Efficient Solution Selection for Two-stage Stochastic Programs

Xin Fei*, Nalân Gülpınar, Jürgen Branke

Warwick Business School, The University of Warwick, Coventry CV4 7AL, The United Kingdom

Abstract

Sampling-based stochastic programs are extensively applied in practice. However, the resulting models tend to be computationally challenging. A reasonable number of samples needs to be identified to represent the random data, and a group of approximate models can then be constructed using such a number of samples. These approximate models can produce a set of potential solutions for the original model. In this paper, we consider the problem of allocating a finite computational budget among numerous potential solutions of a two-stage linear stochastic program, which aims to identify the best solution among potential ones by conducting simulation under a given computational budget. We propose a two-stage heuristic approach to solve the computational resource allocation problem. First, we utilise a Wasserstein-based screening rule to remove potentially inferior solutions from the simulation. Next, we use a ranking and selection technique to efficiently collect performance information of the remaining solutions. The performance of our approach is demonstrated through well-known benchmark problems. Results show that our method provides good trade-offs between computational effort and solution performance.

Keywords: Stochastic Programming, Sample Average Approximation, Wasserstein Metric, Ranking and Selection

1. Introduction

Real-life optimisation problems often involve uncertainties and require solutions that can handle such uncertainties in the modelling process. Techniques such as the two-stage linear stochastic programming with recourse (SPR) incorporate random data within the model formulation and

*Corresponding author

Email addresses: phd14xf@mail.wbs.ac.uk (Xin Fei), Nalan.Gulpinar@wbs.ac.uk (Nalân Gülpınar), Juergen.Branke@wbs.ac.uk (Jürgen Branke)

determine a solution that satisfies the constraints and leads to the best expected objective function value for all possible scenarios. The development of SPR can be traced back to research conducted in the 1950s and 1960s, e.g., Beale (1955), Dantzig (1955) and Wets (1966). The successful applications of SPR can be found in various sectors such as portfolio management (Dupačová, 1999; Miller and Ruszczyński, 2011), energy planning (Beraldi et al., 2008; Zhou et al., 2013; Feng and Ryan, 2014), supply chain management (Joensson et al., 1993; Santoso et al., 2005; Dillon et al., 2017), and transportation planning (Cheung and Chen, 1998; Barbarosolu and Arda, 2004; Liu et al., 2009).

SPR problems can become computationally intractable in numerous applications because each possible sample generated from random data is associated with one or several decision variables and constraints within the model formulation. If the sample space is considerably large or continuous, then determining an optimal solution within a reasonable timeframe will be impossible for such a model. Studies have proposed the utilisation of sample average approximation (SAA) to identify approximate solutions to large-scale SPR problems, e.g., see Gürkan et al. (1994) and Shapiro and Homem-de Mello (1998). SAA leverages computational challenges in such a way that a subset of samples generated from random data is used to construct approximate models, which provide potential solutions to the original SPR model. Monte-Carlo sampling as well as several variance reduction techniques such as quasi-Monte-Carlo sampling (Leövey and Römisch, 2015; Heitsch et al., 2016), importance sampling (Parpas et al., 2015) and Latin hyper-cube sampling (Linderoth et al., 2006) can be utilised to generate such samples. Moreover, some authors suggested the generation of samples that should satisfy a specified criterion, such as probability distances (Pflug, 2001; Dupačová et al., 2003) or moment discrepancies (Høyland et al., 2003; Gülpınar et al., 2004).

Once samples are generated, various optimisation algorithms can be used to solve the resulting SAA model. One approach is to utilise the simplex algorithm, which is conveniently implemented by modern optimisation solvers. Alternatively, some studies exploited the problem structure and proposed decomposition-based optimisation algorithms, for instance, see Dantzig and Wolfe (1960) and Van Slyke and Wets (1969). Subsequently, numerous authors introduced advanced procedures such as the multi-cut approach (Birge and Louveaux, 1988), the trust region method (Linderoth and Wright, 2003), the regularised decomposition (Ruszczyński and Świetanowski, 1997) and the level bundle method (Wolf et al., 2014; van Ackooij et al., 2017) to improve the efficiency of utilising

the decomposition principle. For a comprehensive review on decomposition approaches, the readers are referred to Rahmaniani et al. (2017).

The identification of high-quality solutions has been widely studied because of their importance for performance-sensitive SAA users. The approximate solutions can asymptotically converge to the optimal one as the number of samples gets sufficiently large, for details, see Shapiro and Homem-de Mello (1998) and Homem-de Mello and Bayraksan (2014). Shapiro et al. (2002) showed that, given an arbitrary number of samples, each solution has a certain probability of being the optimal one, and the probability value is related to problem-specific factors and the number of samples. If the best solution is selected from several potential ones on the basis of their performances, then the probability of determining the optimal solution is significantly increased. Therefore, large quantities of samples and potential solutions are both important in searching for high-quality SAA solutions. However, these requirements may be difficult to satisfy simultaneously within a given period because they compete with each other on time allocations. As suggested by Lee et al. (2006), a practical remedy is to determine a proper balance between these objectives by using computing time allocation algorithms. The authors also showed that the algorithmic efficiency significantly influences performance of the final solution. The application of highly efficient optimisation approaches, such as the aforementioned decomposition-based method, is beneficial because the SAA user can implement a large sample size to strengthen the approximation of random data and obtain improved solutions within the same timeframe. Moreover, an effective solution selection method is also important because it can promptly determine the best option among a large group of potential solutions. However, only a limited number of studies is concerned with solution selection for the SPR problems.

Defourny et al. (2013) applied a brute-force approach that runs extensive simulations for each potential solution to identify a good policy in multi-stage linear stochastic programming. Instead of individually evaluating solution performance, Kleywegt et al. (2002) used a ranking and selection approach called indifference zone to determine a good solution in two-stage stochastic discrete optimisation. The indifference zone approach, which assigns simulation replications for each potential solution on the basis of performance statistics and guarantees the overall procedure at least a certain probability of selecting the best solution, was proposed by Nelson et al. (2001). However, this approach is not an anytime algorithm, which means that a specified simulation rule (i.e.

simulation replications for each potential solution) is followed to finish all requested simulations so that a probability guarantee is achieved. Also, this method is highly conservative and usually takes many more samples than necessary (Branke et al., 2007).

In the present study, we propose a solution selection method for the large-scale two-stage linear SPR problems that can deal with numerous potential solutions and return fairly efficient solutions within a finite computational budget. The contributions of this study are threefold.

- First, a Wasserstein-based screening ($\mathcal{WS}$) approach is proposed to identify potentially promising solutions. We demonstrate that the worst-case performances of SAA solutions in the respective Wasserstein distance regions can be ranked by using the Wasserstein distance between the sampling measure used in the SAA model and the original probability measure. Solutions with small distance values have good worst-case performances and thus be classified as the most promising evaluated in the simulation.

- Second, an optimal computing budget allocation technique ($\mathcal{OCBA}$) (He et al., 2007) is used to determine how many simulation replications to use for each potential solution. The technique is an anytime algorithm, which asymptotically minimises the penalty of selecting an incorrect solution, so that the probability of achieving a good potential solution is greatly increased. We then introduce a new two-stage selection process called $\mathcal{WS}$-$\mathcal{OCBA}$, which integrates $\mathcal{OCBA}$ with $\mathcal{WS}$ to improve the simulation efficiency.

- Third, we conduct several numerical experiments to analyse performance of $\mathcal{WS}$ and the $\mathcal{WS}$-$\mathcal{OCBA}$ approaches. Results show that $\mathcal{WS}$ achieves a satisfactory trade-off between the number of potential solutions in the promising group and the performance loss. The findings also indicate that $\mathcal{WS}$-$\mathcal{OCBA}$ outperforms the existing approaches under relatively limited run times.

The remainder of this paper is structured as follows. In Section 2, we provide a brief overview of two-stage stochastic programming and introduce the solution performance estimation procedure. Section 3 introduces the definition of Wasserstein distance and its estimation. Section 4 discusses the underlying principle of $\mathcal{WS}$-$\mathcal{OCBA}$ and describes the overall algorithm. In Section 5, we study the efficacy of our proposed strategies. Section 6 concludes the paper by summarising our findings.

2. Sample Average Approximation for Two-stage Linear SPR

2.1. Two-stage Linear SPR Formulation

Although SPR can be applied to problems with arbitrary number of decision stages, we restrict our discussion to the two-stage linear SPR problems because the specific stability result of Wasserstein distance used in the solution screening is limited to this class of SPR. We let $\xi \in \mathbb{R}^{\mu}$ be a random vector with a finite first moment. Specifically, random vector ξ is defined on the probability space $(\Xi, \mathcal{B}(\Xi), \mathbb{P})$, where Ξ is the sample space, $\mathcal{B}(\Xi)$ is the Borel sigma algebra with respect to Ξ, and $\mathbb{P} : \mathcal{B}(\Xi) \rightarrow [0, 1]$ is the probability measure. Without loss of generality, a two-stage linear SPR problem with fixed recourse can be formulated as

$$\min_{x \in \mathscr{X}} \; f(x) := \min_{x \in \mathscr{X}} \; c'x + \int_{\Xi} g(x, \xi) \, \mathbb{P}(d\xi) \tag{1}$$

where $c \in \mathbb{R}^{\kappa}$ is a vector of constant parameters and $\mathscr{X} \in \mathbb{R}^{\kappa}$ represents a non-empty convex feasible set for the first-stage decision vector $x \in \mathbb{R}^{\kappa}$. In addition, let $g(x, \xi)$ denote the optimal value of the second-stage decision problem, formulated as follows:

$$g(x, \xi) = \min_{y \in \mathbb{R}^{\iota}} \{q'y \mid Wy = H(\xi) - T(\xi)x, \; y \geq 0\} \tag{2}$$

where $q \in \mathbb{R}^{\iota}$ and $W \in \mathbb{R}^{\mu \times \iota}$ are a fixed vector and a fixed matrix, respectively. Moreover, $T(\xi) \in \mathbb{R}^{\mu \times \kappa}$ and $H(\xi) \in \mathbb{R}^{\mu}$ affinely depend on random vector ξ in this study.

We also make the following assumptions throughout this study:

A(1) **Relatively complete recourse**: For each tuple $(\hat{x}, \xi)$, the corresponding second-stage decision problem (2) is feasible.

A(2) **Dual feasibility**: There exists at least one π such that the set $\{\pi \mid \pi' W \leq q\}$ is not empty.

Assumption A(1) ensures the feasibility of the primal second-stage decision problem. Assumption A(2) implies dual feasibility in the second-stage decision problem, i.e., a sub-gradient value exists for any potential solution. Assumptions A(1) and A(2) represent necessary conditions for the stability result of the Wasserstein metric (for detailed information, see Heitsch and Römisch (2007)) which will be used in our approach.

For the two-stage linear SPR problems, properties of the probability space majorly influence the computational burden. The problems, in the case of random data with continuous sample

space, are rarely solvable because the resulting model formulation consists of an infinite number of second-stage decision variables and constraints. Moreover, the SPR problems might still suffer from computational intractability even when the probability distribution of random data is discrete. For instance, consider random data with 10 components, each of which follows a uniform distribution and can take 200 possible values. If we select one possible value for each component according to its distribution and then combine them as one sample, then the number of distinctive scenarios reaches 200^{10}. Since the computational complexity increases exponentially with the number of samples taken into account, the optimal solution is difficult to obtain. The SAA approach can be applied for identifying near-optimal solutions to the SPR problem.

2.2. The Near-optimal Solution and its Performance Estimator

Suppose that a group of samples $\hat{\Xi} = \{\hat{\xi}_m : m = 1, \ldots, M\}$ with respective probability values $\{\mathbb{Q}(\xi_m) : m = 1, \ldots, M\}$ is generated from the random data, thereby we can obtain the following approximate model,

$$\begin{aligned} \min_{x,\ y(\hat{\xi}_m)} \ & c'x + \sum_{m=1}^{M} \mathbb{Q}(\xi_m) q' y(\hat{\xi}_m) \\ s.t. \quad & x \in \mathscr{X}, \\ & W y(\hat{\xi}_m) = H(\hat{\xi}_m) - T(\hat{\xi}_m)x, \quad m = 1, \ldots, M, \\ & y(\hat{\xi}_m) \geq 0, \quad m = 1, \ldots, M. \end{aligned} \tag{3}$$

The resulting solution is typically not the optimal solution for the original SPR model, so it is important to evaluate its performance in the original model.

Let $\hat{x}$ and $f(\hat{x})$ denote a potential solution and its performance in the original model, respectively. Mak et al. (1999) suggested using Monte-Carlo estimation to infer the value $f(\hat{x})$. Assume that we have K *i.i.d* batches of samples with size N and equal probabilities; that is, $\tilde{\Xi}_k = \{\tilde{\xi}_n^k : n = 1, \ldots, N_E\}$, for $k = 1, \ldots, K$. We can estimate a true solution performance for each batch k of samples by computing the optimal value $\hat{f}_{N_E}^k(\hat{x})$:

$$\begin{aligned} \hat{f}_{N_E}^k(\hat{x}) := \frac{1}{N_E} \sum_{n=1}^{N_E} \min_{y(\tilde{\xi}_n^k)} \ & \left[c'\hat{x} + q' y(\tilde{\xi}_n^k) \right] \\ s.t. \ & W y(\tilde{\xi}_n^k) = H(\tilde{\xi}_n^k) - T(\tilde{\xi}_n^k)\hat{x} \\ & y(\tilde{\xi}_n^k) \geq 0. \end{aligned} \tag{4}$$

By averaging over all optimal values $\hat{f}^k_{N_E}(\hat{x})$, for $k = 1, \ldots, K$, we obtain an estimator of $f(\hat{x})$ as

$$\mathcal{J}(\hat{x}) = \frac{1}{K}\sum_{k=1}^{K} \hat{f}^k_{N_E}(\hat{x}). \tag{5}$$

Let $\sigma^2(\hat{f}_{N_E}(\hat{x}))$ denote the population variance of the optimal values of SAA models with the first-stage decision $\hat{x}$ and sample size N_E. Mak et al. (1999) demonstrated that $\mathcal{J}(\hat{x})$ is an unbiased estimator and follows the Central Limit Theorem:

$$\sqrt{K}\Big[\mathcal{J}(\hat{x}) - f(\hat{x})\Big] = \sqrt{K}\Big[\frac{1}{K}\sum_{k=1}^{K} \hat{f}^k_{N_E}(\hat{x}) - f(\hat{x})\Big] \to \mathcal{N}\Big(0, \sigma^2(\hat{f}_{N_E}(\hat{x}))\Big), \text{ when } K \to \infty. \tag{6}$$

Note that $\mathcal{N}\Big(0, \sigma^2(\hat{f}_{N_E}(\hat{x}))\Big)$ is a Gaussian distribution with variance $\sigma^2(\hat{f}_{N_E}(\hat{x}))$ and zero mean. Then, the population variance $\sigma^2(\hat{f}_{N_E}(\hat{x}))$ can be estimated by using the following estimator as

$$\mathcal{V}(\hat{x}) = \frac{\sum_{k=1}^{K}\Big[\hat{f}^k_{N_E}(\hat{x}) - \mathcal{J}(\hat{x})\Big]^2}{K-1}. \tag{7}$$

In the SAA framework, the performance of an approximate solution depends on the number of samples used to represent the random data; therefore, a sufficient number of samples should be included in the model formulation. Moreover, rather than focusing on only one solution, considering multiple potential solutions can be also beneficial for finding the optimal solution. In the following subsection, we will introduce the solution selection problem.

2.3. Solution Selection Under a Fixed Computing Budget

Suppose that a group of potential solutions is given and a simulation is required to select the best one as the final solution under the fixed computational budget. One of the challenges we might encounter is that the "best" solution based on Monte-Carlo estimation may not be really the best solution if insufficient information is available for analysing the solution performance. In this study, we model the solution selection process as a computing time allocation problem. Let $\{\hat{x}^\lambda : \lambda = 1, \ldots, \Lambda\}$ denote a set of potential solutions. The performance of each solution $\hat{x}^\lambda$ is evaluated by K^λ batch samples with size N_E. The CPU time to compute each batch sample in (4) is denoted by $t(\hat{x}^\lambda, N_E)$. Let the batch number K^λ for $\lambda = 1, \ldots, \Lambda$ represent unknown decision variables. Moreover, we define $\hat{x}^s$ as the best solution according to simulation results. Given the

total simulation budget T^{total}, the computing time allocation problem can be formulated as follows:

$$\begin{aligned} \min_{K^1,K^2,\ldots,K^\Lambda} \quad & f(\hat{x}^s) \\ s.t. \quad & \sum_{\lambda=1}^{\Lambda} t(\hat{x}^\lambda, N_E)K^\lambda \leq T^{total}, \\ & \hat{x}^s = \arg\min\left\{\frac{1}{K^\lambda}\sum_{k=1}^{K^\lambda}\hat{f}^k_{N_E}(\hat{x}) : \lambda = 1,\ldots,\Lambda\right\}, \\ & K^\lambda \in \mathbb{N}, \quad \lambda = 1,\ldots,\Lambda. \end{aligned} \tag{8}$$

It is challenging to solve the solution selection problem in the sense that its objective function represents true performance of potential solutions. In Section 3, we introduce a heuristic method to address the above mentioned resource allocation problem. The proposed heuristic method utilises the Wasserstein distance to adjust the simulation focus on several potentially promising solutions.

3. The Wasserstein Distance Metric and its Estimation

In this section, we first present the definition of the Wasserstein distance between two probability measures, and then introduce a method for estimating the Wasserstein distance.

3.1. Formulation of Wasserstein Distance Metric

The Wasserstein metric is a kind of statistical metric for quantifying the dissimilarity between two probability measures, and has been widely applied in stochastic programming. One application is scenario reduction wherein the Wasserstein distance is used as the quality indicator of samples in the approximate model. Given a fixed number of samples used in the approximate model, some authors proposed heuristics to select the "best" samples with minimum distance. For example, Dupačová et al. (2003) presented two myopic scenario reduction heuristics for two-stage SPR, namely, forward selection and backward reduction. Furthermore, Heitsch and Römisch (2009) extended these heuristics for multi-stage SPR. In another application, the Wasserstein distance was used to define an ambiguity set for stochastic programs with distributional uncertainty (Mohajerin Esfahani and Kuhn, 2017). In addition, the Wasserstein distance was applied to reduce the optimality gap estimator balance, and this application benefits testing the optimality of a given solution (Stockbridge and Bayraksan, 2013). In this study, we use the Wasserstein distance metric

to roughly rank the performance of potential solutions. The Wasserstein distance can be calculated as follows.

Definition 1. ***Wasserstein Distance Metric*** *(Dudley, 2002): Consider an approximate model that is defined by a probability measure $\mathbb{Q}$ with sample $\hat{\Xi} = \{\hat{\xi}_m : m = 1, \ldots, M\}$. Let $\Xi = \{\xi_n : n = 1, \ldots, N\}$ represent the samples associated with probability measure $\mathbb{P}$ in the original SPR model. Moreover, let $\theta = \{\theta_{n,m} : n = 1, \ldots, N; m = 1, \ldots, M\}$ denote the joint measure with marginal measures $\mathbb{P}$ and $\mathbb{Q}$. The Wasserstein distance $W(\mathbb{P}, \mathbb{Q})$ between probability measures $\mathbb{P}$ and $\mathbb{Q}$ can be formulated as the following optimisation problem.*

$$
\begin{aligned}
W(\mathbb{P}, \mathbb{Q}) := \min_{\theta_{n,m}} \; & \sum_{m=1}^{M} \sum_{n=1}^{N} Z(\xi_n, \hat{\xi}_m) \theta_{n,m} \\
s.t. \; & \sum_{m=1}^{M} \theta_{n,m} = \mathbb{P}(\xi_n), \; n = 1, \ldots, N \\
& \sum_{n=1}^{N} \theta_{n,m} = \mathbb{Q}(\hat{\xi}_m), \; m = 1, \ldots, M \\
& \theta_{n,m} \geq 0, n = 1, \ldots, N, \; m = 1, \ldots, M
\end{aligned}
\tag{9}
$$

where $Z(\xi_n, \hat{\xi}_m) = \|\xi_n - \hat{\xi}_m\|_2$ is the Euclidean distance between samples ξ_n and $\hat{\xi}_m$.

The optimisation problem (9) has $N \times M$ decision variables and $N + M + N \times M$ linear constraints. As discussed in Section 2.2, the number of samples N can be very large or even infinite, thus an exact calculation of $W(\mathbb{P}, \mathbb{Q})$ can be difficult. In order to tackle this, we provide an estimation method based on the random measures in the following section.

3.2. The Wasserstein Distance Estimation

We can replace the probability measure $\mathbb{P}$ with a group of sampling measures, and take the average Wasserstein distance between random measures and probability measure $\mathbb{Q}$ as the estimator of $W(\mathbb{P}, \mathbb{Q})$. The following Lemma states that the bias of such an estimator is bounded.

Lemma 1. *Let P^k for $k = 1, \ldots, K_W$ denote the sampling measure induced by N_W realisations generated from the probability measure $\mathbb{P}$. If we use the estimator*

$$
\hat{W}(\mathbb{P}, \mathbb{Q}) = \frac{1}{K_W} \sum_{k=1}^{K_W} W(P^k, \mathbb{Q})
$$

to infer the Wasserstein distance value $W(\mathbb{P},\mathbb{Q})$, then its bias satisfies the following inequality,

$$\left|W(\mathbb{P},\mathbb{Q}) - \hat{W}(\mathbb{P},\mathbb{Q})\right| \leq \frac{1}{K_W}\sum_{k=1}^{K_W} W(P^k,\mathbb{P}). \tag{10}$$

Proof. Since the Wasserstein distance is a metric, it satisfies the reverse triangle inequality:

$$|W(\mathbb{P},\mathbb{Q}) - W(P^k,\mathbb{Q})| \leq W(P^k,\mathbb{P}). \tag{11}$$

Next, we can compute the sum of inequalities (11) over all P^k for $k = 1,\ldots,K_W$ as follows;

$$-\sum_{k=1}^{K_W} W(P^k,\mathbb{P}) \leq K_W \times W(\mathbb{P},\mathbb{Q}) - \sum_{k=1}^{K_W} W(P^k,\mathbb{Q}) \leq \sum_{k=1}^{K_W} W(P^k,\mathbb{P}). \tag{12}$$

By dividing both sides of the above inequality by K_W, we obtain inequality (10).

Lemma 1 states that the absolute value of estimation bias is bounded by the average Wasserstein distance. We can increase the sample size of the sampling measure to minimise the fluctuation of bias. According to the Glivenko-Cantelli theorem, if the sample size becomes sufficiently large, then P^k tends to $\mathbb{P}$, and thus the absolute value of bias fluctuation converges toward zero. The main benefit of using Lemma 1 is to reduce the computational burden due to the large sample number N. We can now solve a group of relatively small optimisation problems to infer the actual Wasserstein distance $W(\mathbb{P},\mathbb{Q})$. Each "small" optimisation problem only has $N_W \times M$ decision variables and $N_W + M + N_W \times M$ linear constraints. The overall Wasserstein distance estimation procedure is described in Algorithm 1.

Algorithm 1: Wasserstein Distance Estimation Procedure

input : number of sampling measures K_W;
number of samples in each sampling measure N_W.

output: estimated Wasserstein distance value $\hat{W}(\mathbb{P},\mathbb{Q})$.

generate K_W groups of i.i.d N_W realisations from probability measure $\mathbb{P}$;

for $k = 1,\ldots,K_W$ **do**

  compute the Wasserstein distance $W(P^k,\mathbb{Q})$ using (9);

end

$\hat{W}(\mathbb{P},\mathbb{Q}) \leftarrow \frac{1}{K_W}\sum_{k=1}^{K_W} W(P^k,\mathbb{Q})$.

4. The Proposed Approach

In this section we first explain the underlying principles of $\mathcal{WS}$ and $\mathcal{OCBA}$ approaches. Then, we introduce the proposed selection procedure.

4.1. Wasserstein-based Screening Approach

When the number of potential solutions is large and the computing budget is insufficient to perform an extensive simulation for every solution, it is natural to consider performing the simulation for a certain portion of potential solutions. An important question is *how to select a set of promising solutions for the extensive simulation.* In this paper, we propose to select those solutions through the Wasserstein distance due to its low computational cost that allows it to be run for every potential solution. Specifically, the potential solutions are sorted according to their Wasserstein distances, and only the high-rank solutions are selected for extensive simulation. A similar paradigm called "ordinal transformation" is used in a simulation optimisation study (Xu et al., 2016), wherein the authors considered the simulation output on a user-defined low-fidelity model as the low-cost measure for each potential solution. The potential solutions were clustered according to low-fidelity simulation results and the extensive simulation was applied to select the best solution cluster.

We first establish the relationship between the solution performance and the corresponding Wasserstein distance in the following theorem.

Theorem 1. *Suppose that $\mathbb{P}$ and $\mathbb{Q}$ are original probability measure and sampling measure, respectively. Let $x^* \in \mathbb{R}^\kappa$ denote the unique optimal solution of the original SPR model, $\hat{x} \in \mathbb{R}^\kappa$ be the solution of an approximate model. Under assumptions A(1) and A(2), the performance difference between solutions x^* and $\hat{x}$ satisfies the following inequality:*

$$f(\hat{x}) \leq f(x^*) + \langle |\mathcal{L}_{\hat{x}}|, \tau W(\mathbb{P}, \mathbb{Q}) I \rangle, \tag{13}$$

where $\mathcal{L}_{\hat{x}} = \mathbb{E}[\pi_{\hat{x}}(\xi)T(\xi)] + c$ with $\pi_{\hat{x}}(\xi) = \operatorname{argmax}\{\pi'[H(\xi) - T(\xi)\hat{x}] \ : \ \pi' W \leq q\}$, τ is a positive coefficient related to the Lipschitz constants for $f(\cdot)$, and $I \in \mathbb{R}^\kappa$ is the all one vector.

Proof. Consider the two-stage SPR given in (1). Let $\mathcal{L}_{\hat{x}}$ denote one of sub-gradients for solution $\hat{x}$. Under assumptions A(1) and A(2), the following sub-gradient inequality holds,

$$f(x^*) \geq f(\hat{x}) + \langle \mathcal{L}_{\hat{x}}, x^* - \hat{x} \rangle. \tag{14}$$

Let $\pi_{\hat{x}}(\xi)$ represent a vector of dual decisions associated with constraints of the second-stage problem (2). It is computed as follows,

$$\pi_{\hat{x}}(\xi) = \arg\max_{\pi} \{\pi'[H(\xi) - T(\xi)\hat{x}] \ : \ \pi' W \leq q\}.$$

Moreover, the sub-gradient of the two-stage SLP problem at $\hat{x}$ can be calculated as

$$\mathcal{L}_{\hat{x}} = \mathbb{E}[\pi'_{\hat{x}}(\xi)T(\xi)] + c.$$

By substituting the sub-gradient and re-arranging both sides of inequality (14), we obtain

$$f(\hat{x}) - f(x^*) \leq -\langle \mathbb{E}[\pi'_{\hat{x}}(\xi)T(\xi)] + c, x^* - \hat{x}\rangle. \tag{15}$$

Since $\hat{x}$ is a potential solution, $f(\hat{x})$ cannot be smaller than $f(x^*)$ for the minimisation problem. Therefore, we can obtain

$$|f(\hat{x}) - f(x^*)| \leq \langle |\mathbb{E}[\pi'_{\hat{x}}(\xi)T(\xi)] + c|, |x^* - \hat{x}|\rangle. \tag{16}$$

Using Proposition 3.1 introduced by Heitsch and Römisch (2007), the distance between solutions x^* and $\hat{x}$ is upper bounded by the Wasserstein distance $W(\mathbb{P}, \mathbb{Q})$ as follows,

$$\|x^* - \hat{x}\|_2 = \sqrt{\sum_{\kappa'=1}^{\kappa} (x^*_{\kappa'} - \hat{x}_{\kappa'})^2} \leq \tau W(\mathbb{P}, \mathbb{Q}), \tag{17}$$

where τ is a positive coefficient related to the Lipschitz constants for $f(\cdot)$. For each element of solutions x^* and $\hat{x}$, we can write

$$\tau W(\mathbb{P}, \mathbb{Q}) \geq \sqrt{\sum_{\kappa'=1}^{\kappa} (x^*_{\kappa'} - \hat{x}_{\kappa'})^2} \geq \sqrt{(x^*_{\kappa'} - \hat{x}_{\kappa'})^2} = |x^*_{\kappa'} - \hat{x}_{\kappa'}|, \ \kappa' = 1, \ldots, \kappa. \tag{18}$$

Therefore, $|x^* - \hat{x}|$ can be approximated by using the Wasserstein distance, $W(\mathbb{P}, \mathbb{Q})$, as follows,

$$|x^* - \hat{x}| \leq \tau W(\mathbb{P}, \mathbb{Q}) I. \tag{19}$$

By combining inequality (19) with (15), we obtain (13). □

Theorem 1 implies that performance of potential solution $\hat{x}$ is bounded by the sub-gradient $\mathcal{L}_{\hat{x}}$ and the Wasserstein distance $W(\mathbb{P}, \mathbb{Q})$. The potential solution $\hat{x}$ becomes the optimal solution x^*

for the actual problem if either probability measures $\mathbb{Q}$ and $\mathbb{P}$ are identical or the sub-gradient at $\hat{x}$ becomes zero.

As mentioned before, the Wasserstein distance $W(\mathbb{P}, \mathbb{Q})$ can be efficiently estimated using Monte-Carlo estimation; however, the calculation of sub-gradients $\mathcal{L}_{\hat{x}}$ for two-stage liner SPRs is computationally expensive. Next, we introduce the worst-case solution performance in the Wasserstein-bounded region. We will use this performance measure in the solution screening procedure.

Definition 2. ***Worst-case Solution Performance in the Wasserstein-bounded Region.*** *Let $\Gamma(\hat{x})$ represent a set possessing sub-gradients of all feasible solutions within the bounded region $\{x : |x - x^*| \leq \tau W(\mathbb{P}, \mathbb{Q})I\}$. The worst-case performance $\mathcal{G}^w(\hat{x})$ of a given solution $\hat{x}$ within its corresponding Wasserstein-bounded region can be determined as follows:*

$$\mathcal{G}^w(\hat{x}) = \max_{\mathcal{L}' \in \Gamma(\hat{x})} \left\{ f(x^*) + \left\langle \left| \mathcal{L}' \right|, \tau W(\mathbb{P}, \mathbb{Q})I \right\rangle \right\}. \tag{20}$$

Notice that $\mathcal{G}^w(\hat{x})$ is defined as a maximisation problem with respect to the sub-gradient value $\mathcal{L}'$. The next theorem states the applicability of the Wasserstein distance in sequencing the worst-case performance of potential solutions.

Theorem 2. *Consider a set of potential solutions $\{x^\lambda : \lambda = 1, \ldots, \Lambda\}$ with respective Wasserstein distances $\{W(\mathbb{P}, \mathbb{Q}^\lambda) : \lambda = 1, \ldots, \Lambda\}$. Let $[\lambda]$ denote the λ-th potential solution in the increasing sequence of Wasserstein distances as*

$$W(\mathbb{P}, \mathbb{Q}^{[1]}) \leq W(\mathbb{P}, \mathbb{Q}^{[2]}) \leq \cdots \leq W(\mathbb{P}, \mathbb{Q}^{[\Lambda]}). \tag{21}$$

Then, the worst-case solution performances of these solutions satisfy,

$$\mathcal{G}^w(\hat{x}^{[1]}) \leq \mathcal{G}^w(\hat{x}^{[2]}) \leq \cdots \leq \mathcal{G}^w(\hat{x}^{[\Lambda]}). \tag{22}$$

Proof. Assume that the Wasserstein distances satisfy the sequence as in (21). Since the inequality (19) can be written for all potential solutions $\hat{x}^\lambda$, we can construct the same relationship as in sequence of

$$\begin{aligned} &\left\{ x : |x - x^*| \leq \tau W(\mathbb{P}, \mathbb{Q}^{[1]})I \right\} \subseteq \left\{ x : |x - x^*| \leq \tau W(\mathbb{P}, \mathbb{Q}^{[2]})I \right\} \\ &\subseteq \cdots \subseteq \left\{ x : |x - x^*| \leq \tau W(\mathbb{P}, \mathbb{Q}^{[\Lambda]})I \right\}. \end{aligned} \tag{23}$$

Hence, the feasibility set of all sub-gradients at approximate potential solutions holds the sequence of

$$\Gamma(\hat{x}^{[1]}) \subseteq \Gamma(\hat{x}^{[2]}) \subseteq \cdots \subseteq \Gamma(\hat{x}^{[\Lambda]}). \tag{24}$$

From (21) and (24), one can say that the solution with a smaller Wasserstein distance leads to a smaller feasibility set and a small coefficient vector in the optimisation problem (20). Hence, the same sequence order also holds for the worst-case performances at potential solutions as stated in (22). □

Theorem 2 indicates that the sequence of Wasserstein distances for SAA solutions encapsulates the trend of the worst-case solution performances. We should note that the rank of actual solution performances in general does not follow the sequence of Wasserstein distances. Therefore, when the solution screening is applied according to the sequence of Wasserstein distances, a performance loss that is caused by eliminating the best solution might arise. We describe the performance loss as follows.

Definition 3. ***Performance Loss****. Let $\mathcal{P}$ denote a set of promising solutions obtained from a specific screening procedure. If the computing budget is restricted on those promising solutions, then the performance loss due to screening out the best solution $\hat{x}^b$ can be computed as*

$$PL(\mathcal{P}, \hat{x}^b) = \min\{f(p) : p \in \mathcal{P}\} - f(\hat{x}^b). \tag{25}$$

If $PL(\mathcal{P}, \hat{x}^b) = 0$, then the promising group contains the best solution. Otherwise, $PL(\mathcal{P}, \hat{x}^b)$ is always greater than zero. The value $PL(\mathcal{P}, \hat{x}^b)$ reflects the quality of the promising group; thus, having a screening procedure that has a performance guarantee is desirable. Next, we prove that the proposed screening approach provides an upper bound for the performance loss.

Theorem 3. *Assume that for a set of potential solutions $\{\hat{x}^\lambda : \lambda = 1, \ldots, \Lambda\}$, the corresponding set of Wasserstein distances $\{W(\mathbb{P}, \mathbb{Q}^\lambda) : \lambda = 1, \ldots, \Lambda\}$ possesses an increasing sequence of distance values. In other words, the following inequalities hold:*

$$W(\mathbb{P}, \mathbb{Q}^{[1]}) \leq W(\mathbb{P}, \mathbb{Q}^{[2]}) \leq \cdots \leq W(\mathbb{P}, \mathbb{Q}^{[\Lambda]}). \tag{26}$$

Then, the Wasserstein-based screening provides the following upper bound for the performance loss:

$$PL(\mathcal{P}, \hat{x}^b) \leq \mathcal{G}^w(\hat{x}^{[1]}) - f(\hat{x}^b). \tag{27}$$

PROOF. From (13) and (20), we can write the following inequality for each promising solution

$$f(p) \leq \mathcal{G}^w(p), \quad \forall p \in \mathcal{P}. \tag{28}$$

Hence, from the definition of performance loss, it follows:

$$\min\{f(p) : p \in \mathcal{P}\} \leq \min\{\mathcal{G}^w(p) : p \in \mathcal{P}\}. \tag{29}$$

This yields

$$PL(\mathcal{P}, \hat{x}^b) = \min\{f(p) : p \in \mathcal{P}\} - f(\hat{x}^b) \leq \min\{\mathcal{G}^w(p) : p \in \mathcal{P}\} - f(\hat{x}^b). \tag{30}$$

Clearly, selecting a promising subgroup of solutions out of the top $|\mathcal{P}|$ of the lowest Wasserstein distances (using Theorem 2) provides

$$\min\{\mathcal{G}^w(p) : p \in \mathcal{P}\} - f(\hat{x}^b) \leq \mathcal{G}^w(\hat{x}^{[1]}) - f(\hat{x}^b). \tag{31}$$

So by combining with (30), we find that (27) holds. □

Theorem 3 implies that the Wasserstein-based screening provides a fixed upper bound for the performance loss even without performing any simulation. The tightness of the bound depends on the stability result of the Wasserstein distance.

4.2. Optimal Computing Budget Allocation

The $\mathcal{OCBA}$ technique is a class of ranking and selection methods that maximises confidence in the correct selection evidence under the simulation budget restriction. In this study, we use an $\mathcal{OCBA}$ variant in which the correct selection evidence is defined as expected opportunity cost (He et al., 2007). Let us denote $\hat{x}^b$ as the true best solution and $\hat{x}^s$ as the solution with best performance based on the simulation results. Then, the expected opportunity cost $\mathbb{E}(\text{OC})$ quantifies the penalty due to wrong selection and is defined as follows:

$$\mathbb{E}(\text{OC}) = \mathbb{E}\Big[f(\hat{x}^s) - f(\hat{x}^b)\Big] = \sum_{\lambda=1,\ \lambda\neq s}^{\Lambda} Prob(\hat{x}^\lambda = \hat{x}^b)\Big[f(\hat{x}^s) - f(\hat{x}^\lambda)\Big], \tag{32}$$

where $Prob(\hat{x}^\lambda = \hat{x}^b)$ denotes the probability of solution $\hat{x}^\lambda$ being the true best solution $\hat{x}^b$.

As $Prob(\hat{x}^\lambda = \hat{x}^b)$ is unknown in practice, He et al. (2007) proposed to use an upper bound approximation of $\mathbb{E}(\text{OC})$ that can be estimated during the simulation procedure. Let $\phi(\cdot)$ and $\Phi(\cdot)$

be the probability density function (PDF) and cumulative distribution function of standard normal distribution, respectively. Moreover, K^s is the number of evaluated samples for the solution $\hat{x}^s$ and K^λ is the number of evaluated samples for solution $\hat{x}^\lambda$. The upper bound approximation of expected opportunity cost (abbreviated as AEOC) is presented as follows:

$$\mathbb{E}(\text{OC}) \leq \text{AEOC} = \sum_{\lambda=1,\lambda\neq s}^{\Lambda} \left\{ \mathcal{V}_{s,\lambda}\phi(z_{s,\lambda}) + \delta_{s,\lambda}\Phi(-z_{s,\lambda}) \right\}, \tag{33}$$

where $\mathcal{V}_{s,\lambda} = \frac{\mathcal{V}(\hat{x}^s)}{K^s} + \frac{\mathcal{V}(\hat{x}^\lambda)}{K^\lambda}$, $\delta_{s,\lambda} = \mathcal{J}(\hat{x}^s) - \mathcal{J}(\hat{x}^\lambda)$ and $z_{s,\lambda} = \frac{-\delta_{s,\lambda}}{\sqrt{\mathcal{V}_{s,\lambda}}}$. Note that $\mathcal{V}(\hat{x}^\lambda)$ and $\mathcal{V}(\hat{x}^s)$ are the estimation variances of solution $\hat{x}^\lambda$ and $\hat{x}^s$, which can be computed by (7). The derivation of the upper bound approximation of opportunity cost is provided in Appendix A.

Given a finite computing budget, we aim to allocate the simulation budget to sequentially minimise the upper bound of expected opportunity cost. If we evaluate one additional sample for the solution $\hat{x}^\lambda$, then the upper bound of expected opportunity cost will change to

$$\widehat{\text{AEOC}}_\lambda = \sum_{\lambda'=1,\lambda'\neq s}^{\Lambda} \int_0^\infty x \ \eta_{s,\lambda',\lambda} \ dx, \ \lambda = 1, \ldots, \Lambda, \tag{34}$$

where $\eta_{s,\lambda',\lambda}$ is the PDF of the normally distributed random variable and defined as

$$\begin{cases} \mathcal{N}\left(f(\hat{x}^s) - f(\hat{x}^\lambda), \frac{\mathcal{V}(\hat{x}^s)}{K^s+1} + \frac{\mathcal{V}(\hat{x}^{\lambda'})}{K^{\lambda'}} \right), & \text{if } \hat{x}^\lambda = \hat{x}^s \\ \mathcal{N}\left(f(\hat{x}^s) - f(\hat{x}^\lambda), \frac{\mathcal{V}(\hat{x}^s)}{K^s} + \frac{\mathcal{V}(\hat{x}^{\lambda'})}{K^{\lambda'}+1} \right), & \text{if } \hat{x}^\lambda = \hat{x}^{\lambda'} \\ \mathcal{N}\left(f(\hat{x}^s) - f(\hat{x}^\lambda), \frac{\mathcal{V}(\hat{x}^s)}{K^s} + \frac{\mathcal{V}(\hat{x}^{\lambda'})}{K^{\lambda'}} \right), & \text{if } \hat{x}^\lambda \neq \hat{x}^s \text{ and } \hat{x}^\lambda \neq \hat{x}^{\lambda'}. \end{cases}$$

The above integration can be computed by using (33). Then, the possible reduction of AEOC can be computed as,

$$\mathcal{Y}_\lambda = \text{AEOC} - \widehat{\text{AEOC}}_\lambda \geq 0. \tag{35}$$

Next, the sample would be assigned to the solution that leads to the maximum reduction of AEOC, i.e.,

$$\lambda^* = \arg\max\{\mathcal{Y}_\lambda : \lambda = 1, \ldots, \Lambda\}. \tag{36}$$

The overall procedure is described in Algorithm 2. The $\mathcal{OCBA}$ technique can reduce the overall simulation budget necessary to identify the best solution. However, the $\mathcal{OCBA}$ procedure might fail to identify the best solution given abundant potential solutions. For instance, if the limited

Algorithm 2: The $\mathcal{OCBA}$ Procedure

input : potential solutions: $\{\hat{x}^\lambda :; \lambda = 1, \ldots, \Lambda\}$;
size of sample N_E in the performance estimation;
number of samples K_0 evaluated in the initial estimation.

output: best solution based on the simulation results.

while *simulation budget is available* **do**
  Step 1: Initial Estimation;
  for $\lambda = 1, \ldots, \Lambda$ **do**
    evaluate the performance of solution $\hat{x}^\lambda$ using K_0 samples;
    compute the performance statistics of solution $\hat{x}^\lambda$ using (6) and (7);
    select the current best solution $\hat{x}^s$;
  end
  Step 2: Sequential Decision Process;
  for $\lambda = 1, \ldots, \Lambda$ **do**
    compute the expected opportunity cost reduction $\mathcal{Y}_\lambda$ as in (33) and (34);
  end
  find $\lambda^* = \text{argmax}\{\mathcal{Y}_\lambda, \lambda = 1, \ldots, \Lambda\}$;
  simulate one additional sample for solution $\hat{x}^{\lambda *}$ and update its statistics;
end
select the current best solution $\hat{x}^s$.

computing budget runs out during the initial estimation of Algorithm 2, then some solutions will not be evaluated. If the best solution is one of those ignored, then it is not possible to correctly identify the best solution. Moreover, the solution selection method is challenging when the computing budget is insufficient in the sequential decision process because the initial estimation has taken a large proportion of the computing budget. Then insufficient estimation might mislead our choice of the final solution.

4.3. The Proposed Two-stage Selection Approach

The above difficulties motivate the use of a solution screening method prior to $\mathcal{OCBA}$. Algorithm 3 describes integration of the solution screening approach into $\mathcal{OCBA}$. The Wasserstein-based solution screening method selects potential solutions that are then more closely examined using $\mathcal{OCBA}$. The solution screening procedure takes a proportion of the computing budget. Thus, it is necessary to properly allot the computing budget among different procedures; namely, solution

Algorithm 3: The $\mathcal{WS}$-$\mathcal{OCBA}$ Procedure

input : potential solutions: $\{\hat{x}^\lambda : \lambda = 1, \ldots, \Lambda\}$;
size of sample N_E in the performance estimation;
number of samples K_W with size N_W in the Wasserstein distance estimation;
number of promising solutions: $|\mathcal{P}|$;
number of samples K_0 evaluated in the initial estimation.

output: best solution based on the simulation results.

for $\lambda = 1, \ldots, \Lambda$ **do**
| run Algorithm 1 for solution x^λ to estimate the corresponding Wasserstein distance;

end

rank and select the top $|\mathcal{P}|$ potential solutions according to their Wasserstein distances;

while *computing budget is available* **do**
| run Algorithm 2.

end

screening, initial estimation, and sequential allocation. We now highlight some rules for a practical application of Algorithm 3.

- A proper computing budget allocation is important for $\mathcal{WS}$ and $\mathcal{OCBA}$. The solution screening process provides the sequence of potential solutions based on their worst-case performances in the corresponding Wasserstein-bounded region whereas the actual performance is exploited in $\mathcal{OCBA}$. Therefore, we should allocate the computing budget to $\mathcal{OCBA}$ such that there are sufficient simulation replications to guarantee the accuracy of the selection procedure.

- A large sample size is preferred for the Wasserstein distance estimation because it helps to reduce potential estimation bias; however, the computational cost of the Wasserstein distance estimation should be controlled to a certain level, which aims to secure sufficient computing budget in $\mathcal{OCBA}$. The rule of thumb is that, for a single solution, the computational cost of the Wasserstein distance estimation should be smaller than that of the initial estimation in Algorithm 2.

- The number of potential solutions to be run for simulation (i.e., promising solutions) is determined by the computing budget to be allocated for $\mathcal{OCBA}$. We should guarantee to

have sufficient computing budget for Step 2 of Algorithm 2. Otherwise, the insufficient simulation may lead to selection errors.

5. Numerical Experiments

5.1. Overview of Test Instances

In this section, we describe four benchmark problem instances to be used for the numerical experiments. Table 1 presents complexity of these instances in terms of number of decision variables (D), constraints (C), dimension of uncertainties (Dim), number of scenarios (NS) as well as the optimal values (mean $\pm$ standard error). The corresponding problem descriptions are briefly

Table 1: Complexity of benchmark problems

Problem Instances	D/C		Dim	NS	Optimal Values
	$1st$ stage	$2nd$ stage			
LandS	4/2	12/7	3	1×10^6	225.620±0.020
Retail	7/0	70/22	7	1×10^{11}	154.410±0.770
20term	64/3	764/124	40	1.1×10^{12}	254298.572±38.743
SSN	89/1	706/175	86	1.1×10^{70}	9.840±0.100

summarised as below:

- **LandS** is an electricity planning problem (Louveaux and Smeers, 1988). The first-stage decisions are concerned with an allocation of four power terminals, and the second-stage decisions are related to allocating the power supply to various residential areas.
- **Retail** is taken from Herer et al. (2006), which is a supply chain optimisation problem involving multiple retailers and one supplier. The objective is to design the optimal replenishment policy for each retailer.
- **20term**, adopted from Linderoth et al. (2006), is a large-scale vehicle management problem. The first-stage decisions find the vehicle locations at the beginning of the plan, and the second-stage decisions optimise the fleet transportation plan on the basis of the initial vehicle locations.

- **SSN** (Sen et al., 1994) aims to design an optimal telecommunication network that can minimise the number of lost demands. In view of randomness of the communication demand in the network, the bandwidth between destination and origin nodes must be sufficiently large to satisfy the customer demand. Otherwise, that demand will be lost.

All test instances are minimisation problems, which are written in SMPS format and publicly available online [1]. This study implements an SMPS parser that is based on the Julia programming language and the COIN-OR linear program solver. All numerical experiments are computed on a machine with i7-6700K CPU and 32GB memory.

5.2. Performance of Wasserstein-based Screening Approach

We adopt the following benchmark strategies to compare with $\mathcal{WS}$:

- Random screening ($\mathcal{RS}$): a fixed number of solutions is randomly selected from the potential solutions.
- Moment discrepancy ($\mathcal{MD}$): a fixed number of solutions is selected according to the discrepancy of the first four statistical moments between samples of the original model and those of the approximate model.

The experimental settings are designed as follows. We implement SAA with 200 samples to generate 200 potential solutions for each test instance. Table 2 presents the statistical description of those potential solutions. Figure 1 graphically illustrates performances of the potential solutions in terms of the mean, range and confidence intervals. We apply three strategies to select various numbers (from 1 to 200) of solutions and compute the corresponding performance loss (as discussed in Section 4.1). The performance of each solution is evaluated with 30 groups of 500,000 samples. For $\mathcal{WS}$, the solution's Wasserstein distance is estimated using four samples with a size of 1,000.

Figure 2 presents the average performance loss over 30 runs in terms of the number of promising solutions for all test instances. The results in Figure 2 confirm the superiority of $\mathcal{WS}$ for all test instances. The performance losses of $\mathcal{RS}$ and $\mathcal{MD}$ are much worse than that of $\mathcal{WS}$ when the number of promising solutions is less than 30. The $\mathcal{MD}$ approach outperforms $\mathcal{RS}$ in all benchmark problems except LandS. The average performance losses obtained by three strategies coincide as the

[1] `http://pages.cs.wisc.edu/~swright/stochastic/sampling/` and `http://plato.asu.edu/sd/instances/`

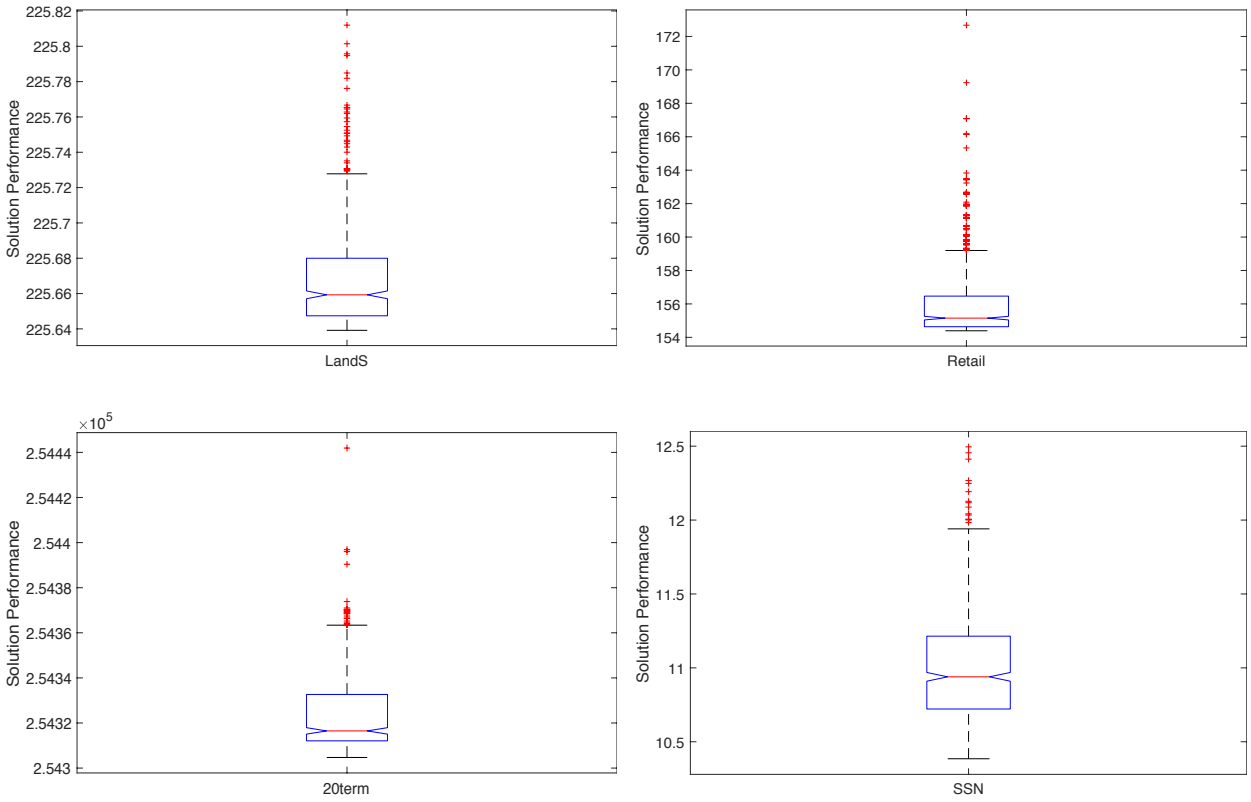

Figure 1: Box plots providing the mean (horizontal line in box), range (vertical lines extending from box), and 95% confidence intervals (vertical extent of box) for the performances of potential solutions

number of promising solutions approaches to 200. Moreover, the convergence of $\mathcal{WS}$ in LandS and Retail is more rapid than that in SSN and 20term, indicating that the stability of the quantitative result varies with the problem structure. The results also imply that the required number of promising solutions for guaranteeing a relatively small performance loss changes with the problem structure. For example, more solutions need to be included in the simulation procedure for 20term and SSN than LandS and Retail.

Table 2: Statistical description of potential solutions

Problem Instances	Mean	Standard Deviation	Maximum	Minimum
LandS	225.659	0.029	225.823	225.639
Retail	156.034	2.225	172.673	154.393
20term	254318.492	20.011	254441.922	254305.193
SSN	10.990	0.375	12.496	10.385

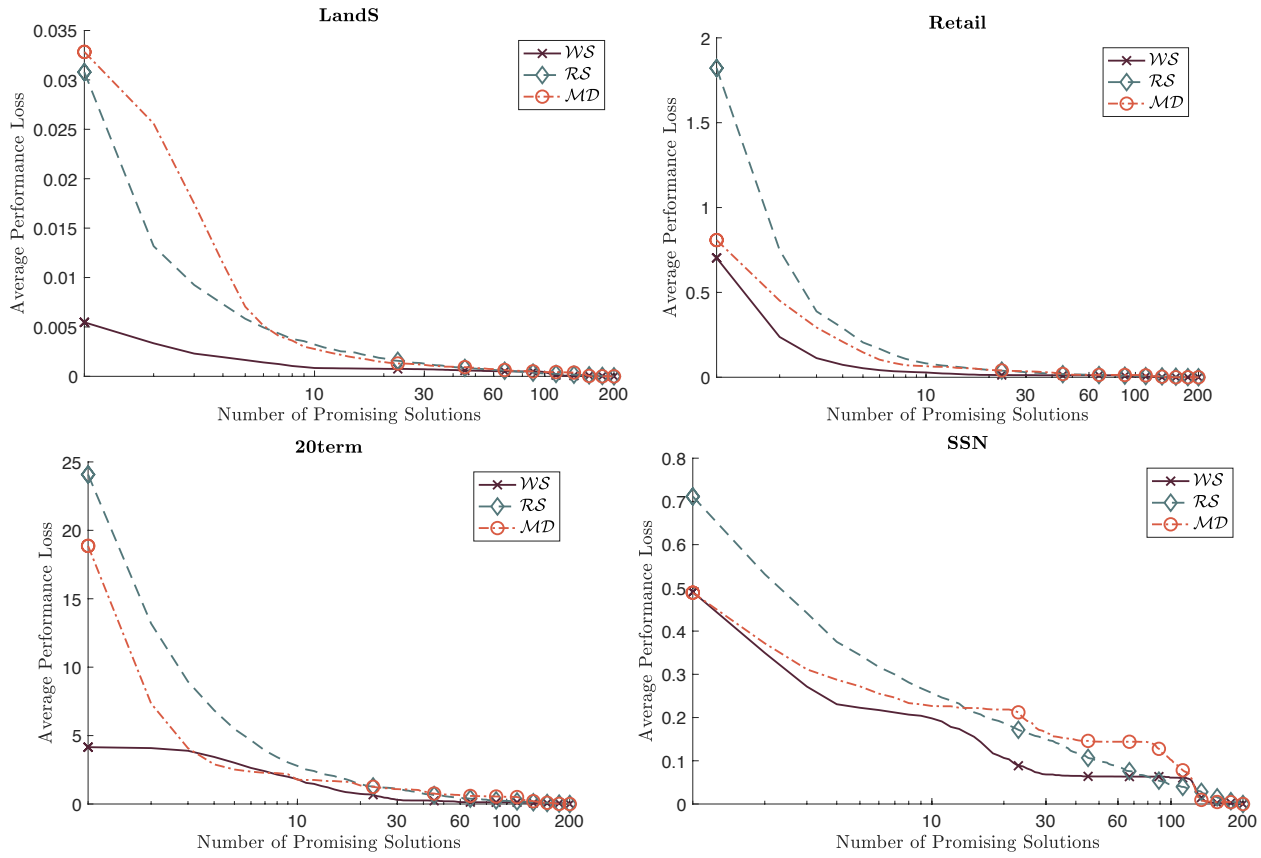

Figure 2: Comparison of average performance losses obtained by $\mathcal{RS}$ and $\mathcal{WS}$ using various numbers of promising solutions.

5.3. Average Performance Comparison with Benchmark Algorithms

We also conduct numerical experiments to study the performance of the solution selection algorithm in terms of the CPU time. The performance of $\mathcal{WS}$-$\mathcal{OCBA}$ is compared with that of the following approaches.

- $\mathcal{OCBA}$ is the standard $\mathcal{OCBA}$ algorithm as described in Algorithm 2.
- $\mathcal{EAS}$ uses the equal allocation and selection algorithm, which sequentially allocates the same number of simulations to each potential solution and selects the current best solution based on simulation results.
- $\mathcal{WS}$-$\mathcal{EAS}$ employs $\mathcal{EAS}$ with the proposed Wasserstein-based solution screening procedure.

For each test instance, we utilise SAA with 200 samples to generate 200 potential solutions and then apply all algorithms to identify the best potential solution. This procedure is repeated 100

times to examine the average performance of each algorithm. The detailed experimental settings are listed in Table 3. Figure 3 presents the convergence comparisons over 100 runs in terms of CPU time and average solution performance. The computational costs of solution screening for the $\mathcal{WS}$-based algorithms are included in the comparison results.

Table 3: Experimental settings

Problem Instances	Algorithms	Solution Screening				N_E	C/E(s)	Initial Estimation	
		$\|\mathcal{P}\|$	K_W	N_W	TC(s)			K	TC(s)
LandS	$\mathcal{EAS}$	–	–	–	–	50,000	1.5	–	–
	$\mathcal{OCBA}$							5	1,460
	$\mathcal{WS}$-$\mathcal{EAS}$	10	4	1,000	91			–	–
	$\mathcal{WS}$-$\mathcal{OCBA}$							5	1,460
Retail	$\mathcal{EAS}$	–	–	–	–	50,000	2.4	–	–
	$\mathcal{OCBA}$							5	2,400
	$\mathcal{WS}$-$\mathcal{EAS}$	10	4	1,000	105			–	–
	$\mathcal{WS}$-$\mathcal{OCBA}$							5	2,400
20term	$\mathcal{EAS}$	–	–	–	–	20,000	20.4	–	–
	$\mathcal{OCBA}$							5	20,400
	$\mathcal{WS}$-$\mathcal{EAS}$	10	4	1,000	128			–	–
	$\mathcal{OCBA}$							5	20,400
SSN	$\mathcal{EAS}$	–	–	–	–	20,000	24.8	–	–
	$\mathcal{OCBA}$							5	24,800
	$\mathcal{WS}$-$\mathcal{OCBA}$	10	4	1,000	154			–	–
	$\mathcal{WS}$-$\mathcal{OCBA}$							5	24,800

$|P|$: number of promising solutions; K_W: number of samples for the Wasserstein distance estimation;
N_W : sample size for Wasserstein estimation; TC: total CPU time;
N_E: sample size for performance evaluation; C/E: CPU time of each evaluation;
K: number of samples used in the initial estimation.

As shown in Figure 3, the $\mathcal{EAS}$ algorithm provides the worst results for all test instances among all algorithms because the equal allocation strategy spends unnecessary evaluations on potential solutions with a large mean. We also observe that $\mathcal{EAS}$ and $\mathcal{OCBA}$ exhibit similar convergence patterns on LandS (before 1,500s) and Retail (before 2,500s) whereas the overall patterns of $\mathcal{EAS}$ and $\mathcal{OCBA}$ for 20term and SSN are almost identical. The reason behind this phenomenon could

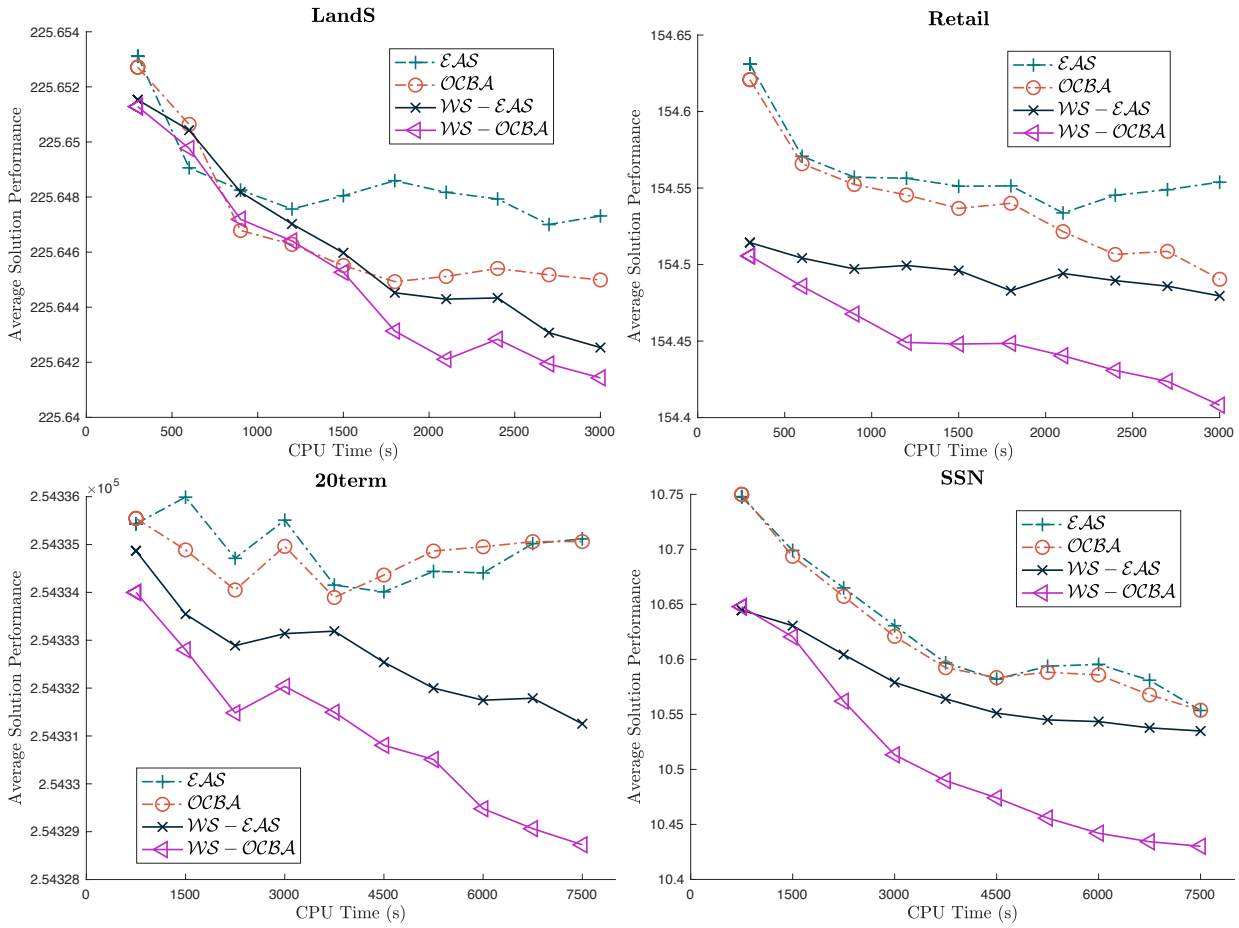

Figure 3: Convergence comparison with respect to CPU time

be because $\mathcal{OCBA}$ has a sufficient computing budget to finish the initial estimation on LandS and Retail, but this is not the case for test instances 20term and SSN (see Table 3). When $\mathcal{OCBA}$ stays in the stage of initial estimation, its evaluation behaviour (as described in Algorithm 2) is similar to $\mathcal{EAS}$.

However, $\mathcal{OCBA}$ performs worse than any of the $\mathcal{WS}$-based algorithms on all test instances, indicating that $\mathcal{WS}$ enhances performance of the solution selection algorithm for the case of numerous potential solutions. For $\mathcal{WS}$-$\mathcal{OCBA}$, the computational cost for initial estimation is greatly reduced. We observe that $\mathcal{WS}$-$\mathcal{OCBA}$ achieves the fastest convergence in all test instances, as this algorithm has more time to explore the performance of each promising solution compared with $\mathcal{OCBA}$.

We display the performance of solutions obtained from various algorithms averaged over 100

repetitions in Table 4. The best results and those statistically not different from the best are highlighted in bold. The results confirm the importance of adopting solution screening in the selection procedure for numerous potential solutions. The $\mathcal{WS}$-based algorithms significantly outperform the others in all benchmark problems although all algorithms display similar performance in LandS and 20terms when the CPU time is 1,000s. Moreover, we find that $\mathcal{WS}$-$\mathcal{OCBA}$ performs better than $\mathcal{WS}$-$\mathcal{EAS}$ due to its advanced computing budget allocation scheme.

Table 4: Average solution performance of the selected subset (mean ± std. err).

Problem	CPU	Algorithms			
Instances	Time (s)	$\mathcal{EAS}$	$\mathcal{OCBA}$	$\mathcal{WS}$-$\mathcal{EAS}$	$\mathcal{WS}$-$\mathcal{OCBA}$
	1,000	**225.648±0.002**	**225.647±0.001**	**225.648±0.002**	**225.647±0.001**
LandS	2,000	225.648±0.001	225.645±0.001	**225.644±0.001**	**225.643±0.001**
	3,000	225.647±0.001	225.645±0.001	**225.643±0.001**	**225.641±0.001**
	1,000	154.556±0.013	154.545±0.017	154.499±0.009	**154.449±0.003**
Retail	2,000	154.545±0.012	154.506±0.008	154.490±0.009	**154.431±0.001**
	3,000	154.554±0.013	154.490±0.006	154.479±0.003	**154.408±0.001**
	2,500	254334.892±1.041	254334.267±1.050	254333.018±0.894	**254331.562±0.835**
20term	5,000	254334.172±0.952	254334.797±0.948	254332.134±0.707	**254330.358±0.665**
	7,500	254335.117±0.936	254335.062±0.931	254331.255±0.792	**254328.726±0.706**
	2,500	10.654±0.015	10.645±0.014	10.595±0.014	**10.543±0.012**
SSN	5,000	10.589±0.015	10.586±0.016	10.546±0.011	**10.430±0.009**
	7,500	10.553±0.013	10.553±0.014	10.534±0.011	**10.461±0.008**

6. Conclusions

The solution selection problem for large-scale two-stage problems is challenging for decision makers with a relatively limited computational budget when numerous potential solutions are present. Thus, we may consider removing several potential solutions from the simulation. This study shows that the worst-case solution performance in the corresponding Wasserstein-based regions satisfies the sequence of the corresponding Wasserstein distances. On the basis of this property, we propose a new solution screening approach and integrate this approach with an optimal computing budget allocation algorithm. Empirical results for various benchmark problems demonstrate the benefit of the Wasserstein-based screening and the advantage of applying the op-

timal computing budget allocation algorithm in selecting the near-best solutions. Future studies can refine the idea of the Wasserstein-based screening in a sophisticated manner to determine the number of promising solutions. The Wasserstein-based solution screening may also be extended to the multi-stage SPR problems.

Appendix A. An Upper Bound Approximation of Expected Opportunity Cost

He et al. (2007) provided an upper bound approximation for the probability $Prob(\hat{x}^{\lambda} = \hat{x}^{b})$ as follows,

$$Prob(\hat{x}^{\lambda} = \hat{x}^{b}) \leq Prob\bigg(f(\hat{x}^{\lambda}) < f(\hat{x}^{s})\bigg). \tag{A.1}$$

Hence, we can obtain

$$\begin{aligned} \mathbb{E}(\mathrm{OC}) &\leq \sum_{\lambda=1,\lambda\neq s}^{\Lambda} Prob\bigg(f(\hat{x}^{\lambda}) < f(\hat{x}^{s})\bigg)\bigg[f(\hat{x}^{s}) - f(\hat{x}^{\lambda})\bigg] \\ &= \sum_{\lambda=1,\lambda\neq s}^{\Lambda} \int_{0}^{+\infty} t\ \eta_{s,\lambda}(t)\ dt = \mathrm{AEOC}, \end{aligned} \tag{A.2}$$

where $\eta_{s,\lambda}$ denotes the PDF of random value $\mathcal{N}\bigg(f(\hat{x}^{s}) - f(\hat{x}^{\lambda}), \mathcal{V}_{s,\lambda}\bigg)$. The integration in (A.2) can be calculated by using the following equation:

$$\int_{0}^{+\infty} \eta_{s,\lambda}(t) = \mathcal{V}_{s,\lambda}\phi(z_{s,\lambda}) + \delta_{s,\lambda}\Phi(-z_{s,\lambda}). \tag{A.3}$$

References

Barbarosolu, G., Arda, Y., 2004. A two-stage stochastic programming framework for transportation planning in disaster response. Journal of the Operational Research Society 55 (1), 43–53.

Beale, E. M., 1955. On minimizing a convex function subject to linear inequalities. Journal of the Royal Statistical Society. Series B (Methodological), 173–184.

Beraldi, P., Conforti, D., Violi, A., 2008. A two-stage stochastic programming model for electric energy producers. Computers & Operations Research 35 (10), 3360–3370.

Birge, J. R., Louveaux, F. V., 1988. A multicut algorithm for two-stage stochastic linear programs. European Journal of Operational Research 34 (3), 384–392.

Branke, J., Chick, S. E., Schmidt, C., 2007. Selecting a selection procedure. Management Science 53 (12), 1916–1932.

Cheung, R. K., Chen, C.-Y., 1998. A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem. Transportation Science 32 (2), 142–162.

Dantzig, G. B., 1955. Linear programming under uncertainty. Management Science 1 (3-4), 197–206.

Dantzig, G. B., Wolfe, P., 1960. Decomposition principle for linear programs. Operations Research 8 (1), 101–111.

Defourny, B., Ernst, D., Wehenkel, L., 2013. Scenario trees and policy selection for multistage stochastic programming using machine learning. INFORMS Journal on Computing 25 (3), 488–501.

Dillon, M., Oliveira, F., Abbasi, B., 2017. A two-stage stochastic programming model for inventory management in the blood supply chain. International Journal of Production Economics 187, 27–41.

Dudley, R. M., 2002. Real analysis and probability. Vol. 74. Cambridge University Press.

Dupačová, J., 1999. Portfolio optimization via stochastic programming: Methods of output analysis. Mathematical Methods of Operations Research 50 (2), 245–270.

Dupačová, J., Gröwe-Kuska, N., Römisch, W., 2003. Scenario reduction in stochastic programming. Mathematical Programming 95 (3), 493–511.

Feng, Y., Ryan, S. M., sep 2014. Solution sensitivity-based scenario reduction for stochastic unit commitment. Computational Management Science 13 (1), 29–62.

Gülpınar, N., Rustem, B., Settergren, R., 2004. Simulation and optimization approaches to scenario tree generation. Journal of Economic Dynamics and Control 28 (7), 1291–1315.

Gürkan, G., Özge, A. Y., Robinson, S. M., 1994. Sample-path optimization in simulation. In: Proceedings of the 26th conference on Winter simulation. Society for Computer Simulation International, pp. 247–254.

He, D., Chick, S. E., Chen, C.-H., 2007. Opportunity cost and ocba selection procedures in ordinal optimization for a fixed number of alternative systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37 (5), 951–961.

Heitsch, H., Leövey, H., Römisch, W., 2016. Are quasi-monte carlo algorithms efficient for two-stage stochastic programs? Computational Optimization and Applications 65 (3), 567–603.

Heitsch, H., Römisch, W., 2007. A note on scenario reduction for two-stage stochastic programs. Operations Research Letters 35 (6), 731–738.

Heitsch, H., Römisch, W., 2009. Scenario tree reduction for multistage stochastic programs. Computational Management Science 6 (2), 117.

Herer, Y. T., Tzur, M., Yücesan, E., 2006. The multilocation transshipment problem. IIE Transactions 38 (3), 185–200.

Homem-de Mello, T., Bayraksan, G., 2014. Monte carlo sampling-based methods for stochastic optimization. Surveys in Operations Research and Management Science 19 (1), 56–85.

Høyland, K., Kaut, M., Wallace, S. W., 2003. A heuristic for moment-matching scenario generation. Computational Optimization and Applications 24 (2), 169–185.

Joensson, H., Jörnsten, K., Silver, E. A., 1993. Application of the scenario aggregation approach to a two-stage, stochastic, common component, inventory problem with a budget constraint. European Journal of Operational Research 68 (2), 196–211.

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

Lee, L. H., Chew, E. P., Manikam, P., 2006. A general framework on the simulation-based optimization under fixed computing budget. European Journal of Operational Research 174 (3), 1828–1841.

Leövey, H., Römisch, W., 2015. Quasi-monte carlo methods for linear two-stage stochastic programming problems. Mathematical Programming 151 (1), 315–345.

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

Linderoth, J., Wright, S., 2003. Decomposition algorithms for stochastic programming on a computational grid. Computational Optimization and Applications 24 (2-3), 207–250.

Liu, C., Fan, Y., Ordóñez, F., 2009. A two-stage stochastic programming model for transportation network protection. Computers & Operations Research 36 (5), 1582–1590.

Louveaux, F. V., Smeers, Y., 1988. Optimal investments for electricity generation: A stochastic model and a test problem. Numerical Techniques for Stochastic Optimization, 33–64.

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

Miller, N., Ruszczyński, A., 2011. Risk-averse two-stage stochastic linear programming: Modeling and decomposition. Operations Research 59 (1), 125–132.

Mohajerin Esfahani, P., Kuhn, D., 2017. Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations. Mathematical Programming.

Nelson, B. L., Swann, J., Goldsman, D., Song, W., 2001. Simple procedures for selecting the best simulated system when the number of alternatives is large. Operations Research 49 (6), 950–963.

Parpas, P., Ustun, B., Webster, M., Tran, Q. K., 2015. Importance sampling in stochastic programming: A markov chain monte carlo approach. INFORMS Journal on Computing 27 (2), 358–377.

Pflug, G. C., 2001. Scenario tree generation for multiperiod financial optimization by optimal discretization. Mathematical Programming 89 (2), 251–271.

Rahmaniani, R., Crainic, T. G., Gendreau, M., Rei, W., 2017. The benders decomposition algorithm: A literature review. European Journal of Operational Research 259 (3), 801–817.

Ruszczyński, A., Świetanowski, A., 1997. Accelerating the regularized decomposition method for two stage stochastic linear problems. European Journal of Operational Research 101 (2), 328–342.

Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A., 2005. A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research 167 (1), 96–115.

Sen, S., Doverspike, R. D., Cosares, S., 1994. Network planning with random demand. Telecommunication Systems 3 (1), 11–30.

Shapiro, A., Homem-de Mello, T., 1998. A simulation-based approach to two-stage stochastic programming with recourse. Mathematical Programming 81 (3), 301–325.

Shapiro, A., Homem-de Mello, T., Kim, J., 2002. Conditioning of convex piecewise linear stochastic programs. Mathematical Programming 94 (1), 1–19.

Stockbridge, R., Bayraksan, G., 2013. A probability metrics approach for reducing the bias of optimality gap estimators in two-stage stochastic linear programming. Mathematical Programming 142 (1-2), 107–131.

van Ackooij, W., de Oliveira, W., Song, Y., 2017. Adaptive partition-based level decomposition methods for solving two-stage stochastic programs with fixed recourse. Informs Journal on Computing 30 (1), 57–70.

Van Slyke, R. M., Wets, R., 1969. L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics 17 (4), 638–663.

Wets, R., 1966. Programming under uncertainty: The complete problem. Probability Theory and Related Fields 4 (4), 316–339.

Wolf, C., Fábián, C. I., Koberstein, A., Suhl, L., 2014. Applying oracles of on-demand accuracy in two-stage stochastic programming–a computational study. European Journal of Operational Research 239 (2), 437–448.

Xu, J., Zhang, S., Huang, E., Chen, C.-H., Lee, L. H., Celik, N., 2016. Mo2tos: Multi-fidelity optimization with ordinal transformation and optimal sampling. Asia-Pacific Journal of Operational Research 33 (03), 1650017.

Zhou, Z., Zhang, J., Liu, P., Li, Z., Georgiadis, M. C., Pistikopoulos, E. N., 2013. A two-stage stochastic programming model for the optimal design of distributed energy systems. Applied Energy 103, 135–144.