

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

Can Li · David E. Bernal · Kevin C. Furman ·
Marco A. Duran[†] · Ignacio E. Grossmann

Received: date / Accepted: date

Abstract 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. Previous work has considered this approach for convex two-stage SP [37]. 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. Since the theoretical sample size estimate is too conservative in practice, 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.

Keywords Stochastic programming · Sample Average Approximation · Mixed-integer Nonlinear Programming · Outer-Approximation

Can Li, David E. Bernal, Ignacio E. Grossmann
Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA E-mail: {canli,bernalde,grossmann}@cmu.edu

Kevin C. Furman
ExxonMobil Upstream Research Company, 22777 Springwoods Village Parkway, Spring, TX 77389 E-mail: kevin.c.furman@exxonmobil.com

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 [26], batch plant design [18], and water network synthesis [11]. Although there have been significant advances to solve deterministic MINLPs [16], 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 [33]. 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 [25] 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 [18]. Li and Grossmann further propose a generalized Benders decomposition-based branch and bound algorithm [17] with finite ϵ -convergence for convex stochastic MINLPs with mixed-binary first and second stage variables.

Among the different applications, engineering design problems can be posed as mathematical programs with uncertain parameters, with possible nonlinearity in the equations, and requirements of discreteness in certain design choices. The framework of stochastic mixed-integer nonlinear programs fits well to solve these types of problems [29].

For nonconvex stochastic MINLP, where the nonlinear functions in the stochastic MINLPs can be nonconvex, Li et al. [21] 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 [28] propose a joint decomposition algorithm. A perfect information-based branch and bound algorithm that solves nonseparable nonconvex stochastic MINLPs to global optimality is proposed by Cao and Zavala [5]. Kannan and Barton [10] 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 non-convex stochastic MINLPs with mixed-binary first and second stage variables was proposed in [19].

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) [15, 33]. SAA approximates the ‘true’ value of SP with the sample average of N i.i.d. (independent and identically distributed) 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. [24] prove bounding properties for external sampling. Kleywegt et al. [15] study a Monte Carlo simulation-based approach for SP with only discrete variables in the first stage. Linderoth et al. [23] analyze the quality of solutions obtained from sample-average approximations to two-stage stochastic linear programs in a computational study.

‘Internal sampling’ means that the sampling is done within the solution procedure. Norikin et al. [27] propose a stochastic branch and bound method for solving stochastic global optimization problems. Higle and Sen [9] 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 [37] 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 continuous probability distribution \mathbb{P} using SAA. Preliminary results of this work have been presented in technical conferences [7, 36], where progress was made for two-stage stochastic nonconvex MINLP 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. [12]. The contributions of this paper are as follows,

1. An internal sampling-based outer-approximation (OA) algorithm for two-stage stochastic nonconvex MINLP with continuous distribution is proposed.
2. The proposed algorithm is proved to converge to the ‘true’ value and optimal solution of the stochastic program (SP) as the sample sizes in all the internal sampling steps tend to infinity.
3. Sample size estimates and convergence rates for the proposed algorithm are provided.
4. We propose an efficient way to implement the sample average-based OA algorithm using confidence interval estimates.
5. Two policies are proposed 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 describe the steps of the SAAOA algorithm. In section 4, we prove that the proposed algorithm converges as the sample size tends 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 conclusions in section 7.

2 Problem Statement

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

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

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

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 recourse variables, which can vary depending on the realization of uncertain parameters, which are represented by the vector θ . The uncertain parameter $\theta \in \Theta$ follows a continuous joint probability distribution \mathbb{P} , where Θ is the support of the uncertain parameter. 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.*

With this assumption, Θ is usually a bounded set in practice. 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) [33] in the SP literature.

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

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

where $[N]$ represents the set $\{1, 2, \dots, N\}$, from which we obtain θ_i , $i \in [N]$, that are the N i.i.d. samples of the uncertain parameter vector θ , and z_i is the stage 2 variable corresponding to each 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 widely studied [1, 33]. 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) to the original MINLP’s 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 yields an upper bound (UB) of the optimal objective function value.

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

$$\sum_{j=1}^m a_j^{k'} y_j \leq b^{k'}, \text{ where } a_j^{k'} = \begin{cases} 1 & \text{if } y_j^{k'} = 1 \\ -1 & \text{if } y_j^{k'} = 0 \end{cases} \quad \text{and } b^{k'} = \sum_{j=1}^m y_j^{k'} - 1, \forall k' < k \quad (3)$$

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

The deterministic nonconvex OA algorithm can be extended to the solution of (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

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

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

where \hat{f} and \hat{g} are polyhedral relaxations for function f , and g , respectively, which can be obtained by convexifying (SP-MINLP). 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 i th 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.

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

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

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

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

$$\text{s.t.} \quad g(x, \bar{y}, z, \theta) \leq 0 \quad \forall \theta \in \Theta \quad (6b)$$

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

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

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

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 are described in Algorithm 1.

4 Convergence results

4.1 Convergence of objective values and solutions

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

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 vector of the values for the integer variables 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.

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

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

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 $\lim_{N \rightarrow \infty} \hat{u}_N(\bar{y}) = UB(\bar{y})$ and the event $\{S^*(\bar{y}) = \hat{S}_N(\bar{y})\}$ happens w.p. 1 as $N \rightarrow \infty$.

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

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

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

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

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

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

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 $N \rightarrow \infty$. Moreover, under some mild conditions ([14, 31, 32, 33]), 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. [33].

Assumption 2 All the samples θ_i , $i \in [N]$, 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 $\bar{y} \in \{0, 1\}^m$.

Remark 1 Note that for a given uncertain parameter θ_i , the cost-to-go function is discontinuous in general. However, $UB(\bar{y}, x)$ is the expected cost over a continuous distribution. Under some mild assumptions, $UB(\bar{y}, x)$ is continuous for all $x \in X$ [30].

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

Under assumptions 3-7, the following lemma holds.

Lemma 1 If Assumptions 3-7 hold true, $\lim_{N \rightarrow \infty} \hat{u}_N(\bar{y}) = UB(\bar{y})$ and the event $\{S^*(\bar{y}) = \hat{S}_N(\bar{y})\}$ happens w.p. 1 as $N \rightarrow \infty$, for any vector $\bar{y} \in \{0, 1\}^m$ (Theorem 5.3 of Shapiro et al. [33]).

Proof According to Assumption 7,

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

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

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

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

Now we need to prove that the event $\{S^*(\bar{y}) = \hat{S}_N(\bar{y})\}$ happens w.p. 1 as $N \rightarrow \infty$, 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 $\operatorname{dist}(\hat{x}_N(\bar{y}), S^*(\bar{y})) \geq \epsilon$, 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 $UB(\bar{y}, x^*(\bar{y})) > UB(\bar{y})$.

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

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 $\lim_{N \rightarrow \infty} \hat{u}_N(\bar{y}, \hat{x}_N(\bar{y})) = UB(\bar{y}, x^*(\bar{y})) > UB(\bar{y})$, which contradicts with the convergence of the objective value $\hat{u}_N(\bar{y})$ to $UB(\bar{y})$. \square

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

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

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

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

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

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

$$\text{s.t. } \hat{g}(x, y, z_i, \theta_i) \leq 0 \quad \forall i \in [N]; \quad \text{Eq. (3)} \quad (11b)$$

The optimal values of (SP-OA-MILP) and (SAA-OA-MILP) are defined as LB^k , \hat{w}_N^k , respectively. The sets of optimal solutions for (SP-OA-MILP) and (SAA-OA-MILP) at iteration k are denoted as T_k^* , \hat{T}_k^N , respectively. We want to prove that $\lim_{N \rightarrow \infty} \hat{w}_N^k = LB^k$ and the event $\{T_k^* = \hat{T}_k^N\}$ happens w.p. 1 as $N \rightarrow \infty$. 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 obtained by 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 proof is similar to Lemma 1, and hence we omit it.

Lemma 2 *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 $N \rightarrow \infty$, i.e., $\lim_{N \rightarrow \infty} \hat{w}_N^k = LB^k$, the event $\{T_k^* = \hat{T}_k^N\}$ happens w.p. 1 as $N \rightarrow \infty$.*

4.1.3 Convergence of optimal objective values and solutions of Algorithm 1

The convergence of the optimal values and solutions of the LB and the UB is proved setting up the main theorem of this section.

Theorem 1 *The proposed Algorithm 1 returns the optimal value and the set of optimal solutions (x^*, y^*) of (SP-MINLP) w.p. 1 as $N \rightarrow \infty$.*

Proof Note that 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 $N \rightarrow \infty$. Note that Lemmas 1 and 2 prove 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 Algorithm 1 finite number of iterations to converge. The optimal value returned is the optimal value of the true (SP-MINLP) and the optimal solution corresponds to the y^* that yields the best UB. \square

4.2 Estimates of sample sizes

Having 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

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

where D is the diameter of set X , i.e., $D = \sup_{x \in X, x' \in X} \|x - x'\|$, 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 [15]).

Remark 2 Eq. (12) gives the rate of convergence for any SAA estimators of any two-stage SP. Theoretically, it can be used to 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 [15]. Therefore, we need to design an algorithm based on Algorithm 1 that is more efficient in practice. Note that the convergence results and the sample size estimates only apply to Algorithm 1. The algorithm that will be discussed in the next section is only a ‘practical’ algorithm to solve the (SP-MINLP) problem.

5 Algorithm Design

Here we extend Algorithm 1 to a more practical algorithm based on the ideas of Kleywegt et al. [15] where an empirical method is proposed to construct

confidence intervals for the optimal objective value of the ‘true’ SP problem. Throughout this section, we assume that all the problems are solved to global optimality.

For a finite sample size N , Mak et al. [24] prove that the expectation of the SAA problems provides lower bounds for the corresponding true SP problems. In Algorithm 1, the expectation of the objective of (SAA-OA-MILP), \hat{w}_N^k , provides a lower bound for the objective of (SP-OA-MILP), LB^k , at every iteration k ,

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

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,

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

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. [24] 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 , $i = 1, \dots, N'_L$

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

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

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'_L samples of θ_i . The values for (\bar{x}, \bar{y}) can come from the optimal solution of (SAA-nonconvex-NLP) (\bar{x}^k, \bar{y}^k) . The i th single size nonconvex NLP at iteration k is defined as,

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

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

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

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

Note that the major difference between (13) and (14) is that in (13) polyhedral relaxations of functions f and g are used.

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 2 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., $\mathbb{E}[\hat{u}_N(\bar{y})] \leq UB(\bar{y})$. 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) and (SAA-OA-MILP) problems are independent and identically distributed.*

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 $\hat{u}_{N_U}^{(m),k}$ to denote the optimal objective value of the m th batch of (SAA-nonconvex-NLP) at iteration k . From Mak et al. [24], random variable $\hat{u}_{N_U}^{(m),k}$ is a biased estimator of $UB(\bar{y}^k)$, i.e.,

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

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

$$\sqrt{M_U} \left(\bar{u}_{N_U, M_U}^k - \mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\bar{u}_{N_U, M_U}^k \right] \right) \Rightarrow \mathcal{N}(0, \sigma_{u,k}^2) \quad (16)$$

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

$$\frac{(\hat{S}_{M_U}^{u,k})^2}{M_U} = \frac{1}{M_U(M_U - 1)} \sum_{m=1}^{M_U} (\hat{u}_{N_U}^{(m),k} - \bar{u}_{N_U, M_U}^k)^2 \quad (17)$$

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

$$\left(\bar{u}_{N_U, M_U}^k - t_{M_U-1}^{\alpha/2} \frac{\hat{S}_{M_U}^{u,k}}{\sqrt{M_U}}, \quad \bar{u}_{N_U, M_U}^k + t_{M_U-1}^{\alpha/2} \frac{\hat{S}_{M_U}^{u,k}}{\sqrt{M_U}} \right) \quad (18)$$

where $t_{M_U-1}^{\alpha/2}$ 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 obtain the upper estimator of (SP-nonconvex-NLP). We use $\hat{u}_1^{(i),k}$ to denote the optimal objective value of the i th sample of (single-nonconvex-NLP) at iteration k . By definition, we also have the mean of the N'_U samples, $\bar{u}_{N'_U}^k = \frac{1}{N'_U} \sum_{i=1}^{N'_U} \hat{u}_1^{(i),k}$. From Mak et al. [24], we have

$$\mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\bar{u}_{N'_U}^k \right] \geq UB(\bar{y}^k) \quad (19)$$

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

$$\sqrt{N'_U} \left(\bar{u}_{N'_U}^k - \mathbb{E}_{\theta_i \sim \mathbb{P}} \left[\bar{u}_{N'_U}^k \right] \right) \Rightarrow N(0, \sigma_{u',k}^2) \quad (20)$$

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

$$\frac{(\hat{S}_{N'_U}^{u,k})^2}{N'_U} = \frac{1}{N'_U(N'_U - 1)} \sum_{i=1}^{N'_U} (\hat{u}_1^{(i),k} - \bar{u}_{N'_U}^k)^2 \quad (21)$$

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

$$\left(\bar{u}_{N'_U}^k - t_{N'_U-1}^{\alpha/2} \frac{\hat{S}_{N'_U}^{u,k}}{\sqrt{N'_U}}, \quad \bar{u}_{N'_U}^k + t_{N'_U-1}^{\alpha/2} \frac{\hat{S}_{N'_U}^{u,k}}{\sqrt{N'_U}} \right) \quad (22)$$

where $t_{N'_U-1}^{\alpha/2}$ 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,

$$(UB, \overline{UB}) = \left(\bar{u}_{N_U, M_U}^k - t_{M_U-1}^{\alpha/2} \frac{\hat{S}_{M_U}^{u,k}}{\sqrt{M_U}}, \quad \bar{u}_{N'_U}^k + t_{N'_U-1}^{\alpha/2} \frac{\hat{S}_{N'_U}^{u,k}}{\sqrt{N'_U}} \right) \quad (23)$$

where $(\hat{S}_{N'_U}^{u,k})^2$ is the standard sample variance estimator of $\sigma_{u',k}^2$, the variance of the scaled normal distribution to what $\bar{u}_{N'_U}^k$ 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 m th batch is denoted as $\hat{w}_{N_L}^{(m),k}$. We compute the mean and variance of the M_L batches by,

$$\bar{w}_{N_L, M_L}^k = \frac{1}{M_L} \sum_{m=1}^{M_L} \hat{w}_{N_L}^{(m),k} \text{ and}$$

$$\frac{(\hat{S}_{M_L}^{w,k})^2}{M_L} = \frac{1}{M_L(M_L-1)} \sum_{m=1}^{M_L} (\hat{w}_{N_L}^{(m),k} - \bar{w}_{N_L, M_L}^k)^2$$

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

$$\left(\bar{w}_{N_L, M_L}^k - t_{M_L-1}^{\alpha/2} \frac{\hat{S}_{M_L}^{w,k}}{\sqrt{M_L}}, \quad \bar{w}_{N_L, M_L}^k + t_{M_L-1}^{\alpha/2} \frac{\hat{S}_{M_L}^{w,k}}{\sqrt{M_L}} \right) \quad (24)$$

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 i th (single-OA-LP) is denoted as $\hat{w}_1^{(i),k}$. We can compute the mean and variance of the objective values with,

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

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

$$\left(\bar{w}_{N'_L}^k - t_{N'_L-1}^{\alpha/2} \frac{\hat{S}_{N'_L}^{w,k}}{\sqrt{N'_L}}, \quad \bar{w}_{N'_L}^k + t_{N'_L-1}^{\alpha/2} \frac{\hat{S}_{N'_L}^{w,k}}{\sqrt{N'_L}} \right) \quad (25)$$

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

$$(\underline{LB}, \overline{LB}) = \left(\bar{w}_{N_L, M_L}^k - t_{M_L-1}^{\alpha/2} \frac{\hat{S}_{N_L, M_L}^{w,k}}{\sqrt{M_L}}, \quad \bar{w}_{N'_L}^k + t_{N'_L-1}^{\alpha/2} \frac{\hat{S}_{N'_L}^{w,k}}{\sqrt{N'_L}} \right) \quad (26)$$

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. A high-level overview of the SAAOA with confidence intervals is shown in Figure 1. At the initialization step, (SAA-MINLP) problems are convexified to generate (SAA-OA-MILP) and (single-OA-LP) problems. 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 (if $\underline{LB} \geq \overline{UB} - \epsilon$). If not, sample sizes might need to be updated and the algorithm keeps iterating. The steps of the SAAOA method with confidence interval estimators are described in Algorithm 2.

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 m th batch of samples of size N_L and
 generate (SAA-OA-MILP).
end for

Step 2
for $i = 1$ to N'_L **do**
 Convexify (SAA-MINLP) with the i th single sample. Generate a problem to be solved
 in Step 5 as (single-OA-LP).
end for

Step 3
for $m = 1$ to M_L **do**
 Solve the (SAA-OA-MILP) with the m th 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 4 Compute the mean and variance of the solutions:

$$\bar{w}_{N_L, M_L}^k = \frac{1}{M_L} \sum_{m=1}^{M_L} \hat{w}_{N_L}^{(m),k} \quad \text{and} \quad \frac{(\hat{S}_{M_L}^{w,k})^2}{M_L} = \frac{1}{M_L(M_L-1)} \sum_{m=1}^{M_L} (\hat{w}_{N_L}^{(m),k} - \bar{w}_{N_L, M_L}^k)^2$$
Let \bar{y}^k be the most common integer solution among all the $\bar{y}^{(m),k}$, $m \in [M_L]$.

Step 5 For all the following N'_L problems, fix $x := \bar{x}$ (\bar{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:

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

Step 6 Compute the $(1 - \alpha)$ confidence interval of the lower bound

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

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 these parameters are not unique. We discuss the update policies in the next subsection.

5.4 Update policies

As we have discussed, estimating the proper sample sizes 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 become too conservative and increase the

Let $LB = \bar{w}_{N'}^k, \underline{LB} = \bar{w}_{N_L, M_L}^k - t_{M_L-1}^{\alpha/2} \frac{\hat{S}_{N_L, M_L}^{w, k}}{\sqrt{M_L}}, \overline{LB} = \bar{w}_{N'_L}^k + t_{N'_L-1}^{\alpha/2} \frac{\hat{S}_{N'_L}^{w, k}}{\sqrt{N'_L}}$.

If $LB \geq \overline{UB} - \epsilon$, 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,

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

Step 8 For all the following N'_U problems, fix $x := \bar{x}^k$, given by heuristics, e.g., take the mean value of the optimal solutions in (SAA-nonconvex-NLP).

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,

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

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

$$(\underline{UB}, \overline{UB}) = \left(\bar{u}_{N_U, M_U}^k - t_{M_U-1}^{\alpha/2} \frac{\hat{S}_{M_U}^{u, k}}{\sqrt{M_U}}, \quad \bar{u}_{N'_U}^k + t_{N'_U-1}^{\alpha/2} \frac{\hat{S}_{N'_U}^{u, k}}{\sqrt{N'_U}} \right)$$

Update the sample size with policy P1 or P2 described in subsection 5.4. Go to Step 1, Step 2, Step 7, Step 8, or Step 10 depending on the chosen policy. Otherwise if $\bar{u}_{N'_U}^k < \underline{UB} - \epsilon$,

then let $UB = \bar{u}_{N'_U}^k, \underline{UB} = \bar{u}_{N_U, M_U}^k - t_{M_U-1}^{\alpha/2} \frac{\hat{S}_{M_U}^{u, k}}{\sqrt{M_U}}, \overline{UB} = \bar{u}_{N'_U}^k + t_{N'_U-1}^{\alpha/2} \frac{\hat{S}_{N'_U}^{u, k}}{\sqrt{N'_U}}, x_p^* = \bar{x}^k$.

Add 'no-good' cut corresponding to \bar{y}^k to (SAA-OA-MILP).

If $LB \geq \overline{UB} - \epsilon$, 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 .

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, N'_U or N'_L single sample size problems are solved. The numbers N'_U, N'_L affect the accuracy of the estimates of the feasible solutions (\bar{x}, \bar{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 sizes of the batches 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 the sample sizes 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. [24].

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

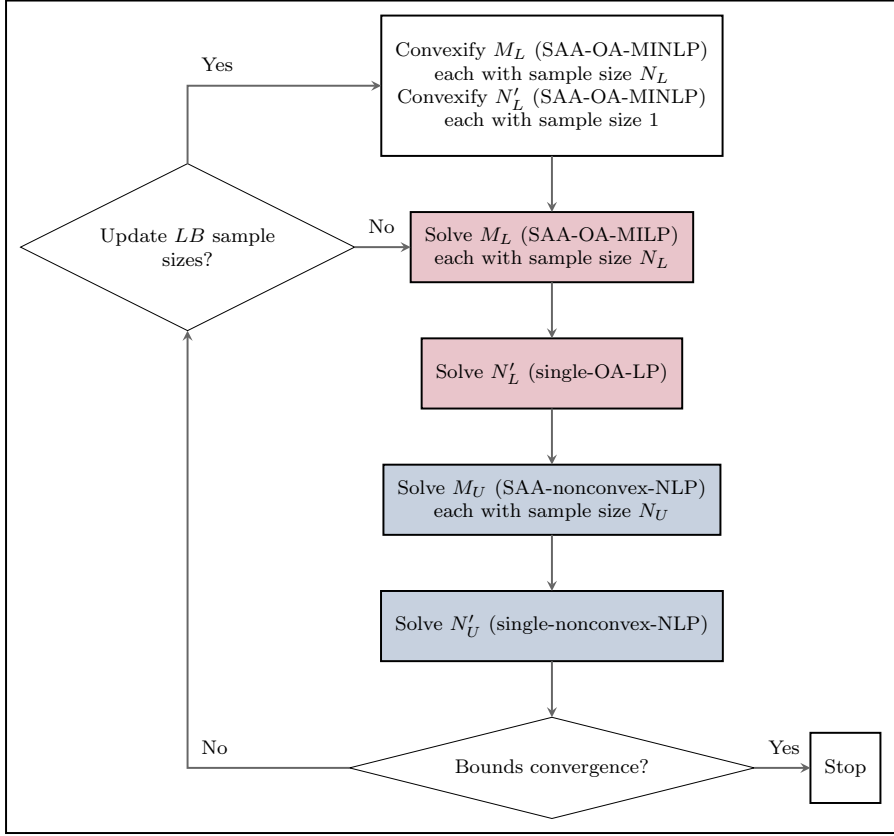


Fig. 1 Flowchart of the SAAOA algorithm with confidence interval estimators. Boxes filled in red show the MILP phase of the algorithm, boxes filled in blue show the NLP phase of the algorithm

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 a 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 at Step 9 of the algorithm. A confidence interval is considered tight if the relative gap between its upper and lower estimators is less than a certain threshold, e.g., 5%. If it is not tight, multiply the number of samples and batches 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. Four cases can happen when the confidence intervals are checked at Step 9.

- The confidence intervals for both the lower bound and the upper bound are tight at iteration k . Go to Step 10.
- The confidence interval for the lower bound is tight. The confidence interval for the upper bound is not tight. Increase sample size N_U , M_U if $t_{M_U-1}^{\alpha/2} \frac{\hat{S}_{M_U}^{u,k}}{\sqrt{M_U}}$ is large. Increase sample size N'_U if $t_{N'_U-1}^{\alpha/2} \frac{\hat{S}_{N'_U}^{u,k}}{\sqrt{N'_U}}$ is large. If both cases are true or only the former case is true, go back to Step 7. Otherwise, go back to Step 8.
- The confidence interval for the upper bound is tight. The confidence interval for the lower bound is not tight. Increase sample size M_L , N_L if $t_{M_L-1}^{\alpha/2} \frac{\hat{S}_{N_L, M_L}^{w,k}}{\sqrt{M_L}}$ is large. Increase sample size N'_L if $t_{N'_L-1}^{\alpha/2} \frac{\hat{S}_{N'_L}^{w,k}}{\sqrt{N'_L}}$ is large. If both cases are true or only the former case is true, go back to Step 1. Otherwise, go back to Step 2.
- Both of the confidence intervals are loose. Update the sample sizes as in the two former cases. Go back to Step 1 or Step 2.

Note that the ‘no-good’ cut is only added at Step 9 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 Similar to P1, P2 also checks the tightness of the confidence intervals at Step 9 of the algorithm. However, P2 does 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 relies on 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 \overline{UB} is less than the \underline{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 one keeps 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 make the following remark to end this section.

Remark 3 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, because fixing the first stage binary variables leads to fully decomposable (single-nonconvex-NLP) subproblems. Therefore, we only need to solve (single-nonconvex-NLP), potentially in parallel to obtain the confidence interval for the upper bound, which is given by equation (22).

6 Computational Results

Algorithm 2 with the two proposed update policies is implemented in Pyomo / Python [8] as a package named `saaoa.py`, which takes a two-stage model in the data structure of PySP [35].

The convex relaxation for (SAA-MINLP) is obtained using a special version of BARON v.19.4.4 [13] which provides the root node polyhedral relaxation. (SAA-OA-MILP) and (single-OA-LP) are solved using CPLEX v.12.9. The (SAA-nonconvex-NLP) (single-nonconvex-NLP) problems are solved with BARON v.19.3.24. All the problems in this paper are solved using one processor of an Intel Xeon (2.67GHz) machine with 64 GB RAM.

6.1 Illustrative example: a process network problem

To show how Algorithm 2 works, we present a small process network problem involving concave investment cost functions. Consider producing a chemical C which can be manufactured with either process 2 or process 3, both of which use chemical B as raw material. B can be purchased from another company and/or manufactured with process 1 which uses A as a raw material. The demand for chemical C, denoted as d , is the source of uncertainty. The superstructure of the process network is shown in Figure 2.

The problem is formulated as a two-stage stochastic program. The first stage decisions are investment decisions on the three processes, which include binary variables Y_i to denote whether process i is selected and continuous variables CAP_i to denote the capacity of process i for $i = 1, 2, 3$. The second stage decisions are the material flows. We denote the uncertainty scenarios as ω and explicitly state the dependency of the second stage decision on them by presenting them as functions of ω . Variables $PA(\omega)$, $PB(\omega)$ represent the purchase amount of chemical A and B, respectively. Other material flows for chemical B and C are shown in the superstructure in Figure 2. The MINLP formulation is shown as follows.

$$\begin{aligned} \min \quad & 10Y_1 + 15Y_2 + 20Y_3 + 25CAP_1^{0.6} + 8CAP_2^{0.6} + 10CAP_3^{0.6} \\ & + \mathbb{E}_{d(\omega) \sim \mathbb{P}} 5PA(\omega) + 20PB(\omega) - 30(C_2(\omega) + C_3(\omega)) \end{aligned} \quad (28a)$$

$$\text{s.t.} \quad CAP_1 \leq U \cdot Y_1, \quad CAP_2 \leq U \cdot Y_2, \quad CAP_3 \leq U \cdot Y_3, \quad Y_2 + Y_3 \leq 1 \quad (28b)$$

$$PA(\omega) \leq CAP_1, \quad B_2(\omega) \leq CAP_2, \quad B_3(\omega) \leq CAP_3 \quad \forall \omega \quad (28c)$$

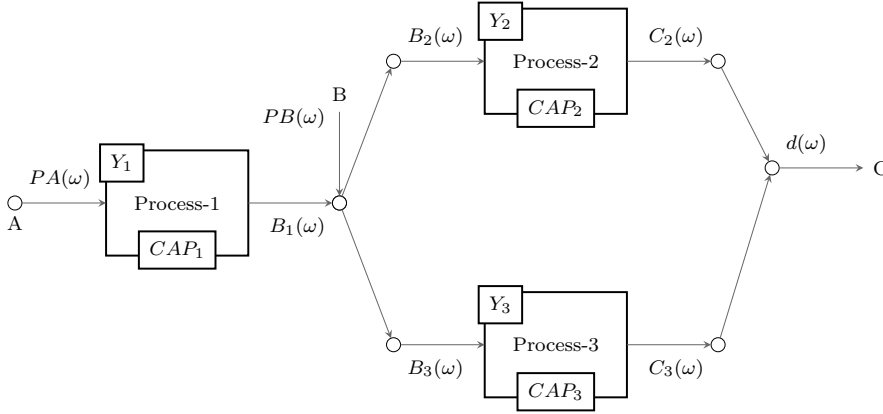


Fig. 2 Superstructure for the process network problem.

$$B_1(\omega) = \frac{25}{1 + e^{-PA(\omega)+13}}, \quad C_2(\omega) = 0.82B_2(\omega), \quad C_3(\omega) = 0.95B_3(\omega) \quad \forall \omega \quad (28d)$$

$$B_1(\omega) + PB(\omega) = B_2(\omega) + B_3(\omega) \quad \forall \omega \quad (28e)$$

$$C_2(\omega) + C_3(\omega) \leq d(\omega) \quad \forall \omega \quad (28f)$$

$$Y_i \in \{0, 1\} \quad \forall i \in \{1, 2, 3\} \quad (28g)$$

In stage 1, (28b) forces the capacity of any given process to zero if that process is not selected and only one of process 2 and 3 can be selected. The stage 2 equations are defined for each scenario ω . Equation (28c) limits the input to each process within the installed capacity. Equation (28d) describes the input-output relationship of each process. Note that the output of process 1, $B_1(\omega)$, is approximated by a sigmoidal function of the input, which is a nonlinear nonconvex function. Equation (28e) describes a material balance for chemical B. The production of chemical C is less than the demand $d(\omega)$. The objective is to maximize the expected profit. To be consistent with the notation of this paper, we denote the objective as minimizing the negative of the expected profit. Note that the variable investment cost is a concave function of the capacity of each process. The demand is assumed to follow a truncated Gaussian distribution $\mathcal{N}(10, 0.5)$ where the values less than 9 or greater 11 are truncated. To relate this problem with the problem statement in equation (1), $x = CAP_i$, $y = \{Y_i\}$, $\forall i \in \{1, 2, 3\}$. $z = \{PA, PB, B_1, B_2, B_3, C_2, C_3\}$, $\theta = d$

Algorithm 2 is applied to solve this small problem. We fix the sample sizes: $N_L = 200$, $M_L = 10$, $N'_L = 100$, $N_U = 50$, $M_U = 10$, $N'_U = 100$. First, 10 batches of (SAA-MINLP) each with sample size 200 are convexified to obtain 10 batches of (SAA-OA-MILP) problems. Each (SAA-MINLP) has 3 binary variables, 1,403 continuous variables, and 1,604 constraints. 100 batches of (SAA-MINLP) each with sample size 1 are convexified to be solved as (single-OA-LP) problems.

At iteration $k = 1$, 10 batches of (SAA-OA-MILP) each with sample size 200 are solved. We examine the optimal solutions of the 10 batches and found that in most cases processes 1 and 3 are installed. The binary variables are fixed at $Y_1 = 1$, $Y_2 = 0$, $Y_3 = 1$, and the capacities of processes 1 and 3 are fixed at the optimal capacities of one batch of (SAA-OA-MILP) where processes 1 and 3 are selected. The 95% confidence interval for the lower bound is $(-115.33, -110.36)$ calculated using equation (26). We proceed to the NLP phase of the algorithm with the binary variables fixed at $Y_1 = 1$, $Y_2 = 0$, $Y_3 = 1$. 10 batches of (SAA-nonconvex-NLP) each with sample size 50 are solved to obtain the lower estimator of the lower bound. 100 (single-nonconvex-NLP) problems are solved with the capacity of each process fixed at the mean of the optimal solutions of the 10 batches of (SAA-nonconvex-NLP). The 95% confidence interval for the upper bound is $(-50.59, -44.82)$ calculated by equation (23). Since the confidence intervals for the upper and the lower bound do not satisfy the termination criteria, $\underline{LB} \geq \overline{UB} - \epsilon$, we add an integer cut corresponding to the solution $Y_1 = 1$, $Y_2 = 0$, $Y_3 = 1$ to each batch of the (SAA-OA-MILP) problems and proceed to iteration 2.

At iteration $k = 2$, the confidence intervals for the lower bound becomes $(-102.1, -97.19)$ after the integer cut is added. The optimal binary variables found by (SAA-OA-MILP) problems are $Y_1 = 1$, $Y_2 = 1$, $Y_3 = 0$. The confidence interval for the upper bound corresponding to this binary variable combination is $(-58.07, -52.32)$. Since the new confidence interval is strictly lower than that at iteration 1, the best feasible solution is updated to $Y_1 = 1$, $Y_2 = 1$, $Y_3 = 0$. The termination criteria is still not satisfied. An integer cut corresponding to the solution $Y_1 = 1$, $Y_2 = 1$, $Y_3 = 0$ is added to each batch of the (SAA-OA-MILP) problems.

At iteration 3, the confidence interval for the lower bound is $(-41.31, -40.42)$. Since the lower bound is strictly greater than the upper bound, i.e., $\underline{LB} \geq \overline{UB} - \epsilon$, the algorithm terminates. We conclude that the optimal solution is $Y_1 = 1$, $Y_2 = 1$, $Y_3 = 0$ with a confidence interval of $(-58.07, -52.32)$.

6.2 Stochastic Pooling problem

The pooling problem arises in several engineering applications, such as petrochemical and agricultural industries. This nonconvex network flow optimization problem considers different flows that can be mixed and need to satisfy constraints concerning their quality attributes, which depend on their composition [34]. The stochastic pooling problem has been studied by Li et al. [20, 22]. The first stage decisions are investment decisions on sources, pools, and pipelines, which are represented as binary variables. The second stage decisions are the mass flows and the split fractions, which are represented as continuous variables that give rise to nonconvex bilinear terms. The constraints include mass balance, investment capacity, and quality specifications. The objective is to minimize the expected cost. This problem is an example of

an engineering design problem posed as a stochastic mixed-integer nonlinear problem.

6.2.1 Small pooling instance

We first consider a small pooling instance with 4 sources, 1 pool, 2 terminals, and 2 quality attributes. This instance has 16 binary variables and 26 constraints in the first stage; 21 continuous variables and 56 constraints per scenario in the second stage. The uncertainty is assumed to arise 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 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 Algorithm 2, 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%.

To compare with the SAAOA algorithm, we use BARON v.19.3.24 to solve 10 batches of (SAA-MINLP) problems each with a sample size of 50. Each of these MINLP problems has 1,066 variables and 2,826 constraints. 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. (24). 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.

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 estimators when the SAAOA algorithm terminates are shown in the table. Note that we allow the algorithm to terminate after the confidence interval of the LB is strictly greater than the confidence interval of the UB, i.e., $\underline{LB} \geq \overline{UB} - \epsilon$. 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

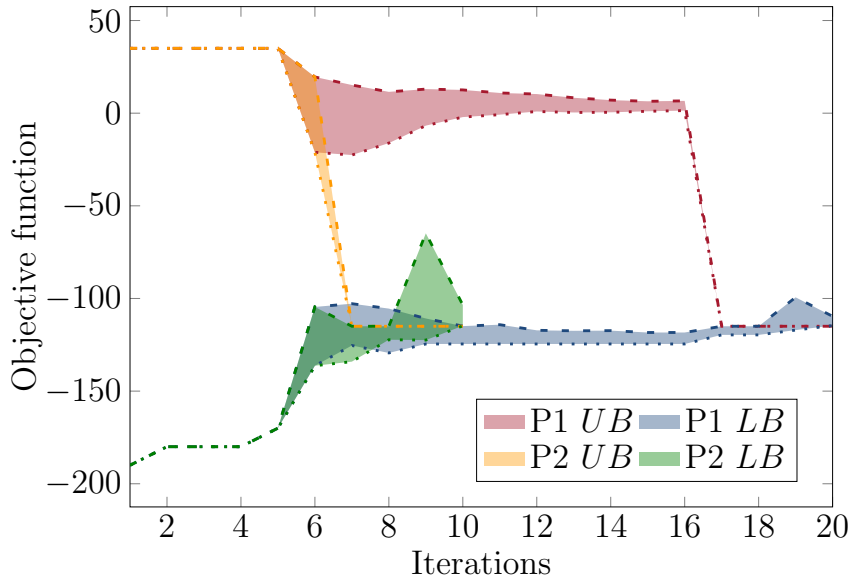


Fig. 3 Bounds convergence for test case with $\sigma = 0.004$, $\beta = 1.5$, $\gamma = 2$

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 algorithm needs 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 3. 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.

We also show a breakdown of the computational time of SAAOA Algorithm 2 in solving (SAA-OA-MILP), (single-OA-LP), (single-nonconvex-NLP), and

Table 1 Computational results for the small pooling instance

Variance	$\sigma = 0.0031$				$\sigma = 0.004$									
	P1 and P2		DE		P1		P2		DE					
Policy														
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	-
Iterations	9	9	9	9	-	22	24	20	20	21	22	10	12	-
Wall time (s)	784	675	584	529	100,412	19,845	16,376	16,708	11,094	13,098	14,364	7,177	10,190	100,471
(SAA-OA-MILP)	110	101	97	89	-	3,273	1,576	2,309	751	2,288	1,319	566	754	-
(single-OA-LP)	1	1	1	1	-	45	45	25	29	22	43	1	1	-
(single-nonconvex-NLP)	167	172	175	181	-	10,044	10,094	5,839	5,738	6,109	8,685	206	299	-
Convexification	496	390	300	248	-	5,946	4,086	8,098	4,219	4,207	3,772	6,384	9,108	-
\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
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

convexification. In most cases, the majority of computational time is spent on solving (single-nonconvex-NLP) and generating convex relaxations.

6.2.2 Large pooling instance

A large pooling instance with 5 sources, 2 pools, 4 terminals, and 7 quality attributes is solved. This instance has 24 binary variables and 39 constraints in the first stage; 89 continuous variables and 184 constraints per scenario in the second stage. The uncertainty is assumed to come from three different sources, the qualities of two feed streams and the upper bound of one terminal demand. The deviations of the feed qualities from the nominal value follow a truncated $\mathcal{N}(0, 0.02)$ distribution; the deviation from the upper bound of the demand follows a $\mathcal{N}(0, 3)$ distribution, where the parameter values that are more than two standard deviations away from the mean values are truncated.

Table 2 Computational results for the large pooling instance

Policy	P1		P2	
	β	β	β	β
Iterations	32	28	32	28
Wall time (s)	48,462	48,230	45,721	45,133
(SAA-OA-MILP)	1,950	1,731	1,963	1,750
(single-OA-LP)	9	7	7	8
(single-nonconvex-NLP)	45,958	45,959	43,223	42,837
Convexification	327	307	307	317
\overline{UB}	-400.5	-400.2	-399.8	-400.2
UB	-384.7	-385.8	-383.8	-385.8
\overline{LB}	-383.8	-385.5	-383.8	-385.5
LB	-373.7	-373.9	-370.6	-373.9
N_L	50	50	50	50
M_L	10	10	10	10
N'_L	112	200	75	100
N'_U	168	100	168	200

We apply Algorithm 2 to solve this problem using the same setting (initial sample sizes, probability of confidence intervals, relative gap, solvers, etc.) as in the small pooling instance. The computational results are shown in Table 2. Different values of the fixed ratio parameters β and γ are tested. In this case, the computational results do not depend on the value of γ , i.e., the value M_L is never updated in all the cases. Therefore, only the β values are reported in

Table 2. Compared with the small pooling instance, the computational time increases significantly due to the larger NLP subproblems. However, Algorithm 2 is still able to solve the problem within one day of computational time. P2 still takes less wall time than P2 but the difference is not significant. Most of the computational time is spent on (single-nonconvex-NLP) problems. The convexification time is small because the sizes of N_L have not increased.

Remark 4 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 one prefers to have a tight estimate for each solution so that there is more certainty that one solution is suboptimal, P1 is preferable.

7 Conclusions

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 tends 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 problem’s lower bound and upper bound, respectively. The sample sizes are updated dynamically using some update policies. 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 4.

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. Although the algorithm can be used to address stochastic nonconvex MINLPs with mixed-binary first stage variables in theory, it is only computationally efficient for stochastic nonconvex MINLPs with pure binary first stage variables. Therefore, another

possible future direction would be making the algorithm more efficient for problems with mixed-binary first stage variables.

Acknowledgements

The authors gratefully acknowledge financial support from ExxonMobil Corporation through the Center of Advanced Process Decision-making at Carnegie Mellon University. The authors would like to acknowledge Jing Wei for starting this project during her postdoc at ExxonMobil Corporate Strategic Research. The authors would like to thank Prof. Nick Sahinidis for providing the special version of BARON that generates the convex relaxations for the nonconvex MINLPs.

References

1. Ahmed, S., Shapiro, A., Shapiro, E.: The sample average approximation method for stochastic programs with integer recourse 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. Furman, K., Wei, J., Duran, M., Tawarmalani, Sahinidis, N.: Global optimization of nonconvex stochastic mixed-integer nonlinear programming (MINLP) problems. In: *INFORMS Annual Meeting (2005)*. San Francisco, CA
8. 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)
9. Hige, J.L., Sen, S.: Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research* **16**(3), 650–669 (1991)
10. Kannan, R.: Algorithms, analysis and software for the global optimization of two-stage stochastic programs. Ph.D. thesis, Massachusetts Institute of Technology (2018)

11. 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)
12. 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)
13. 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)
14. 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)
15. 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)
16. 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)
17. 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
18. 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)
19. 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)
20. 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)
21. 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)
22. Li, X., Tomasgard, A., Barton, P.I.: Decomposition strategy for the stochastic pooling problem. *Journal of Global Optimization* **54**(4), 765–790 (2012)
23. Linderoth, J., Shapiro, A., Wright, S.: The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* **142**(1), 215–241 (2006)
24. 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)
25. Mijangos, E.: An algorithm for two-stage stochastic mixed-integer nonlinear convex problems. *Annals of Operations Research* **235**(1), 581–598

- (2015)
26. 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)
 27. 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)
 28. 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)
 29. Popela, P.: Stochastic programming models for engineering design problems. *Engineering Mechanics* **17**(5-6), 351–362 (2010)
 30. Schultz, R.: On structure and stability in stochastic programs with random technology matrix and complete integer recourse. *Mathematical Programming* **70**(1-3), 73–89 (1995)
 31. Shapiro, A.: Asymptotic analysis of stochastic programs. *Annals of Operations Research* **30**(1), 169–186 (1991)
 32. Shapiro, A.: Asymptotic behavior of optimal solutions in stochastic programming. *Mathematics of Operations Research* **18**(4), 829–845 (1993)
 33. Shapiro, A., Dentcheva, D., Ruszczyński, A.: *Lectures on stochastic programming: modeling and theory*. SIAM (2009)
 34. Tawarmalani, M., Sahinidis, N.V.: *The Pooling Problem*, pp. 253–283. Springer US, Boston, MA (2002). DOI 10.1007/978-1-4757-3532-1_9
 35. 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)
 36. Wei, J., Furman, K., Duran, M., Tawarmalani, M., Sahinidis, N.: Global optimization for nonconvex stochastic mixed-integer nonlinear programs. In: *International Federation of Operational Research Societies Triennial Conference* (2005). Honolulu, HI
 37. Wei, J., Realff, M.J.: Sample average approximation methods for stochastic MINLPs. *Computers & Chemical Engineering* **28**(3), 333–346 (2004)