 (2011)

[21] Li, X., Tomasgard, A., Barton, P.I.: Decomposition strategy for the stochastic pooling problem. Journal of Global Optimization **54**(4), 765–790 (2012)

[22] Linderoth, J., Shapiro, A., Wright, S.: The empirical behavior of sampling methods for stochastic programming. Annals of Operations Research **142**(1), 215–241 (2006)

[23] Mak, W.K., Morton, D.P., Wood, R.K.: Monte carlo bounding techniques for determining solution quality in stochastic programs. Operations Research Letters **24**(1-2), 47–56 (1999)

[24] Mijangos, E.: An algorithm for two-stage stochastic mixed-integer nonlinear convex problems. Annals of Operations Research **235**(1), 581–598 (2015)

[25] Misener, R., Thompson, J.P., Floudas, C.A.: Apogee: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. Computers & Chemical Engineering **35**(5), 876–892 (2011)

[26] Norkin, V.I., Pflug, G.C., Ruszczyński, A.: A branch and bound method for stochastic global optimization. Mathematical programming **83**(1-3), 425–450 (1998)

[27] Ogbe, E., Li, X.: A joint decomposition method for global optimization of multiscenario nonconvex mixed-integer nonlinear programs. Journal of Global Optimization pp. 1–35 (2018)

[28] Shapiro, A.: Asymptotic analysis of stochastic programs. Annals of Operations Research **30**(1), 169–186 (1991)

[29] Shapiro, A.: Asymptotic behavior of optimal solutions in stochastic programming. Mathematics of Operations Research **18**(4), 829–845 (1993)

[30] Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on stochastic programming: modeling and theory. SIAM (2009)

[31] Watson, J.P., Woodruff, D.L., Hart, W.E.: Pysp: modeling and solving stochastic programs in python. Mathematical Programming Computation **4**(2), 109–149 (2012)

[32] Wei, J., Realff, M.J.: Sample average approximation methods for stochastic minlps. Computers & Chemical Engineering **28**(3), 333–346 (2004)

Algorithm 2

Initialization: Iteration counter $k = 1$; upper bound $UB = +\infty$.
Generate M_l i.i.d. batches of samples with size N_l, one batch of samples with size N'_l, M_u i.i.d. batches of samples with size N_u and one batch of samples with size N'_u of $\theta \sim \mathbb{P}$

Step 1

for $m = 1$ to M_l **do**

Convexify the (SAA-nonconvex-MINLP) with the mth batch of samples of size N_l and generate (SAA-OA-MILP).

end for

Step 2

for $m = 1$ to M_l **do**

Solve the (SAA-OA-MILP) with the mth batch of samples. Denote the objective value as $\hat{w}_{N_l}^{(m),k}$ and the optimal binary variable value as $\bar{y}^{(m),k}$

end for

Step 3 Compute the mean and variance of the solutions:

$$\bar{w}_{N_l,M_l}^k = \frac{1}{M_l}\sum_{m=1}^{M_l} \hat{w}_{N_l}^{(m),k} \text{ and } \frac{(\hat{S}_{M_l}^{w,k})^2}{M_l} = \frac{1}{M_l(M_l-1)}\sum_{m=1}^{M_l}(\hat{w}_{N_l}^{(m),k} - \bar{w}_{N_l,M_l}^k)^2$$

Let $\bar{y}^k$ be the most common integer solution among all the $\bar{y}^{(m),k}$, $m \in [M_l]$.

Step 4

for $i = 1$ to N'_l **do**

Convexify (SAA-MINLP) with the ith single sample. Generate (single-OA-LP).

end for

Step 5 For all the following N'_l problems, fix $x := \tilde{x}$ ($\tilde{x}$ can be any feasible solution or a solution of any batch of (SAA-OA-MILP) at iteration k where the optimal solution $\bar{y}^k$ is found).

for $i = 1$ to N'_l **do**

Solve (single-OA-LP), with a single sample θ_i. Denote the objective value as $\hat{w}_1^{(i),k}$

end for

Compute the mean and variance of the objective values:

$$\bar{w}_{N'_l}^k = \frac{1}{N'_l}\sum_{i=1}^{N'_l} \hat{w}_1^{(i),k} \text{ and } \frac{(\hat{S}_{N'_l}^{w,k})^2}{N'_l} = \frac{1}{N'_l(N'_l-1)}\sum_{i=1}^{N'_l}(\hat{w}_1^{(i),k} - \bar{w}_{N'_l}^k)^2$$

Step 6 Compute the confidence interval of the lower bound

$$(\underline{LB}, \overline{LB}) = \left(\bar{w}_{N_l,M_l}^k - t_{M_l-1}^{\alpha/2}\frac{\hat{S}_{N_l,M_l}^{w,k}}{\sqrt{M_l}}, \quad \bar{w}_{N'_l}^k + t_{N'_l-1}^{\alpha/2}\frac{\hat{S}_{N'_l}^{w,k}}{\sqrt{N'_l}}\right)$$

Update the sample size with some update policy. Go back to step 1 if necessary.If $\underline{LB} \geq \overline{UB} - \epsilon$, then go to step 11; otherwise go to step 7.

Step 7 Fix $y := \bar{y}^k$. **for** $m = 1$ to M_u **do** Solve (SAA-nonconvex-NLP) to global optimality. Denote objective value as $\hat{u}_{N_u}^{(m),k}$ **end for** Compute the mean and variance of the lower estimator of the upper bound,

$$\bar{u}^k_{N_u,M_u} = \frac{1}{M_u}\sum_{m=1}^{M_u}\hat{u}_{N_u}^{(m),k}, \quad \frac{(\hat{S}^{u,k}_{M_u})^2}{M_u} = \frac{1}{M_u(M_u-1)}\sum_{m=1}^{M_u}(\hat{u}_{N_u}^{(m),k} - \bar{u}^k_{N_u,M_u})^2$$

Step 8 For all the following N'_u problems, fix $x := \tilde{x}^k$, given by heuristics.
for $i = 1$ to N'_u **do**
Solve the single-nonconvex-NLP. Denote the objective value as $\hat{u}_1^{(i),k}$
end for
Compute the mean and variance of the upper estimator of the upper bound,

$$\bar{u}^k_{N'_u} = \frac{1}{N'}\sum_{i=1}^{N'_u}\hat{u}_1^{(i),k}, \quad \frac{(\hat{S}^{u,k}_{N'_u})^2}{N'_u} = \frac{1}{N'_u(N'_u-1)}\sum_{i=1}^{N'_u}\left(\hat{u}_1^{(i),k} - \bar{u}^k_{N'_u}\right)^2$$

Step 9 Compute the confidence interval of the optimality gap for the upper bound

$$\left(\bar{u}^k_{N_u,M_u} - t^{\alpha/2}_{M_u-1}\hat{S}^{u,k}_{M_u}, \quad \bar{u}^k_{N'_u} + t^{\alpha/2}_{N'_u-1}\hat{S}^{u,k}_{N'_u}\right)$$

Update the sample size with some update policy. Go back to step 1 or step 7 if necessary. Otherwise if $\bar{u}^k_{N'_u} < UB - \epsilon$, then let $UB = \bar{u}^k_{N'_u}$, $\underline{UB} = \bar{u}^k_{N_u,M_u} - t^{\alpha/2}_{M_u-1}\hat{S}^{u,k}_{M_u}$, $\overline{UB} = \bar{u}^k_{N'_u} + t^{\alpha/2}_{N'_u-1}\hat{S}^{u,k}_{N'_u}$, $y^*_p = \bar{y}^k$, $x^*_p = \tilde{x}^k$. Add 'no-good' cut corresponding to $\bar{y}^k$ to (SAA-OA-MILP).
If $\underline{LB} \geq \overline{UB} - \epsilon$, then go to step 11; otherwise go to step 10.
Step 10 Let $k = k + 1$ and return to step 2
Step 11 Stop, the optimal solution is y^*_p, x^*_p and the optimal objective value is UB.