

Rapid screening algorithms for stochastically constrained problems

Shing Chih Tsai¹ · Tse Yang¹

Published online: 8 March 2017
© Springer Science+Business Media New York 2017

Abstract A simulation optimization framework containing three fundamental stages (feasibility check, screening, and selection) is proposed for solving the zero-one optimization via simulation problem in the presence of a single stochastic constraint. We present three rapid screening algorithms that combine these three stages in different manners, such that various sampling mechanisms are applied, therefore yielding different statistical guarantees. An empirical evaluation for the efficiency comparison between the proposed algorithms and other existing works is provided.

Keywords Simulation · Output analysis · Optimization via simulation · Ranking and selection · Stochastic constraint

1 Introduction

Most real-world problems usually have complicated and difficult features, and thus some simplified modeling assumptions have to be imposed in order to create analytical tractability. However, another promising and straightforward approach is to build and study a simulation system that is more reflective of reality that may possibly result in better decisions. In many simulation studies the analyst is concerned about identifying the decision variables that can maximize or minimize the expected performance measure of interest, and this is called the optimization via simulation (OvS) problem [see [Amaran et al. \(2016\)](#) for a survey]. An appropriate solution approach for the small-scale OvS problem (often less than 500 candidate solutions) is ranking and selection (R&S) procedures [see, for instance, [Chen \(2011\)](#)]. Traditional R&S procedures are characterized by the consideration of a single performance

Electronic supplementary material The online version of this article (doi:[10.1007/s10479-017-2459-6](https://doi.org/10.1007/s10479-017-2459-6)) contains supplementary material, which is available to authorized users.

✉ Shing Chih Tsai
sctsai@mail.ncku.edu.tw

¹ National Cheng Kung University, No. 1, University Road, Tainan City, Taiwan

measure and the presence of a fixed set of candidate solutions available before the simulation experiment. For instance, [Nelson et al. \(2001\)](#) developed a two-stage procedure that employs the subset-selection procedure to eliminate inferior solutions in the first stage, and then determines the number of total samples required to find the best among the remaining solutions in the second stage. [Kim and Nelson \(2001\)](#) proposed a fully sequential procedure that obtains a single additional observation from the surviving solutions and then performs the elimination process at each stage. [Boesel et al. \(2003\)](#) and [Pichtlamken et al. \(2006\)](#) extended the aforementioned two-stage and fully sequential research works, respectively, to be able to handle the case of unequal initial sample sizes, which is particularly useful during the search process of an OvS algorithm. [Tsai \(2013\)](#) designed rapid screening algorithms, which consist of multiple sampling stages along with an intelligent solution-generation mechanism, to solve zero-one OvS problems efficiently. All of the above works considered only a single performance measure.

Recently, more research effort has been dedicated to OvS problems involving multiple performance measures. For instance, [Andradóttir and Kim \(2010\)](#) developed fully sequential procedures that can select the best solution in the presence of a single stochastic constraint, which was later extended by [Batur and Kim \(2010\)](#) to consider multiple stochastic constraints. [Park and Kim \(2015\)](#) proposed an approach called “penalty function with memory”, which is imposed on the objective function, and then the constrained OvS problem is replaced by a series of unconstrained problems. The constrained or multicriteria OvS problems arise in a wide variety of decision-making scenarios, with practical examples including water resource management (e.g., [Udías et al. 2014](#)), healthcare operations management (e.g., [Williams et al. 2010](#)) and call center staffing problems (e.g., [Atlason et al. 2004](#)).

The goal of this paper is to develop rapid screening algorithms for solving the zero-one OvS problem considering a single stochastic constraint. The proposed simulation optimization framework contains three fundamental stages: a Feasibility Check, Screening, and Selection. We present three algorithms that combine these three stages in different ways, such that various sampling mechanisms are applied, therefore yielding different statistical guarantees. A pre-specified number (R) of screening iterations is performed, in which inferior solutions are eliminated through a comparison with other chosen solutions, and then a random number of new solutions are generated from the neighborhood of surviving solutions. In each objective screening iteration, the number of replications allocated to each solution is specified in advance, and it grows at a constant rate. If more than one solution is left at the end of the R screening iterations, we then apply the selection procedure of [Boesel et al. \(2003\)](#) to choose the best solution. The first algorithm implements the feasibility check procedure of [Andradóttir and Kim \(2010\)](#) and then uses the objective screening procedure of [Tsai \(2013\)](#) in each screening iteration. In other words, it only applies screening procedure (with respect to the objective performance) to those solutions that have been identified as feasible (with a given confidence level). Notice that the number of simulation samples used in the feasibility check is random and that we have to take new samples in the following objective screening procedure (to maintain statistical validity). The second proposed algorithm is more heuristic-oriented; it implements constraint screening and then objective screening (in each iteration) based on simulation observations collected from the same replications. It performs feasibility check once only to the solutions surviving through the R screening iterations. We also introduce an alternative version that considers more candidate solutions in the feasibility check stage, and the statistical validity can thus be maintained to some extent. The third proposed algorithm allows us to perform objective screening between feasible solutions and other solutions whose feasibility has not been determined yet. Prior to this operation, it employs the ordinary feasibility check procedure based on a pre-specified number of observations allocated to each

iteration. We adopt a stricter rule for the elimination of inferior solutions, and after these R iterations, we also have to conduct a feasibility check procedure on the surviving solutions whose feasibility has not been determined yet (with the accumulated observations).

The paper is organized as follows. In Sect. 2 we define the stochastically constrained zero-one OvS problem and describe the statistical guarantee that our algorithms can provide. Section 3 introduces the relevant notations and presents three rapid screening algorithms for the constrained problem. An empirical evaluation for the comparison between the proposed algorithms and other existing works is provided in Sect. 4. Finally, Sect. 5 provides some concluding remarks and possible future research directions. All proofs and some details of our algorithms are relegated to the Appendix in the Online Supplement.

2 Framework

The purpose of this study is to determine a set of zero-one decision variables that yields the best expected primary performance measure, while also satisfying a single stochastic constraint with respect to a secondary performance measure. We let X_{ij} and Y_{ij} denote the j th simulation observation of the i th solution for the primary and secondary performance measures, respectively. The i th solution is a vector of d zero-one decision variables, and is denoted by $\mathbf{S}_i = (S_{i1}, S_{i2}, \dots, S_{id})$. The expected performances of solution \mathbf{S}_i with respect to the objective and the constraint are defined as $\mathbf{X}(\mathbf{S}_i) = E[X_{ij}]$ and $\mathbf{Y}(\mathbf{S}_i) = E[Y_{ij}]$. Therefore, the constrained zero-one simulation optimization problem can be formulated as follows:

$$\max_{\mathbf{S}_i \in \Omega} \mathbf{X}(\mathbf{S}_i)$$

where the feasible region Ω is required to satisfy the following stochastic constraint:

$$\mathbf{Y}(\mathbf{S}_i) \leq Q,$$

where Q is a constant specified by the analyst. The expected performance of $\mathbf{X}(\mathbf{S}_i)$ and $\mathbf{Y}(\mathbf{S}_i)$ is unknown and analytically intractable, but it can be measured or estimated by running simulation experiments. Similar to the recent R&S literature considering multiple performance measures (e.g., [Andradóttir and Kim 2010](#)), we make the following bivariate normal (denoted as **BN**) assumption on (X_{ij}, Y_{ij}) for $i = 1, 2, \dots, j = 1, 2, \dots$:

$$\begin{bmatrix} X_{ij} \\ Y_{ij} \end{bmatrix} \stackrel{iid}{\sim} \mathbf{BN} \left(\begin{bmatrix} \mathbf{X}(\mathbf{S}_i) \\ \mathbf{Y}(\mathbf{S}_i) \end{bmatrix}, \Sigma_i \right)$$

where $\stackrel{iid}{\sim}$ denotes independent and identically distributed, and Σ_i is the variance-covariance matrix of a vector of observations (X_{ij}, Y_{ij}) . The normality assumption is quite reasonable when X_{ij} and Y_{ij} are themselves the within-replication averages of some output variables from multiple independent replications, or when they are the batch means of a large number of basic outputs in the context of steady-state simulation. It should be noted that X_{ij} and Y_{ij} can be correlated because they are simulated observations taken from the same solution design. For instance, the time-based service level (to satisfy customer demands) and the backorder cost of an inventory system are often negatively correlated. However, the observation vectors (X_{ij}, Y_{ij}) and $(X_{\ell j}, Y_{\ell j})$ are assumed to be independent for $i \neq \ell$ (i.e., common random numbers (CRN) are not employed). Further, in the case of a complex system with random noise, it is almost impossible to check exactly whether the candidate solution is feasible with respect to the stochastic constraint. We thus follow the same setting as in [Andradóttir and](#)

Kim (2010) to allow some relaxation of the constraint. Specifically, we have to specify a so-called tolerance level ϵ , and then define the upper bound $Q^+ = Q + \epsilon$ and the lower bound $Q^- = Q - \epsilon$ (for the constraint threshold). Subsequently, we can divide the solution space (for S_i) into three regions as follows: (1) if $Y(S_i) \leq Q^-$, then we call S_i a desirable solution (i.e., $S_i \in \mathcal{R}_D$) (2) if $Q^- < Y(S_i) \leq Q^+$, then we call S_i an acceptable solution (i.e., $S_i \in \mathcal{R}_A$) (3) if $Y(S_i) > Q^+$, then we call S_i an unacceptable solution (i.e., $S_i \in \mathcal{R}_U$). Andradóttir and Kim (2010) and Batur and Kim (2010) developed feasibility check procedures that can guarantee the choice of all desirable solutions and probably some acceptable solutions. Suppose that the set I contains the indexes of all the candidate solutions visited and evaluated by the proposed algorithm (therefore, $|I| \leq 2^d$). Without any loss of generality, we also assume that the set I includes k desirable solutions and that the index $[k]$ represents the best desirable solution. That is, we can denote the ordered, but unknown expected performance (with respect to the primary measure) for the k visited desirable solutions as:

$$X_{[1]} \leq X_{[2]} \leq \dots \leq X_{[k]}.$$

The desirable solution associated with $X_{[i]}$ is unknown, but is denoted as $S_{[i]}$. We let δ denote an indifference-zone parameter representing the smallest difference worth detecting (with respect to the expected primary performance measure). Most of the proposed rapid screening algorithms can provide the following statistical guarantee in terms of probability of correct selection (PCS):

$$PCS[I] = \Pr\{\text{select } S_{[k]} \mid X_{[k]} \geq X_i + \delta, \forall i \in (\mathcal{R}_D \cup \mathcal{R}_A), \text{ with } i \neq [k]\} \geq 1 - \alpha.$$

This means that we can select solution $S_{[k]}$ with a pre-specified confidence level $1 - \alpha$ whenever its expected measure (with respect to the objective performance) is superior to other visited desirable and acceptable solutions by at least a practically significant amount δ . One of the presented algorithms provides a statistical guarantee that may be somewhat less stringent.

3 Algorithms

In this section, we introduce the setting and notation, which are similar to that of Tsai (2013). We also present three rapid screening algorithms in detail for the zero-one OvS problem considering a single stochastic constraint.

3.1 Setting and notation

The proposed algorithms consist of R screening iterations $0, 1, \dots, R - 1$ and one final selection iteration R . Our notation uses “ (r) ” to indicate quantities defined at the r th iteration. At each screening iteration r of the algorithm, there are $b(r)$ new solutions generated from the solution space (whose feasibility has not been identified yet). We use the set $M(r)$ to represent the candidate solutions that need to be evaluated with the feasibility check procedure at iteration r . The detailed definition of $M(r)$ varies depending on which version of the algorithm we are using. We also use $I(r)$ to denote the set of surviving solutions through iteration r , and we use $F(r)$ to represent the set of feasible solutions identified from $M(r)$ or from all visited solutions (which also depends on which algorithm we use). Let $N_i(r)$ be the number of replications allocated to solution S_i at objective screening of iteration r . If S_i is a newly generated solution, we then use $N_i(r) = \lceil N_0 G^r \rceil$, where N_0 represents the number of initial replications and G is a constant growth factor. If S_i is a surviving solution (i.e., $i \in I(r - 1)$), we then use $N_i(r) = N_i(r - 1) + \lceil N_0 G^r \rceil$ (i.e., the accumulated

Table 1 Summary of notations defined in Sect. 3.1

Parameters	Functionality
R	Number of screening iterations
G	Growth factor of sample size
$b(r)$	Number of new solutions generated at r th iteration
$M(r)$	Set of solutions evaluated by FCP at r th iteration
$F(r)$	Set of feasible solutions identified by FCP at r th iteration
$I(r)$	Set of surviving solutions through r iterations
$N_i(r)$	Sample size allocated to S_i at r th iteration
$W_{i\ell}(r)$	Screening threshold for S_i and S_ℓ at r th iteration

observations) to compute the sample mean $\bar{X}_i(N_i(r))$ and variance $S_i^2(N_i(r))$ for the objective performance. We assume that each candidate solution is simulated independently (i.e., common random numbers are not used), and in this case, it is more appropriate to reuse the observations collected previously for the objective screening of the surviving solutions [see Tsai (2013) for a detailed discussion]. We let the number of allocated replications grow at a constant geometric rate to yield tighter screening thresholds, which results in more effective elimination of inferior solutions in later iterations. For the objective screening stage, we let $W_{i\ell}(r)$ represent the screening threshold for the comparison between solutions S_i and S_ℓ at iteration r , which is a function of t critical values and the sample variance of their objective performance. For the final selection stage, we have to obtain additional replications from surviving solutions, and the total number of observations for solution S_i (denoted as N_i) is determined by Rinott's (1978) constant h to deliver the desired statistical validity. Typically, the constant h is determined by the number of solutions being evaluated, the desired confidence level, and the number of observations used to compute the variance estimator [see Nelson et al. (2001) for a discussion]. A summary of all the aforementioned notations is presented in Table 1.

3.2 Algorithm A

The first proposed approach represents an intuitive way to incorporate the feasibility check procedure into rapid-screening type algorithms when encountering a stochastic constraint. This approach is denoted as Algorithm A and consists of R screening iterations $0, 1, \dots, R-1$ and one final selection iteration R . In each screening iteration r , we want to identify the feasibility of every newly-generated solution with a specified confidence level, and then return the set of feasible solutions $F(r)$. Then the screening procedure with respect to the objective performance is implemented upon the union of the feasible solutions of the current iteration (i.e., $F(r)$) and the surviving solutions through iteration $r-1$ (i.e., $I(r-1)$). In this case, we can simply employ the ordinary feasibility check procedure (FCP) of Andradóttir and Kim (2010) that requires an equal initial sample size since it is applied to these $b(r)$ new solutions (therefore, no previous samples are retained). If there is more than one solution left at the end of the screening iterations (i.e., iteration $R-1$), we then apply the selection procedure due to Boesel et al. (2003) to pick up the best solution in terms of the objective performance among the solutions in the remaining set $I(R-1)$. This algorithm can be proven to select the best feasible solution among those candidate solutions visited by all the R iterations (i.e., $\sum_{r=0}^{R-1} b(r)$) with a pre-specified confidence level $1 - \alpha$. A flowchart

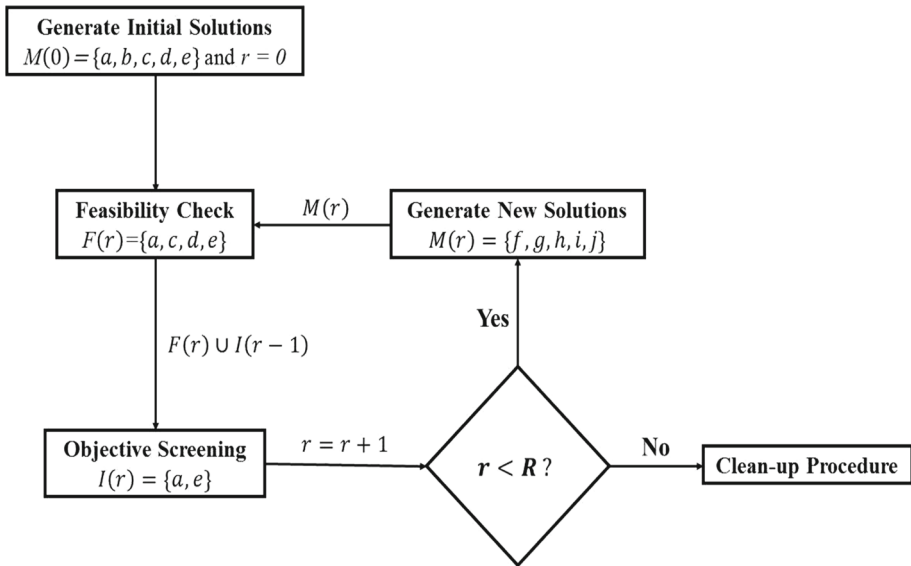


Fig. 1 Flowchart of Algorithm \mathcal{A}

of Algorithm \mathcal{A} is given as Fig. 1. A detailed description of Algorithm \mathcal{A} and a theorem regarding its statistical guarantee are presented as follows:

Algorithm A

Setup: Choose the desired overall confidence level $0 < 1 - \alpha < 1$. Determine $\alpha_1, \alpha_2, \alpha_s,$ and α_f such that $\alpha_1 + \alpha_2 = \alpha$ and $(1 - \alpha_s)(1 - \alpha_f) = 1 - \alpha_1$. Specify the indifference-zone parameter $\delta,$ the constraint tolerance level $\epsilon,$ the number of iterations $R,$ the initial sample size n_0 (for feasibility check), the number of initial replications N_0 (for objective screening), and the constant growth factor $G \geq 1$. Set the iteration number $r = 0$ and $I(-1) = \emptyset$.

Initialization: Generate $b(0)$ initial solutions and let $M(0) = \{1, 2, \dots, b(0)\}$. For all $i \in M(0),$ compute the sample mean $\bar{Y}_i(n_0)$ and the sample variance $S_i^2(n_0)$ with respect to the constraint performance.

Feasibility Check Stage: For all $i \in M(r),$ apply the feasibility check procedure of [Andradóttir and Kim \(2010\)](#) (see Appendix A.3 and Remark A.1). Note that we use α_f, n_0 and $k = \sum_{r=0}^{R-1} b(r)$ to compute the value of η . Return $F(r)$ as the set of feasible solutions (identified from $M(r)$). If $|\{F(r) \cup I(r - 1)\}| \leq 1,$ we go to the **Stopping Rule Stage**. Otherwise (i.e., $|\{F(r) \cup I(r - 1)\}| > 1,$) we go to the following **Objective Screening Stage**.

Objective Screening Stage:

Step 1 For all $i \in \{F(r) \cup I(r - 1)\},$ take $\lceil N_0 G^r \rceil$ replications from solution S_i . If $i \in I(r - 1),$ we let $N_i(r) = N_i(r - 1) + \lceil N_0 G^r \rceil$. Else if $i \in F(r),$ we let $N_i(r) = \lceil N_0 G^r \rceil$.

Step 2 For all $i, \ell \in \{F(r) \cup I(r - 1)\}$ and $i \neq \ell$, compute the sample mean $\bar{X}_i(N_i(r))$, the sample variance $S_i^2(N_i(r))$, and the screening threshold $W_{i\ell}(r)$ (with respect to the objective performance) as follows:

$$W_{i\ell}(r) = \sqrt{\frac{t_i^2 S_i^2(N_i(r))}{N_i(r)} + \frac{t_\ell^2 S_\ell^2(N_\ell(r))}{N_\ell(r)}}$$

where $t_i = t_{(1-\alpha')^{1/(\sum_{z=0}^r |F(z)|-1)}, N_i(r)-1}$ and $\alpha' = \alpha_s/R$.

Step 3 Form the subset

$$E(r) = \{i \in \{F(r) \cup I(r - 1)\} : \exists \ell \in \{F(r) \cup I(r - 1)\}, \bar{X}_i(N_i(r)) - \bar{X}_\ell(N_\ell(r)) \leq -W_{i\ell}(r), \ell \neq i\},$$

and update $I(r) = \{F(r) \cup I(r - 1)\} \setminus E(r)$.

Stopping Rule: Let $r = r + 1$. If $r < R$, then generate $b(r)$ new solutions; let $M(r) = \left\{ \sum_{j=0}^{r-1} b(j) + 1, \sum_{j=0}^{r-1} b(j) + 2, \dots, \sum_{j=0}^r b(j) \right\}$, and go back to **Feasibility**

Check Stage. Otherwise (i.e., $r = R$), let $B = \sum_{r=0}^{R-1} |F(r)|$, and go to the following **Selection Stage.**

Selection Stage: If $|I(R - 1)| = 0$, declare that no feasible solution has been found in these R screening iterations. If $|I(R - 1)| = 1$, then declare that solution as the best. Otherwise, compute the total number of replications N_i , for all $i \in I(R - 1)$, as follows:

$$N_i = \max \left\{ N_i(R - 1), \left\lceil \left(\frac{h S_i(N_i(R - 1))}{\delta} \right)^2 \right\rceil \right\}$$

where $h = h(2, (1 - \alpha_2)^{1/(B-1)}, n_{\min})$ is the extended Rinott’s (1978) constant and $n_{\min} = \min_{i \in I(R-1)} \{N_i(R - 1)\}$. If $N_i(R - 1) < N_i$, then obtain $N_i - N_i(R - 1)$ additional replications from all solutions $S_i, i \in I(R - 1)$. Select the solution with the largest sample mean $\bar{X}_i = (1/N_i) \sum_{j=1}^{N_i} X_{ij}$ as the best.

Theorem 1 *If the BN assumption of (X_{ij}, Y_{ij}) holds, then Algorithm A selects solution $S_{[k]}$ with a probability $\geq 1 - \alpha$ whenever its objective performance is better than other desirable and acceptable solutions by at least a practically significant amount δ .*

It should be noticed that most details of Algorithm A are designed to maintain its provable validity (see Appendix A.1 for the proof). For instance, in the Setup stage, the overall allowable error α is split into α_1 (for incorrect selection on R screening iterations) and α_2 (for incorrect selection on the final selection iteration). We also follow a multiplicative rule to specify α_s (for objective screening) and α_f (for feasibility check). The allowable error α_s is then divided equally into R screening iterations (see Step 2 of the Objective Screening stage). In the Feasibility Check stage, we must use the number of total solutions visited by FCP in all R iterations (i.e., $k = \sum_{r=0}^{R-1} b(r)$) to compute the monitoring statistic in each iteration. In this way, the feasibility guarantee for all R iterations can thus be simultaneously maintained. In many cases, it is possible to specify or conjecture an upper bound on the number of solutions that can be simulated in advance (see Sect. 4 for an illustration). By contrast, in the Objective Screening stage, the screening statistic should depend on the accumulated number of solutions that has ever entered this stage through iteration r (i.e., $\sum_{z=0}^r |F(z)|$). This setting is due to the fact that we reuse the sample information with respect to the objective performance collected previously for the surviving solutions (i.e., use $N_i(r)$ samples).

Another important feature of Algorithm \mathcal{A} is that the observations obtained in the Feasibility Check stage are discarded thereafter. That is, we need to restart the sampling process in the Objective Screening stage (i.e., perform new $\lceil N_0 G^r \rceil$ simulation replications) in order to avoid the dependency between these two stages. Finally, in the Selection stage, we use the total number of feasible solutions identified by FCP (i.e., $\sum_{r=0}^{R-1} |F(r)|$) to compute the extended Rinnott’s (1978) constant, which is similar to the setting adopted by Tsai (2013). We also use the accumulated observations collected in all R Objective Screening stages (with sample size $N_i(R - 1)$) when determining the number of total required replications. The advantage of Algorithm \mathcal{A} is that it represents an intuitive combination (and simple modification) of existing procedures under a carefully designed algorithm framework. The resulting algorithm can be proven to deliver the desired statistical validity. The disadvantage is that it may require excessive sampling effort due to the necessary parameter settings and sampling mechanisms mentioned above. In addition, Algorithm \mathcal{A} is especially not desirable when there are a number of solutions whose constraint means are close to the specified boundary, and in which they meanwhile have uncompetitive expected objective performance. In this case, we need to spend very large sampling budget to determine their feasibility (with a given accuracy), but afterward they might easily be eliminated in the Objective Screening stage. Therefore the sampling during the Feasibility Check stage is quite wasteful and does not contribute much to the generation process intended to lead to potential promising solutions.

3.3 Algorithm B

In this subsection, we propose a more heuristic-oriented algorithm (denoted as Algorithm \mathcal{B}) that does not require the application of FCP to every sampled solution. A critical component of Algorithm \mathcal{B} is the constraint screening procedure (denoted as CSP) that is an extension of the so-called comparison-with-a-standard procedure (see Tsai and Chu (2012) for a detailed introduction). That is, the constraint threshold Q is treated as a known standard and we want to find a subset that contains all the solutions that are better than the standard by at least a significant amount ϵ in terms of the expected constraint performance (i.e., the desirable solutions defined in Sect. 2). Suppose that at iteration r , we have a set of candidate solutions $I(r)$ to be evaluated (i.e., $|I(r)| = k$), and let $N_i(r)$ denote the accumulated sample size, for all $i \in I(r)$. We now describe CSP in detail and then provide a theorem regarding its statistical guarantee. The derivation is relegated to Appendix A.2.

Constraint Screening Procedure

Step 1 Specify the desired confidence level $0 < 1 - \alpha_c < 1$ and the tolerance level ϵ .

Step 2 For all $i \in I(r)$, compute the sample mean $\bar{Y}_i(N_i(r)) = (1/N_i(r)) \sum_{j=1}^{N_i(r)} Y_{ij}$, the sample variance $S_i^2(N_i(r)) = (1/(N_i(r) - 1)) \sum_{j=1}^{N_i(r)} [Y_{ij} - \bar{Y}_i(N_i(r))]^2$, and the screening threshold W_{i0} as follows:

$$W_{i0} = h\sqrt{S_i^2(N_i(r))/N_i(r)}.$$

The setting of h must satisfy the condition $\Pr\{H \leq h\} = 1 - \alpha_c$, where $H = \max\{T_1, T_2, \dots, T_k\}$ and T_1, T_2, \dots, T_k are student’s t random variables with $N_i(r) - 1$ degrees of freedom. We follow a proven procedure of Nelson and Goldman (2001) to obtain an efficient estimator of h .

Step 3 Form the subset

$$I'(r) = \{i : \bar{Y}_i(N_i(r)) - Q \leq (W_{i0} - \epsilon)^+, \forall i \in I(r)\},$$

where $y^+ = \max\{0, y\}$.

Theorem 2 *Under the assumption that the simulation output data Y_{ij} are normally distributed, in each iteration r , our constraint screening procedure provides the following statistical guarantee:*

$$\Pr\{S_D(r) \subseteq I'(r)\} \geq 1 - \alpha_c \text{ where } S_D(r) = \{i : Y_i - Q \leq -\epsilon, \forall i \in I(r)\}.$$

Each screening iteration of Algorithm \mathcal{B} is composed of a Constraint Screening stage and an Objective Screening stage. A specified amount of simulation replications is taken to obtain observations of (X_{ij}, Y_{ij}) for use in both the Constraint Screening and Objective Screening stages, which can significantly reduce the simulation budget. The accumulated observations with respect to Y_{ij} are used to employ CSP to eliminate inferior solutions (with respect to the constraint performance) and then return a subset $I'(r)$. Then, the simulated samples of X_{ij} collected from the same replications are utilized in the following Objective Screening stage, in order to further reduce the subset size (based on the objective performance). We then generate new solutions from the neighborhood of the remaining solutions and go on to the next screening iteration. After implementing these R iterations, we first identify the feasibility of all the surviving solutions with a specified statistical guarantee and then apply the clean-up procedure due to Boesel et al. (2003) to choose the best solution in terms of the objective performance among the remaining feasible solutions. A flowchart of Algorithm \mathcal{B} is given as Fig. 2. Notice that in this case, the number of observations already taken from each solution should be unequal, which implies that the ordinary FCP of Andradóttir and Kim (2010) is not readily applicable. Therefore, we propose an alternative version of FCP that can handle unequal initial sample sizes and exploit the previously taken observations, in a similar manner as in Pichitlamken et al. (2006), and prove the following theorem in Appendix A.3.

Theorem 3 *Under the assumption that the simulation output data Y_{ij} are normally distributed, the feasibility check procedure with unequal sample sizes provides the following statistical guarantee:*

$$\Pr\{\mathcal{R}_D \subseteq F \subseteq (\mathcal{R}_D \cup \mathcal{R}_A)\} \geq 1 - \alpha_f.$$

We propose two versions of Algorithm \mathcal{B} (i.e., \mathcal{B} -1 and \mathcal{B} -2), and the specific steps of Algorithm \mathcal{B} -1 are first described as follows:

Algorithm B-1

Setup: Choose the number of iterations R . Determine the allowable error corresponding to constraint screening (α_c), objective screening (α_s), feasibility check (α_f), and selection of the best (α_2). Specify the indifference-zone parameter δ , the constraint tolerance level ϵ , the number of initial replications N_0 (for both constraint and objective screening), the minimum initial sample size n_0 (for feasibility check with unequal sample size), and the constant growth factor $G \geq 1$. Set the iteration number $r = 0$.

Initialization: Generate $b(0)$ initial solutions and let $I(0) = \{1, 2, \dots, b(0)\}$. For all $i \in I(0)$, take N_0 initial replications from solution \mathbf{S}_i (to obtain (X_{ij}, Y_{ij})), and let $N_i(0) = N_0$.

Constraint Screening Stage: For all $i \in I(r)$, we apply the aforementioned constraint screening procedure to obtain $I'(r)$. If $|I'(r)| \leq 1$, we let $I(r + 1) = I'(r)$ and go to

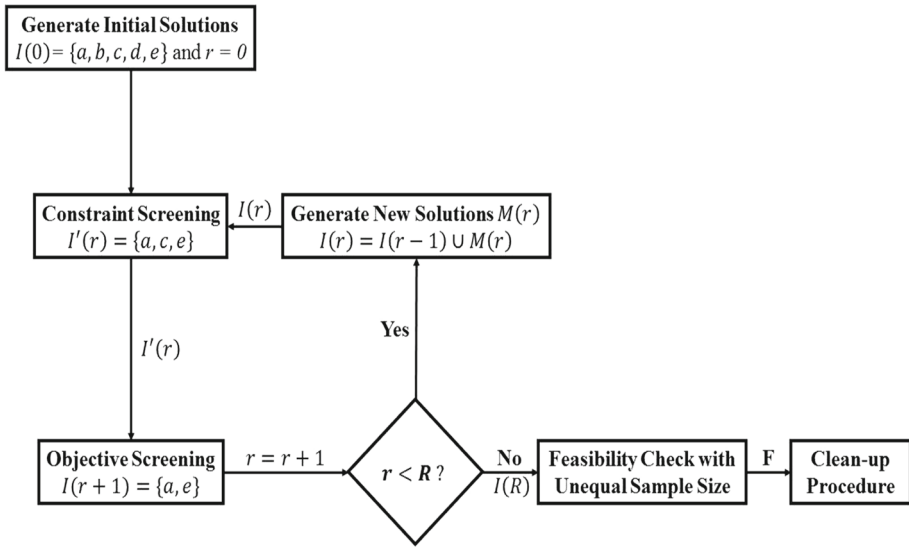


Fig. 2 Flowchart of Algorithm B

the **Stopping Rule Stage**. Otherwise (i.e., $|I'(r)| > 1$), we go to the following **Objective Screening Stage**.

Objective Screening Stage: For all $i \in I'(r)$, we follow the same steps as in the objective screening stage of Algorithm A, except that we do not perform any new simulations (to obtain X_{ij}), and the critical value for screening is now $t_i = t_{(1-\alpha')^{1/(|I'(r)|-1)}, N_i(r)-1}$, where $\alpha' = \alpha_s/R$. We also update $I(r+1) = I'(r) \setminus E(r)$.

Stopping Rule: Let $r = r + 1$. If $r < R$, then generate $b(r)$ new solutions. We take $\lceil N_0 G^r \rceil$ replications from all remaining solutions. If $i \in I(r)$, we let $N_i(r) = N_i(r-1) + \lceil N_0 G^r \rceil$. Else if index i belongs to the set of $b(r)$ new solutions, we let $N_i(r) = \lceil N_0 G^r \rceil$. Add the set of $b(r)$ solutions to $I(r)$, and go back to the **Constraint Screening Stage**. Otherwise (i.e., $r = R$), let $M = I(R)$ and go to the following **Feasibility Check Stage**.

Feasibility Check Stage: If $|M| = 0$, declare that no feasible solution has been found in these R screening iterations. Otherwise, for all $i \in M$, we apply the feasibility check procedure with unequal sample size (see Appendix A.3). Note that we use $n_i = N_i(R-1)$ and $k = |M|$. If $|F| = 0$, declare that no feasible solution has been found in these R screening iterations. Otherwise, return F as the set of feasible solutions, and go to the following **Selection Stage**.

Selection Stage: If $|F| = 1$, then declare that solution as the best. Otherwise, for all $i \in F$, we follow the same steps as in the selection stage of Algorithm A, except that we use $B = |F|$ to compute the extended Rinnott's (1978) constant h .

The advantage of this algorithm is that we only need to implement FCP once in the final stage, and it can utilize simulation observations accumulated in previous R iterations. In other words, FCP is applied to those solutions that have competitive objective performance since they have survived through previous Objective Screening stages (which implies that the feasibility check is worthwhile). This seems to be a more reasonable and efficient scheme compared to Algorithm A, where FCP is applied to any newly generated solutions. Further,

the sampled observations of X_{ij} and Y_{ij} collected from the same replications can be both exploited in the screening iterations (in Algorithm \mathcal{A} they are sampled from different replications), which helps save a significant amount of the sampling budget. Another desirable setting is that we take a number of observations and keep updating the variance estimators of CSP at each Constraint Screening stage, so that eliminations are made more and more efficiently as the optimization search progresses. The variance-updating multistage procedure can be shown to be more efficient (in terms of the required sampling efforts) than the ordinary fully sequential procedure, such as FCP, which takes a number of preliminary samples to estimate the variance and fixes it in the following steps [see Tsai and Chu (2012) for a detailed argument]. It should also be noticed that in the Constraint Screening stage, we can guarantee that the returned subset contains all the desirable solutions, but this subset can also include some acceptable or unacceptable solutions. Therefore, it is possible that feasible solutions with worse objective performance are eliminated by infeasible solutions having superior objective performance in the following Objective Screening stage. As a result, Algorithm \mathcal{B} -1 is a heuristic in the sense that it cannot deliver the overall probability of correct selection for the solution finally chosen (i.e., $\text{PCS}[I] \geq 1 - \alpha$ defined in Sect. 2). We thus do not need to prescribe how to allocate the overall confidence level $1 - \alpha$ in the Setup stage. However, in the empirical study, an additive rule of error allocation is served as a heuristic way. Subsequently, we make slight modifications to Algorithm \mathcal{B} -1 and propose Algorithm \mathcal{B} -2 to improve the PCS guarantee, but the algorithm's statistical efficiency might be sacrificed. More specifically, in the Stopping Rule stage, we now let $M = \left\{ \bigcup_{r=0}^{R-1} E(r) \right\} \cup I(R)$ immediately after consuming all R screening iterations (i.e., when $r = R$). In other words, in the Feasibility Check stage, Algorithm \mathcal{B} -2 considers not only the remaining solutions $I(R)$ (through all R iterations), but also those solutions that survived the Constraint Screening stage and were eliminated in the subsequent Objective Screening stage (i.e., $E(r)$). Further, we employ the accumulated simulation observations and use the number of total solutions visited in all R iterations (i.e., $k = \sum_{r=0}^{R-1} b(r)$) to compute the monitoring statistic of FCP. In addition, after the feasibility check, we implement one more objective screening for those feasible solutions based on previously obtained observations. Then, we keep the surviving solutions and go to the final Selection stage. In the Setup stage of Algorithm \mathcal{B} -2, we now need to specify the confidence levels as follows: $R \times \alpha_c + \alpha_f + \alpha' + \alpha_2 = \alpha$, where α' represents the allowable error used in the last objective screening (which is also the error applied in the objective screening of each iteration). It can be anticipated that in these R screening iterations, the number of simulated observations expended by Algorithm \mathcal{B} -1 (or \mathcal{B} -2) is less than that of Algorithm \mathcal{A} , while Algorithm \mathcal{B} -2 considers many more candidate solutions in the final stage (compared to Algorithm \mathcal{A} and Algorithm \mathcal{B} -1). This setting will definitely lead to an increase in the total required sampling budget of Algorithm \mathcal{B} -2 (when compared to Algorithm \mathcal{B} -1), but the statistical validity can thus be maintained to some extent. See Appendix A.4 for the proof of Theorem 4.

Theorem 4 *If the BN assumption of (X_{ij}, Y_{ij}) holds, then Algorithm \mathcal{B} -2 guarantees that $\Pr\{[k] \in F\} \geq 1 - R \times \alpha_c - \alpha_f$. We conjecture that $\Pr\{\text{select } \mathbf{S}_{[k]} | [k] \in F\} \geq 1 - \alpha' - \alpha_2$ whenever its objective performance is better than other desirable and acceptable solutions by at least a practically significant amount δ , but have been unable to prove it.*

To deliver the overall probability of correct selection (i.e., $\text{PCS}[I] \geq 1 - \alpha$) of Algorithm \mathcal{B} -2, we can restart the sampling process (i.e., without using the previous simulation observations) of the last objective screening procedure (right after the feasibility check), but it may

incur highly excessive simulation sampling effort. The disadvantage is more obvious when there are many remaining solutions after the Feasibility Check stage.

3.4 Algorithm C

In this subsection, we introduce an algorithm that allows the ability to perform objective screening to candidate solutions whose feasibility has not been determined yet (which is similar to Algorithm B), and in the meanwhile, the desired statistical guarantee is provided. We use the notation SS_i to indicate the set of solutions that are superior to solution S_i in terms of the objective performance, and this setting is adopted from [Andradóttir and Kim \(2010\)](#). Each screening iteration of Algorithm C is composed of a One-Time Feasibility Check stage and an Objective Screening stage. Similar to Algorithm B, in each iteration, we take a pre-specified number of simulation replications (i.e., $\lceil N_0 G^r \rceil$) from each remaining solution and then implement the feasibility check and objective screening in sequence. The feasibility check is different from that of Algorithm A in that its decision is based on the accumulated observations (it could be feasible, infeasible, or undetermined), rather than continuing the sequential sampling process until the feasibility has been identified. In other words, we use the accumulated observations to compute the monitoring statistic and examine one time whether a feasibility decision can be made. In order to maintain the feasibility guarantee for all R iterations, we must use the number of total solutions visited by FCP in all R iterations (i.e., $k = \sum_{r=0}^{R-1} b(r)$) in order to compute the monitoring statistic in each iteration, which is the same setting as in Algorithm A and Algorithm B-2. If solution S_i is determined to be feasible, we then eliminate all ℓ (and the corresponding set SS_ℓ) that satisfy $i \in SS_\ell$. If solution S_i is determined to be infeasible, we then delete the index i and the corresponding set SS_i . This is called a one-time feasibility check. In the follow-up Objective Screening stage, we do not need to restart the sampling process to collect new observations with respect to objective performance (i.e., screening is based on accumulated $N_i(r)$ observations). If solution S_i is eliminated by a feasible solution S_ℓ in terms of the objective performance, we then delete index i and the corresponding set SS_i ; otherwise, if the feasibility of S_ℓ is not determined yet, we then add the index ℓ to the set SS_i . After implementing these R iterations, we first identify the feasibility of the surviving solutions whose feasibility has not been determined yet (see Appendix A.3) and then apply the clean-up procedure to choose the best solution among the feasible solutions. It should be noticed that in the final Feasibility Check stage, if solution S_j is determined to be feasible, we then directly delete all solutions S_ℓ that satisfy $i \in SS_\ell$ (without performing the feasibility check), which can also further reduce the sampling cost incurred in the following Selection stage. A flowchart of Algorithm C is given in Fig. 3. A detailed description of Algorithm C and a theorem regarding its statistical guarantee are presented as follows:

Algorithm C

Setup: Choose the desired overall confidence level $0 < 1 - \alpha < 1$. Determine the allowable error corresponding to objective screening (α_s), feasibility check (α_f), and selection of the best (α_2), such that $\alpha_s + \alpha_f + \alpha_2 = \alpha$. Specify the indifference-zone parameter δ , the constraint tolerance level ϵ , the number of iterations R , the number of initial replications N_0 , and the constant growth factor $G \geq 1$. Set the iteration number $r = 0$ and $F(0) = \emptyset$.

Initialization: Generate $b(0)$ initial solutions and let $M(0) = \{1, 2, \dots, b(0)\}$. For all $i \in M(0)$, take N_0 initial replications from solution S_i (to obtain (X_{ij}, Y_{ij})) and let $N_i(0) = N_0$.

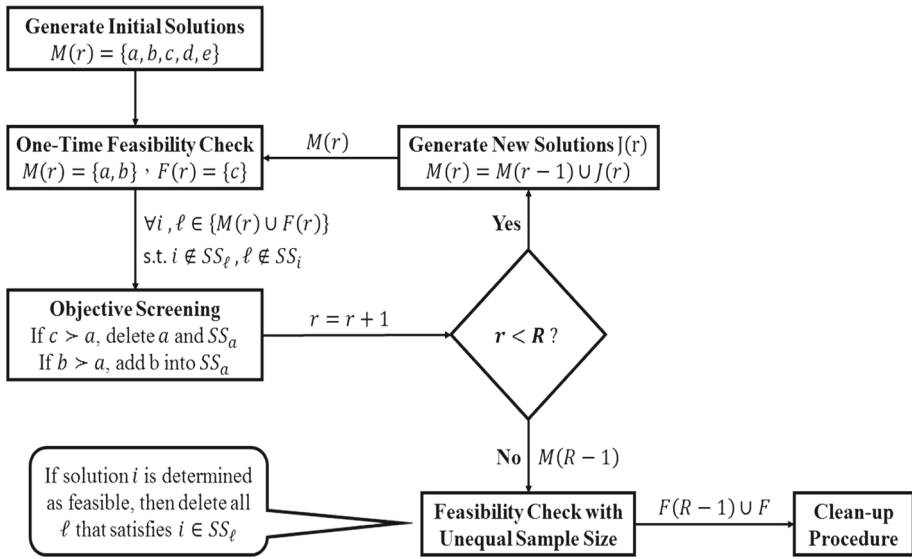


Fig. 3 Flowchart of Algorithm C

Let $SS_i = \emptyset$ be the set of superior solutions (compared to S_i) in terms of expected objective performance $X(S_i)$.

One-Time Feasibility Check Stage: For all $i \in M(r)$, based on the accumulated $N_i(r)$ observations, we apply the feasibility check procedure with unequal sample size (see Appendix A.3). If solution S_i is determined to be feasible, we then move i from $M(r)$ to $F(r)$ and delete all ℓ (from $M(r)$ or $F(r)$) that satisfy $i \in SS_\ell$. Else if S_i is determined to be infeasible, we then delete i from $M(r)$ and eliminate the set SS_i .

Objective Screening Stage: For each $i, \ell \in \{M(r) \cup F(r)\}$ such that $i \neq \ell, i \notin SS_\ell, \ell \notin SS_i$, we follow the same steps as in Algorithm B to compute the screening threshold $W_{i\ell}(r)$, and then check whether the following condition holds:

$$\bar{X}_i(N_i(r)) - \bar{X}_\ell(N_\ell(r)) \leq -W_{i\ell}(r).$$

If the condition holds and $\ell \in F(r)$, we then eliminate i and SS_i ; else if the condition holds, and $\ell \in M(r)$, we then add ℓ to SS_i .

Stopping Rule: Let $r = r + 1$. If $r < R$, then generate $b(r)$ new solutions, and let $J(r) = \left\{ \sum_{j=0}^{r-1} b(j) + 1, \sum_{j=0}^{r-1} b(j) + 2, \dots, \sum_{j=0}^r b(j) \right\}$. We update $M(r) = \{M(r-1) \cup J(r)\}$ and $F(r) = F(r-1)$. We then take $\lceil N_0 G^r \rceil$ replications from all solutions of $M(r)$ and $F(r)$. If index i belongs to the set $J(r)$, we let $N_i(r) = \lceil N_0 G^r \rceil$. Otherwise, we let $N_i(r) = N_i(r-1) + \lceil N_0 G^r \rceil$. Go back to **One-Time Feasibility Check Stage**. Else if $r = R$, go to the following **Feasibility Check Stage**.

Feasibility Check Stage: If $|M(R-1)| = 0$ and $|F(R-1)| = 1$, then declare that solution as the best. If $|M(R-1)| > 0$, then we apply the feasibility check procedure with unequal sample size (see Appendix A.3) to all $i \in M(R-1)$. In the implementation process of FCP, if solution S_i is determined to be feasible, we then delete all ℓ (from $M(R-1)$ or $F(R-1)$) that satisfy $i \in SS_\ell$. Return F as the set of feasible solutions identified by FCP and update

$F = \{F(R - 1) \cup F\}$. If $|F| = 0$, declare that no feasible solution has been found in these R screening iterations. Otherwise, return F as the set of feasible solutions, and go to the following **Selection Stage**.

Selection Stage: If $|F| = 1$, then declare that solution as the best. Otherwise, for all $i \in F$, we follow the same steps as in the selection stage of Algorithm B to select the best solution.

Theorem 5 *If the BN assumption of (X_{ij}, Y_{ij}) holds, then Algorithm C selects solution $S_{[k]}$ with probability $\geq 1 - \alpha$ whenever its objective performance is better than other desirable and acceptable solutions by at least a practically significant amount δ .*

In each iteration of Algorithm C , we continue updating $M(r)$ and $F(r)$, which represents the set of solutions whose feasibility have not been determined yet and the set of feasible solutions, respectively. In the Objective Screening stage, we eliminate candidate solutions only when they are obviously inferior to any feasible solution among $F(r)$ (there may be only a few of them). This implies that in Algorithm C , it is more difficult to eliminate solutions as compared to the other algorithms. Therefore, when given the same amount of simulation budget, the total number of evaluated solutions in the searching process of Algorithm C will be smaller than that of other algorithms, and this might result in a final solution with inferior objective performance. In addition, Algorithm C is more favorable when there exists some feasible solution that is very competitive in terms of objective performance where, in the meantime, the determination of its feasibility is not very difficult. In this case, if at least one of the good solutions is found in early iterations (i.e., add into $F(r)$), then we can use them to eliminate a large number of uncompetitive solutions and thus deliver a great algorithm performance.

4 Empirical results

In this section, we conduct an extensive experimental evaluation to compare the following algorithms for the purpose of solving the zero-one OvS problem subjected to a single stochastic constraint:

1. The rapid screening algorithms considering a single stochastic constraint presented in Sect. 3 are performed, including Algorithm A ($\mathcal{RSC-A}$), Algorithm $B-1$ ($\mathcal{RSC-B1}$), Algorithm $B-2$ ($\mathcal{RSC-B2}$), and Algorithm C ($\mathcal{RSC-C}$).
2. [Andradóttir and Kim \(2010\)](#) proposed a fully sequential procedure (AK) that can select the best solution in the presence of a single stochastic constraint. AK assumes that the set of solutions is available at the beginning of the experiment and requires excessive simulated samples if the number of candidate solutions is large.
3. [Lee et al. \(2012\)](#) developed a new optimal computing budget allocation (OCBA) algorithm for stochastically constrained problems. The goal is to maximize the posterior PCS under a given sampling budget. Notice that the OCBA does not provide a guaranteed PCS, but it might be the most efficient R&S procedure to achieve an approximate PCS (see [Branke et al. 2007](#)).

In the experiments, the AK algorithm is applied to the whole solution space to serve as a benchmark for comparison, which represents a very conservative way to achieve a global PCS guarantee. By contrast, our rapid screening algorithms can provide a PCS guarantee among the visited candidate solutions, and meanwhile, they hopefully consume an affordable amount of simulated replications. We also consider OCBA in the experimental study because

the philosophy of ordinal optimization (see [Ho et al. 2000](#)) matches with that of our rapid screening algorithms [see [Tsai \(2013\)](#) for a discussion].

4.1 Configurations and experiment design

To obtain more control over the factors that affect the performance of the algorithms, we consider several known response-surface functions to which we add normally distributed noise (or some components of the function serve as normal random variables). We assume that each candidate solution is simulated independently (i.e., CRN are not used). The competing algorithms are implemented without knowing the explicit structure of the expected objective and constraint functions. The first test problem is analogous to the zero-one stochastic knapsack problem (e.g., [Kosuch and Lisser 2010](#)). Given d items, each having a prespecified profit per weight unit $p_j > 0$, a random weight w_j ($j = 1, 2, \dots, d$), and a capacity Q , the knapsack problem is intended to choose a subset of items that has maximum expected total profit and an expected total weight not exceeding Q . We assume that the random weight w_j is independently normally distributed with mean μ_j and standard deviation σ_j . The decision of which items are collected has to be made without knowing their weights with certainty. The constrained problem can be formulated as follows:

$$\begin{aligned} \max \quad & \mathbf{X}(\mathbf{S}_i) = \mathbb{E} \left[\sum_{j=1}^d w_j p_j S_{ij} \right] \\ \text{s.t.} \quad & \mathbf{Y}(\mathbf{S}_i) = \mathbb{E} \left[\sum_{j=1}^d w_j S_{ij} \right] \leq Q \end{aligned}$$

where S_{ij} is a binary variable taking the value 1 if and only if item j is selected for the i th solution \mathbf{S}_i . We set $d = 10$, and therefore, we have a total of $2^{10} = 1024$ candidate solutions. The profit per weight unit p_j is set as (2, 2, 3, 2, 2, 1, 2, 2, 1, 2), for $j = 1, 2, \dots, 10$. The expected value μ_j is set as (62, 53, 96, 73, 89, 83, 66, 77, 60, 58), for $j = 1, 2, \dots, 10$. The standard deviation σ_j of random weight is specified as (8, 10, 5, 4, 2, 4, 8, 5, 9, 18) and (13, 15, 10, 9, 7, 9, 13, 10, 14, 23), representing a low or high level of output variability, respectively. When implementing the proposed rapid screening algorithms, we set the indifference-zone parameter (for the objective performance) as $\delta = 20$ and specify the tolerance level (for the constraint performance) as $\epsilon = 15$. To investigate the effect of different proportions of feasible solutions on the algorithm performance, we use different values of Q ($= 260, 370, 460$) in the experiments. For the case of $Q = 260$, we have 200 feasible solutions, which accounts for 19.5% of the solution space and is regarded as a scenario with a low level percentage of feasible solutions in the entire space. In this case the optimal feasible solution is (0, 0, 1, 0, 0, 0, 0, 0, 0, 1) with an objective value of 612. For the case of $Q = 370$, we have 559 feasible solutions (accounting for 54.5% of the solution space), which represents a scenario with a medium percentage of feasible solutions. The optimal feasible solution is located at (0, 0, 1, 1, 0, 0, 1, 1, 0, 1) with an objective value of 836. For the last case of $Q = 460$, we have 832 feasible solutions, which accounts for 81% of the solution space and can be viewed as a scenario with a high percentage of feasible solutions. In this case the optimal feasible solution is located at (0, 0, 1, 1, 1, 0, 1, 1, 0, 1) with an objective value of 1014.

The second evaluated function of the response surface is known as the constrained binary quadratic problem (CBQP), where the objective is represented as a binary quadratic function (see [Barahona et al. 1989](#)):

$$X(S_i) = S_i^T U S_i,$$

where

$$U = \begin{bmatrix} -2 & -1 & -2 & -2 & 2 & -1 & 2 & 2 & -3 & 2 & 2 & -1 \\ -3 & -1 & 8 & -3 & -1 & -2 & 2 & 5 & 5 & -2 & 4 & 6 \\ -1 & -2 & -1 & 1 & -2 & -3 & 3 & 5 & 6 & 5 & 3 & 2 \\ 1 & 2 & 1 & -2 & 2 & -2 & 2 & -2 & -3 & -1 & -2 & -2 \\ 2 & 4 & 1 & 6 & 2 & 1 & -2 & -1 & -2 & -1 & 3 & -1 \\ 5 & -2 & 8 & -2 & 6 & 2 & -1 & -2 & 1 & -1 & 1 & -1 \\ -2 & 2 & 1 & 2 & 3 & 5 & -1 & 8 & 2 & -1 & 8 & 5 \\ -2 & 8 & 5 & -3 & -1 & -1 & 2 & -2 & 3 & -1 & 6 & 2 \\ -1 & -1 & 4 & 2 & 2 & 2 & 1 & 6 & -3 & -1 & -1 & 1 \\ -2 & 1 & 1 & -1 & -2 & -2 & -1 & 3 & -1 & -2 & 2 & 2 \\ 5 & 8 & 6 & -1 & 1 & 1 & 2 & 3 & -1 & -2 & 6 & 1 \\ -1 & 2 & -2 & 3 & 4 & 5 & 5 & -4 & -3 & 2 & -2 & 5 \end{bmatrix}.$$

We set $d = 12$, and therefore, we have a total of $2^{12} = 4096$ candidate solutions. The stochastic objective function is adopted from Tsai (2013), and it represents a response surface where most candidate solutions differ widely in terms of performance. The considered stochastic constraint is a quadratic and convex function, and is formulated as follows:

$$Y(S_i) = \left(\sum_{j=1}^{12} S_{ij} - 12 \right)^2 + \left(\sum_{j=1}^{12} S_{ij} \right)^2 \leq Q.$$

To examine the impact of different percentages of near-boundary solutions (i.e., acceptable solutions as defined in Sect. 2) present in the solution space, we consider two settings of the constraint threshold: $Q = 80$ and $Q = 96$. For the case of $Q = 80$, we have 3498 feasible solutions and 990 near-boundary solutions (accounting for 24% of the solution space), which is considered to be a scenario with a high level percentage of near-boundary solutions. In this case, the optimal feasible solution is $(0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1)$ with an objective value of 140. For the case of $Q = 96$, we have 3938 feasible solutions and there are not any near-boundary solutions, which is considered as a scenario with a low level percentage of near-boundary solutions. In this case, the optimal feasible solution is $(0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1)$ with an objective value of 157. We set the noise variance for both the objective and constraint performance function as $\sigma^2 = 150$. When performing the proposed rapid screening algorithms, we set the indifference-zone parameter (for the objective performance) as $\delta = 5$, and specify the tolerance level (for the constraint performance) as $\epsilon = 5$.

Each algorithm is performed on different experimental configurations, and examining factors include the number of screening iterations (R), the growth factor of the number of replications (G), the number of initial solutions ($b(0)$), and the number of candidate solutions encountered in each iteration (M). We specify a fixed value of M for each iteration, which implies that the number of newly generated solutions ($b(r)$) can be computed by the difference between the value of M and the number of remaining solutions. After some pilot experiments, we find that $M = 40$ is a reasonable choice when implementing the proposed rapid screening algorithms. It should be noted that Tsai (2013) proposed and discussed different methods to generate candidate solutions from the neighborhood of surviving solutions, but in the current experimental study, we only adopt the nearest-neighborhood search method. The initial sample size for feasibility check and objective screening is set as $n_0 = 20$ and $N_0 = 12$,

Table 2 Parameter settings in the experimental study

Parameters	Tested values
Number of screening iterations (R)	6
Sample size growth factor (G)	1.3
Initial number of solutions ($b(0)$)	8
Number of solutions evaluated in each iteration (M)	40
Initial sample size for feasibility check (n_0)	20
Initial sample size for objective screening (N_0)	12
Overall allowable error (α)	0.05
Allowable errors for $\mathcal{RSC}\text{-}A$	$\alpha_1 = \alpha_2 = \alpha/2$ and $\alpha_s = \alpha_f = 1 - \sqrt{1 - \alpha_1}$
Allowable errors for $\mathcal{RSC}\text{-}B1$	$\alpha_c = \alpha_2 = \alpha/2$ and $\alpha_s = \alpha_f = 1 - \sqrt{1 - \alpha_2}$
Allowable errors for $\mathcal{RSC}\text{-}B2$	$\alpha_c = \alpha/(4R)$ and $\alpha_f = \alpha' = \alpha_2 = \alpha/4$
Allowable errors for $\mathcal{RSC}\text{-}C$	$\alpha_s = \alpha_f = \alpha_2 = \alpha/3$
Optimal values for the knapsack problem	612 ($Q = 260$), 836 ($Q = 370$), 1014 ($Q = 460$)
Optimal values for the quadratic problem	140 ($Q = 80$), 157 ($Q = 96$)

respectively. We also specify $R = 6$, $G = 1.3$, and $b(0) = 8$ across different versions of the rapid screening algorithms. In all of the experiments, the overall confidence level is set as $1 - \alpha = 0.95$. For the $\mathcal{RSC}\text{-}A$ algorithm, we divide the allowable error α equally, set $\alpha_1 = \alpha_2 = \alpha/2$, and then specify $\alpha_s = \alpha_f = 1 - \sqrt{1 - \alpha_1}$. An upper bound of k (which is used to compute η in the Feasibility Check stage) is set as $k = 8 + 39 \times 5 = 235$. For the $\mathcal{RSC}\text{-}B1$ algorithm, we set $\alpha_c = \alpha_2 = \alpha/2$, and then specify $\alpha_s = \alpha_f = 1 - \sqrt{1 - \alpha_2}$. For the $\mathcal{RSC}\text{-}B2$ algorithm, we have to satisfy $R \times \alpha_c + \alpha_f = \alpha' + \alpha_2 = \alpha/2$, and each error amount is allocated equally. For the $\mathcal{RSC}\text{-}C$ algorithm, we specify $\alpha_s = \alpha_f = \alpha_2 = \alpha/3$. The AK algorithm is applied to the entire set of candidate solutions, and the allowable error for both the feasibility check stage and the comparison stage is equal to $\alpha/2$. For the OCBA algorithm, the sampling budget is chosen in accordance with the experimental results of the $\mathcal{RSC}\text{-}A$ algorithm. In each trial of the OCBA algorithm, we randomly choose a fixed number of candidate solutions within the solution space, and consider them to be the set of available solutions. The initial replications for each solution is set as $n_0 = 10$, and incremental 100 replications are allocated in each iteration until the specified sampling budget is exhausted. The detailed parameter settings for the empirical study are provided in Table 2.

For each problem configuration, 500 macro-replications (complete repetitions) of each competing algorithm are performed. To compare the performance of the algorithms we record the Average Performance (i.e., expected value of the objective function) of the final selected solution (AP), the Average Number of Samples (ANS) required by each algorithm (without dividing by the number of visited solutions), the Average number of total Generated Solutions (AGS), the Average number of Surviving Solutions through R screening iterations (ASS), the Probability of selecting Feasible Solutions (PFS), and the estimated PCS. To simplify the experimental presentation, we round the values of AP, ANS and AGS to the nearest whole number, round the values of ASS to the nearest hundredth, and round the values of PFS and PCS to the nearest thousandth.

4.2 Summary of results

The estimated PFS and PCS for the proposed rapid screening algorithms and the AK algorithm are higher than the nominal level 0.95 in most investigated configurations. The only excep-

tion is that $\mathcal{RSC-B1}$ shows an inferior PCS performance in a specially designed scenario. Algorithm \mathcal{A} is intuitive and statistically valid, but it also consumes the most simulation replications among the proposed algorithms. By contrast, the advantage of Algorithm \mathcal{B} (compared to other algorithms) is more significant in terms of the required sampling effort especially when we have sampled many solutions close to the constraint boundary. Algorithm \mathcal{C} can be proven to achieve the statistical guarantee and consumes an affordable amount of simulation effort, but it might return an inferior solution because the number of sampled solutions is not large enough. The other advantage of Algorithm \mathcal{C} is that its parameters required to be specified is fewer than the other two algorithms. In almost every scenario, the AK algorithm can identify a final solution with the best quality, as anticipated, but it also requires a substantially larger amount of simulation effort. It should also be noted that in the empirical study, we implement pure OCBA without adopting any efficient solution searching mechanism. Therefore it is not surprising that OCBA delivers inferior performance in terms of AP, PFS and PCS. We do not intend to suggest that pure OCBA is not a good choice for ordinary R&S problems. Please refer to [Branke et al. \(2007\)](#) for a comprehensive comparison between frequentist-type and Bayesian-type procedures.

4.3 Experimental results for the constrained stochastic knapsack problem

The experimental results for the stochastic knapsack problem under the low and high variance scenarios (with different settings of Q and $R = 6$) are summarized in Table 3. When the output variance increases, the compared algorithms all deliver increasing values of ANS and ASS, while their AP and AGS values are decreased. This is an anticipated result since more variance naturally leads to a more difficult feasibility check and solution elimination process. In addition, we can see that among the proposed rapid screening algorithms, $\mathcal{RSC-A}$ presents the best AP performance but also consumes the most simulation replications. The excessive simulation effort required by $\mathcal{RSC-A}$ is not surprising because it has to implement a feasibility check to every newly sampled solution, and the collected simulated observations cannot be exploited in the subsequent Objective Screening stage. The difference in ANS from $\mathcal{RSC-A}$ and other rapid screening algorithms is more significant especially when the output variability is high. For instance, in the case with $Q = 460$ and “High Variance”, the required ANS of $\mathcal{RSC-A}$ is more than two times of the sampling cost of $\mathcal{RSC-B1}$. We find that $\mathcal{RSC-B1}$ uses the smallest ANS and still maintains competitive performance in terms of AP. The estimated PFS and PCS of $\mathcal{RSC-B1}$ and $\mathcal{RSC-B2}$ are higher than the nominal level of 0.95 in the tested configurations, although their statistical validity cannot be provided. It should be noted that $\mathcal{RSC-B2}$ delivers somewhat stronger statistical validity (as compared to $\mathcal{RSC-B1}$) in the sense that it can at least guarantee that the best solution is retained in the subset before performing the Selection step (see Theorem 4). We can also see that $\mathcal{RSC-B2}$ requires more sampling effort than $\mathcal{RSC-B1}$ because it encounters many more candidate solutions in the final stage (i.e., more ASS). For the same reason, $\mathcal{RSC-B2}$ can usually achieve better final selected solution quality in terms of AP (as compared to $\mathcal{RSC-B1}$). It should be noted that in the Objective Screening stages, $\mathcal{RSC-C}$ can only screen out candidate solutions when they are statistically inferior to any *feasible* solution, which implies a more strict condition to perform elimination when compared to other algorithms. Given a setting where a fixed number of solutions (M) is evaluated in each iteration, we can therefore find that $\mathcal{RSC-C}$ visits fewer candidate solutions (i.e., smaller AGS and ASS) in the entire process, which then leads to somewhat inferior AP performance (especially in the cases with “High Variance”). In most cases, $\mathcal{RSC-C}$ requires more sampling cost than $\mathcal{RSC-B1}$ mainly because it allocates a smaller allowable error for selection of the best (so more simulation effort is needed in

the Selection stage). We also implement the experiments of $\mathcal{RSC-A}$ using different values of R (8, 10, and 12) for the constrained stochastic knapsack problem (see Table 4). We use $\mathcal{RSC-A}$ as an illustration because it requires the most sampling effort among the proposed algorithms. We can see that the global optimal solution can be obtained when $R = 10$ or 12, and the required sampling cost is still far less than that of AK.

4.4 Special case for the constrained stochastic knapsack problem

The tested configuration is chosen to make the problem especially difficult for $\mathcal{RSC-B1}$ by placing a competitive solution (in terms of the objective performance) near the boundary of the feasible region. The profit per weight unit p_j is specified as (2, 1, 10, 11, 9, 3, 1, 4, 2, 3), for $j = 1, 2, \dots, 10$. The constraint threshold is specified as $Q = 252$ with a tolerance level $\epsilon = 5$. The other parameter settings are the same as those mentioned in Sect. 4.1. In this case we have 185 feasible solutions, which accounts for only 18% of the solution space, and it represents a scenario where only a small portion among the candidate solutions is feasible. The optimal feasible solution is (0, 0, 1, 1, 0, 0, 0, 1, 0, 0) with an objective value of 2071. The solution (0, 0, 1, 1, 1, 0, 0, 0, 0, 0) is an unacceptable solution (denoted as \mathbf{S}_u) with an objective value of 2564 and where the expected constraint performance is 258 (note that $Q + \epsilon = 257$), which is designated as one of the initial solutions. In this case, it is likely that some competitive feasible solutions (which are sampled in subsequent iterations) are incorrectly eliminated by this specific chosen unacceptable solution \mathbf{S}_u . The experimental results under both low and high variance scenarios are summarized in Table 5. For the comparative results of the “Low Variance” scenario, we can observe similar patterns as seen in Table 3. We also find that $\mathcal{RSC-B1}$ consumes the least amount of sampling cost while still identifying a promising final solution. However, in the “High Variance” scenario, $\mathcal{RSC-B1}$ delivers the worst performance in terms of AP and PCS. This is because the higher output variability makes it more difficult to remove \mathbf{S}_u in the Constraint Screening stage, which increases the chance of promising feasible solutions (or the best solution) being eliminated by \mathbf{S}_u in the subsequent Objective Screening stages.

4.5 Experimental results for the constrained binary quadratic problem

The experimental results for the stochastically constrained binary quadratic problem with different percentage levels of near-boundary solutions are summarized in Table 6. The disadvantage of $\mathcal{RSC-A}$ in terms of sampling cost is more significant when encountering a higher percentage of near-boundary solutions in the solution space (more than three times the sample size compared to $\mathcal{RSC-B1}$). This is because $\mathcal{RSC-A}$ has to implement a feasibility check for every possible near-boundary solution, in which case excessive sampling effort might be incurred. In this case, $\mathcal{RSC-B2}$ also requires more sampling cost because more solutions remain in the subset before the feasibility check process. By contrast, the performance of $\mathcal{RSC-B1}$ is not so sensitive to different levels of near-boundary solutions because it uses the accumulated observations to perform a final feasibility check. On the other hand, when there is a higher percentage of near-boundary solutions, we can see that $\mathcal{RSC-C}$ suffers from visiting a much smaller number of candidate solutions and therefore has worse AP performance (compared to the optimal solutions in these two settings). Similar to the previous results, we find that $\mathcal{RSC-C}$ requires significantly less sampling cost than either $\mathcal{RSC-A}$ or $\mathcal{RSC-B2}$.

Table 3 Experimental results for the six algorithms when applied to the constrained stochastic knapsack problem

	Low variance						High variance					
	A	B1	B2	C	AK	OCBA	A	B1	B2	C	AK	OCBA
<i>Q</i> = 260												
AP	602	597	600	597	612	463	595	590	592	586	612	460
ANS	6387	5858	6121	5891	25,538	6600	7900	5943	6945	6143	29,735	8000
AGS	195	195	197	192	1024	200	187	188	191	173	1024	200
ASS	1.83	1.99	61	1.91	N/A	N/A	4.02	3.98	72	3.48	N/A	N/A
PFS	1	1	1	1	1	0.015	1	1	1	1	1	0.005
PCS	1	0.99	0.99	1	1	0.015	1	0.95	0.98	1	1	0.005
<i>Q</i> = 370												
AP	834	828	830	828	836	679	833	826	828	827	836	701
ANS	8742	5910	7618	6703	36,158	9000	13,279	6619	8513	8183	49,580	14,000
AGS	188	188	192	182	1024	200	183	183	184	155	1024	200
ASS	5.28	5.32	85	5.24	N/A	N/A	7.35	7.13	103	5.74	N/A	N/A
PFS	1	1	1	1	1	0.04	1	1	1	1	1	0.125
PCS	1	1	1	1	1	0.04	1	0.99	0.99	1	1	0.125
<i>Q</i> = 460												
AP	1010	1004	1006	1001	1014	767	1008	1000	1004	998	1014	754
ANS	9571	5903	7804	6380	40,297	10,000	13,973	6635	9017	8181	52,699	14,000
AGS	188	187	189	182	1024	200	180	180	182	153	1024	200
ASS	4.88	4.92	108	4.62	N/A	N/A	6.95	6.93	135	4.88	N/A	N/A
PFS	1	1	1	1	1	0.70	1	1	1	1	1	0.75
PCS	1	0.99	0.99	1	1	0.70	1	0.96	0.98	1	1	0.75

We specifically record the average performance (AP), the average number of samples (ANS), the average number of total generated solutions (AGS), the average number of surviving solutions (ASS), the probability of selecting feasible solutions (PFS), and the probability of correct selection (PCS)

Table 4 Experimental results for $\mathcal{R}SC\text{-}\mathcal{A}$ with different values of R (specified in the parenthesis) when applied to the constrained stochastic knapsack problem under the low variance scenario

	$Q = 260$					$Q = 460$				
	$\mathcal{A}(6)$	$\mathcal{A}(8)$	$\mathcal{A}(10)$	$\mathcal{A}(12)$	AK	$\mathcal{A}(6)$	$\mathcal{A}(8)$	$\mathcal{A}(10)$	$\mathcal{A}(12)$	AK
AP	602	608	612	612	612	1010	1012	1013	1014	1014
ANS	6387	8072	10,167	13,841	25,538	9571	12,124	16,414	21,567	40,297
AGS	195	272	348	426	1024	188	257	330	404	1024
ASS	1.83	1.20	1.03	1.00	N/A	4.88	3.73	2.62	1.92	N/A
PFS	1	1	1	1	1	1	1	1	1	1
PCS	1	1	1	1	1	1	1	1	1	1

Table 5 Experimental results for the rapid screening algorithms when applied to a designed configuration of the constrained stochastic knapsack problem

	Low variance				High variance			
	\mathcal{A}	$B1$	$B2$	C	\mathcal{A}	$B1$	$B2$	C
AP	2062	2065	2068	2059	2031	1785	2030	2012
ANS	7118	5856	6872	6016	15,745	6952	7954	7218
AGS	200	202	205	191	191	190	189	142
ASS	1.64	1.42	51	1.53	3.86	3.77	45	3.31
PFS	1	1	1	1	1	1	1	1
PCS	1	1	1	1	1	0.36	1	1

Table 6 Experimental results for the rapid screening algorithms when applied to the stochastically constrained binary quadratic problem under different percentages of near-boundary solutions

	Low percentage				High percentage			
	\mathcal{A}	$B1$	$B2$	C	\mathcal{A}	$B1$	$B2$	C
AP	149	142	145	139	130	122	126	116
ANS	34,549	18,905	24,873	21,005	71,376	22,975	34,976	28,747
AGS	166	145	154	131	155	135	142	114
ASS	3.45	5.67	46	5.63	3.14	9.97	54	3.06
PFS	1	1	1	1	1	0.99	1	1
PCS	1	0.99	0.99	1	1	0.99	0.99	1

5 Conclusions

In this paper, we develop three rapid screening algorithms to solve the zero-one simulation optimization problem considering a single stochastic constraint. Algorithm \mathcal{A} is more intuitive and is easier to implement (and statistically valid), but it may require excessive sampling effort especially when there are lots of near-boundary solutions. Algorithm $B\text{-}1$ requires the fewest simulation replications, but it is a heuristic (i.e., cannot deliver the PCS guarantee) and is especially undesirable when there are some unacceptable solutions (which are close to the feasible region) with a superior objective performance, which are also sampled in early iterations. We also propose Algorithm $B\text{-}2$ to improve the statistical validity, but this would be at the price of generating an additional number of simulation replications. Algorithm C can

maintain the desired statistical guarantee, and in the meantime, the sampling cost is modest, but it might return a final solution with inferior objective performance. Algorithm *C* is more favorable when there exists some very promising feasible solution where, in the meantime, the determination of its feasibility is not very difficult. In this case, it is better that at least one of the good solutions is identified in early iterations.

There are several possible interesting extensions for future research. For instance, we could design new procedures to exploit variance reduction techniques, such as control variates and CRN, to further improve the statistical efficiency (see Tsai and Kuo 2012). Another possible extension would be to consider the zero-one simulation optimization problem with multiple stochastic constraints (see Batur and Kim 2010). The other possible research topic would be to develop efficient algorithms that can take advantage of Bayesian-type procedures (e.g., OCBA) in the rapid-screening framework for constrained simulation optimization.

References

- Amaran, S., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2016). Simulation optimization: A review of algorithms and applications. *Annals of Operations Research*, *240*, 351–380.
- Andradóttir, S., & Kim, S. H. (2010). Fully sequential procedures for comparing constrained systems via simulation. *Naval Research Logistics*, *57*, 403–421.
- Atlason, J., Epelman, M. A., & Henderson, S. G. (2004). Call center staffing with simulation and cutting plane methods. *Annals of Operations Research*, *127*, 333–358.
- Barahona, F., Jünger, M., & Reinelt, G. (1989). Experiments in quadratic 0–1 programming. *Mathematical Programming*, *44*, 127–137.
- Batur, D., & Kim, S. H. (2010). Finding feasible systems in the presence of constraints on multiple performance measures. *ACM Transactions on Modeling and Computer Simulation*, *20*, 1–26.
- Boesel, J., Nelson, B. L., & Kim, S. H. (2003). Using ranking and selection to “clean up” after simulation optimization. *Operations Research*, *51*, 814–825.
- Branke, J., Chick, S. E., & Schmidt, C. (2007). Selecting a selection procedure. *Management Science*, *53*, 1916–1932.
- Chen, E. J. (2011). A revisit of two-stage selection procedures. *European Journal of Operational Research*, *210*, 281–286.
- Ho, Y. C., Cassandras, C. G., Chen, C. H., & Dai, L. (2000). Ordinal optimisation and simulation. *Journal of the Operational Research Society*, *51*, 490–500.
- Kim, S. H., & Nelson, B. L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, *11*, 251–273.
- Kosuch, S., & Lisser, A. (2010). Upper bounds for the 0–1 stochastic knapsack problem and a branch and bound algorithm. *Annals of Operations Research*, *176*, 77–93.
- Lee, L. H., Pujowidianto, N. A., Li, L. W., Chen, C. H., & Yap, C. M. (2012). Approximate simulation budget allocation for selecting the best design in the presence of stochastic constraints. *IEEE Transactions on Automatic Control*, *57*, 2940–2945.
- Nelson, B. L., & Goldsman, D. (2001). Comparison with a standard in simulation experiments. *Management Science*, *47*, 449–463.
- 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*, 950–963.
- Park, C., & Kim, S. H. (2015). Penalty function with memory for discrete optimization via simulation with stochastic constraints. *Operations Research*, *63*, 1195–1212.
- Pichitlamken, J., Nelson, B. L., & Hong, L. J. (2006). A sequential procedure for neighborhood selection-of-the-best in optimization via simulation. *European Journal of Operational Research*, *173*, 283–298.
- Rinott, Y. (1978). On two-stage selection procedures and related probability-inequalities. *Communications in Statistics—Theory and Methods*, *A7*, 799–811.
- Tsai, S. C. (2013). Rapid screening procedures for zero-one optimization via simulation. *Inform Journal on Computing*, *25*, 317–331.
- Tsai, S. C., & Chu, I. H. (2012). Controlled multistage selection procedures for comparison with a standard. *European Journal of Operational Research*, *223*, 709–721.

- Tsai, S. C., & Kuo, C. H. (2012). Screening and selection procedures with control variates and correlation induction techniques. *Naval Research Logistics*, *59*, 340–361.
- Udías, A., Efremov, R., Galbiati, L., & Cañamón, I. (2014). Simulation and multicriteria optimization modeling approach for regional water restoration management. *Annals of Operations Research*, *219*, 123–140.
- Williams, P., Guangfu, T., & Yiming, L. (2010). Simulation based analysis of patient arrival to health care systems and evaluation of an operations improvement scheme. *Annals of Operations Research*, *178*, 263–279.