2021

Novel Approaches to Feasibility Determination

Solow, Daniel; Szechtman, Roberto; Yücesan, Enver

ACM

Solow, Daniel, Roberto Szechtman, and Enver Yücesan. "Novel Approaches to Feasibility Determination." ACM Transactions on Modeling and Computer Simulation (TOMACS) 31.1 (2021): 1-25.
http://hdl.handle.net/10945/67395

Novel Approaches to Feasibility Determination

DANIEL SOLOW, Case Western Reserve University, USA
ROBERTO SZECHTMAN, Naval Postgraduate School, USA
ENVER YÜCESAN, INSEAD, Singapore

This article proposes two-stage Bayesian and frequentist procedures for determining whether a number of systems—each characterized by the same number of performance measures—belongs to a set Γ defined by a finite collection of linear inequalities. A system is "in (not in) Γ" if the vector of the means is in (not in) Γ, where the means must be estimated using Monte Carlo simulation. We develop algorithms for classifying the systems with a user-specified level of confidence using the minimum number of simulation replications so the probability of correct classification over all r systems satisfies a user-specified minimum value. Once the analyst provides prior values for the means and standard deviations of the random variables in each system, an initial number of simulation replications is performed to obtain current estimates of the means and standard deviations to assess whether the systems can be classified with the desired level of confidence. For any system that cannot be classified, heuristics are proposed to determine the number of additional simulation replications that would enable correct classification. Our contributions include the introduction of intuitive algorithms that are not only easy to implement, but also effective with their performance. Compared to other feasibility determination approaches, they also appear to be competitive. While the algorithms were initially developed in settings where system variance is assumed to be known and the random variables are independent, their performance remains satisfactory when those assumptions are relaxed.

CCS Concepts: • **Computing methodologies** → *Modeling and simulation*; • **Mathematics of computing** → *Probability and statistics*; • **Theory of computation** → *Design and analysis of algorithms*; • **Applied computing** → *Physical sciences and engineering*;

Additional Key Words and Phrases: Multiple performance measures, ranking and selection, stochastic constraints, classifying system

ACM Reference format:
Daniel Solow, Roberto Szechtman, and Enver Yücesan. 2020. Novel Approaches to Feasibility Determination. *ACM Trans. Model. Comput. Simul.* 31, 1, Article 1 (December 2020), 25 pages.
https://doi.org/10.1145/3426359

1 INTRODUCTION, BACKGROUND, AND NOTATION

We introduce two-stage Bayesian and frequentist algorithms for feasibility determination in a stochastic setting, where the objective is to determine whether each system in a finite, but potentially large, collection belongs to a given set $\Gamma \subset R^m$ based on a number of performance measures estimated through Monte Carlo simulation.

Authors' addresses: D. Solow, Case Western Reserve University, Cleveland, OH, 44106; email: daniel.solow@case.edu; R. Szechtman, Naval Postgraduate School, Glasgow Hall 254, Monterey, CA, 93943; email: rszechtm@nps.edu; E. Yücesan, INSEAD, 1, Ayer Rajah Avenue, 138676, Singapore; email: enver.yucesan@insead.edu.

1049-3301/2020/12-ART1 $15.00
https://doi.org/10.1145/3426359

While our discussion will be anchored in computer simulation, feasibility determination has broader applications. We will illustrate this broader relevance through a number of examples from various industries. An area where this challenge is particularly relevant is open innovation through "crowdsourcing," a practice facilitated by web-based platforms where monetary rewards are offered by organizations seeking help in addressing succinctly defined problems such as debottlenecking a chemical process to increase its throughput. An open innovation platform such as *Innocentive* or *Hypios* receives thousands of proposed solutions whose feasibility must be verified through some kind of noisy experiment before the best solution could be selected and the award could be attributed to the winning solution.

Another interesting application of the feasibility determination problem is in agricultural research and development. The identification of robust seeds (say, against diseases or extreme weather conditions) necessitates the "crossing" of a large number of varieties until the desired traits can be obtained [11]. This combinatorial problem not only requires tens of millions of dollars, but also necessitates long development cycles of 9 to 12 years. A recent approach, *in silico* breeding, relies on computer simulation modeling to enable the identification of a feasible set of hybrid seeds (for example, whose germination rate exceeds 95%) among billions of genotypes before expensive field trials are undertaken with varieties in that set.

Other applications include the determination of a set of investment portfolios whose expected payoff exceeds a desired threshold with a certain probability. Finally, in multi-market entry, "seeding" strategies are deployed for new product introductions [14]. The allocation of limited resources (e.g., an advertising budget) among multiple markets to achieve a certain level of market penetration turns out to be an important feasibility determination problem in the field of Marketing.

Feasibility determination has recently attracted much attention within the context of ranking and selection (R&S) in the presence of stochastic constraints. Traditionally, the overwhelming majority of the R&S research has focused on a single unconstrained performance measure: Given a finite set of competing design alternatives, R&S has been concerned with the efficient use of a limited computational budget to identify the design with the best performance, where performance is estimated through expensive Monte Carlo simulation experiments.

R&S procedures aim at allocating the experimental budget in an efficient way among alternative designs that appear to be good and designs whose performance remains highly uncertain. Competing systems therefore receive additional simulation replications based on their potential of contributing useful information to the R&S process. Typically using Bayesian statistical models, the expected improvement (EI) approach allocates the simulation replications one at a time in an adaptive manner based on a probabilistic forecast that quantifies the potential of the additional replication to improve the current estimate of the performance measure (Chick and Inoue [5], Chick et al. [6], Frazier et al. [8], and Xie and Frazier [24]). Optimal computing budget allocation, or OCBA [4], is a Bayesian heuristic technique that identifies inferior systems rapidly to eliminate them from further experimentation and allocate the remaining computing budget to those designs that appear to be good. Ryzhov [15] has recently established the asymptotic equivalence of the OCBA and EI methodologies in the sense that they asymptotically produce virtually identical simulation budget allocations in the absence of any feasibility constraints. Frequentist approaches, however, offer a statistical guarantee whereby, when the (unknown) performance of the best system differs by more than an indifference parameter specified by the analyst, the probability of selecting the best system is at least $1 - \alpha$.

Under the assumption of normality, Lee et al. [13] have recently extended the OCBA approach to the selection of the best design under stochastic constraints. To the best of our knowledge, Santner and Tamhane [16] is the first paper to propose a two-stage procedure that incorporates a constraint on variance. In a setting where R&S is based on a primary performance measure subject

to the feasibility of a (possibly correlated) secondary performance measure, Andradottir et al. [1] and Andradottir and Kim [2] propose a two-phase approach whereby phase I identifies feasible systems while phase II selects the best among them. With the objective of accelerating the first phase, Batur and Kim [3] have introduced efficient procedures for finding a set of feasible or near-feasible systems in the presence of orthant constraints. We will benchmark our algorithms with theirs in our numerical evaluations.

To address the problem of feasibility determination, Szechtman and Yücesan [19] use large-deviations theory whereby they estimate the large deviations rate functions "on the fly." Unfortunately, unless the probability distribution associated with the performance measure of interest has finite support, it may not be possible to control the estimation error—hence, making it impossible to achieve the desired probability of correct selection. However, Gao and Chen [9] show that such heuristics work well in practice. Hunter and Pasupathy [12] avoid this problem by assuming that the underlying distributional family of the simulation estimator is known or assumed; as a result, the form of the large deviations rate function is known.

Our contributions to this growing literature include the introduction of intuitive algorithms that are not only easy to implement, but are also effective and seem to be competitive compared with other feasibility determination approaches.

Throughout this article, the following notation is used (in which all vectors are column vectors):

r	=	the number of systems to be classified.
m	=	the number of performance measures associated with each system.
c	=	the number of linear constraints defining the feasibility set Γ.
i	=	a subscript that denotes system i, taking values from 1 to r.
j	=	a subscript that denotes the jth component of a system, ranging from 1 to m.
k	=	a subscript that denotes the kth constraint of the set Γ, ranging from 1 to c.
A^T	=	the transpose of the matrix A.
$\boldsymbol{a}^2$	=	$(a_1^2, \ldots, a_m^2)^T$, where $\boldsymbol{a}^T = (a_1, \ldots, a_m)$.
Φ	=	the cumulative distribution function of the standard normal random variable $Z \sim N(0,1)$.

The remainder of the article is organized as follows: The next section formally defines the problem and outlines the proposed classification approach. In Section 3, conditions are identified under which the system can be readily classified. If the system cannot be readily classified, a Bayesian and a non-Bayesian approach to determine the number of additional replications required for classification are introduced in Section 4 and Section 5, respectively. The complete algorithm is depicted in Section 6, while numerical experiments are presented in Section 7. Section 8 provides the concluding remarks and highlights venues for future research.

2 OVERVIEW OF THE PROBLEM AND THE PROPOSED CLASSIFICATION APPROACH

This section introduces the intuition behind our algorithms and discusses the assumptions we have made in constructing the algorithms. Recall that we consider a collection of r systems, each with unknown performance measure $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_r \in R^m$, where $\boldsymbol{\mu}_i = E[\mathbf{X}_i]$ for some random vector $\mathbf{X}_i \in R^m$. The objective is to determine whether each $\boldsymbol{\mu}_i$ is (or is not) in a set Γ, called the **feasible region**, defined by a finite collection of linear inequalities. Specifically, given a $(c \times m)$ matrix A, with rows $\boldsymbol{a}_\ell^T$, $\ell = 1, \ldots, c$, and a vector $\boldsymbol{b} \in R^c$, the feasible region is defined as

$$\Gamma = \{\mathbf{x} \in R^m : A\mathbf{x} \leq \boldsymbol{b}\}.$$

Without loss of generality, we assume that Γ has a non-empty interior and that no constraint is redundant, meaning that $\{\mathbf{x} \in R^m : \mathbf{a}_\ell^T \mathbf{x} \leq b_\ell, \forall \ell \neq k\} \subset \Gamma$ for any $k \in \{1, \ldots, c\}$, as it is always possible to find (and remove) redundant constraints by solving a linear program (see Reference [21] for more details).

System i is said to be *in* Γ if $\boldsymbol{\mu}_i \in \Gamma$ and *not in* Γ if $\boldsymbol{\mu}_i \notin \Gamma$. However, because $\boldsymbol{\mu}_i$ is unknown, in Stage 1 of the proposed algorithms, simulation is performed to obtain, say, n_i, initial i.i.d. replicates $\mathbf{X}_i(1), \ldots, \mathbf{X}_i(n_i)$ that are used to form an *estimate*, say, $\boldsymbol{\mu}_i(n_i)$, of $\boldsymbol{\mu}_i$. Based on $\boldsymbol{\mu}_i(n_i)$, one can choose either to classify the system as *in* Γ or *not in* Γ, or else to perform additional simulation replications. It is important to note that any classified system might be *incorrectly classified*. This happens when system i is classified as in Γ when in fact $\boldsymbol{\mu}_i \notin \Gamma$ or when system i is classified as not in Γ when in fact $\boldsymbol{\mu}_i \in \Gamma$. When a system is classified as in Γ or not in Γ, one would like some degree of assurance that such a system is correctly classified. To that end, the analyst is asked to choose a desired level of confidence in the form of a fraction, $1 - \beta$, for which the algorithms developed here guarantee that any classified system is correctly classified with probability $1 - \beta$.

For any system i that is not classified in Stage 1, the goal of Stage 2 of the proposed algorithm is to determine the number, Δn_i, of *additional* simulations to perform, resulting in a new estimate, $\boldsymbol{\mu}_i(n_i + \Delta n_i)$, of $\boldsymbol{\mu}_i$. While one could, in theory, continue this process until all systems are classified with probability $1 - \beta$, doing so could be prohibitive in terms of the total number of computer simulations needed. Furthermore, computational experiments reported in Section 7 showed that more than 90% of systems were classified at the end of Stage 2; therefore, this process is often repeated only once. A variety of Bayesian and Non-Bayesian heuristics are proposed for determining the number of additional replications to perform in Stage 2, each with the goal of being able to classify the system with probability $1 - \beta$, although there is no guarantee of doing so.

To implement these algorithms, the following three assumptions are made in the subsequent development:

Assumption 1 (Known Variances): The true variances, $\boldsymbol{\sigma}_i^2$, of the m random variables of each system i are known.

This assumption significantly simplifies the development of the subsequent algorithms. In practice, however, these variances are not known. In Section 7, modifications to the algorithms that allow for unknown variances are proposed and computational results are presented.

Assumption 2 (Independence): The m random variables in each system i are pairwise independent of each other, so the covariance matrix Σ_i is a diagonal matrix with diagonal elements $\boldsymbol{\sigma}_i^2 = (\sigma_{i1}^2, \ldots, \sigma_{im}^2)$.

This assumption is made without loss of generality, because when the m random variables in each system are not pairwise-independent of each other, it is possible to work in a transformed $\boldsymbol{y}$ coordinate system defined by the eignevectors, say, $\mathbf{v}^1, \ldots, \mathbf{v}^m$, of the true covariance matrix Σ (assumed to be known), as shown by the dotted axes in Figure 1. This is accomplished by using the $(m \times m)$ matrix V whose columns are the eigenvectors of Σ, to convert any vector $\boldsymbol{x}$ in the original coordinate system to the following vector in the $\boldsymbol{y}$ coordinate system:

$$\boldsymbol{y} = V^{-1}\boldsymbol{x}.$$

In particular, the linear inequalities that define Γ become the following in the $\boldsymbol{y}$ coordinate system:

$$AV\boldsymbol{y} \leq \boldsymbol{b}.$$

The proposed algorithms are then applied in the $\boldsymbol{y}$ coordinate system.

Assumption 3 (Classifiability): No system i "lies" on the boundary of the feasible region, that is, $\boldsymbol{\mu}_i \notin \partial\Gamma$.

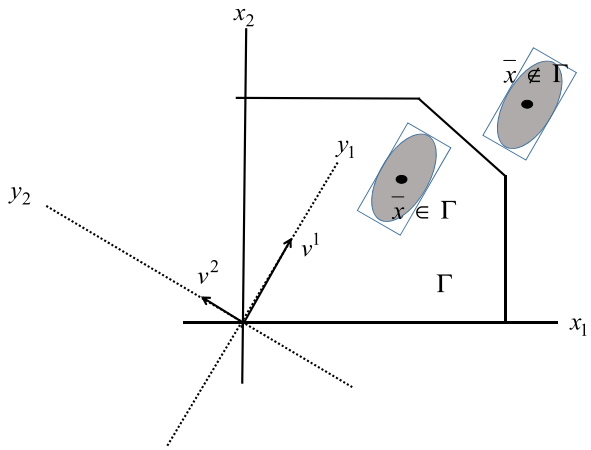

Fig. 1. A transformed coordinate system when the random variables in a system are correlated.

This assumption means that for a sequence of posterior means $\boldsymbol{\mu}_i(n)$ converging to $\boldsymbol{\mu}_i$, for large enough n, all subsequent posterior means in the sequence will either be inside Γ or outside Γ. This enables one to have some degree of confidence in correct classification when the system is classified based on the posterior mean.

The proposed algorithms require prior estimates of $\boldsymbol{\mu}_i$ and Σ_i, say, $\boldsymbol{\mu}_i(0) \in R^m$ and $\Sigma_i(0)$ (an $(m \times m)$ diagonal matrix with its jth diagonal element equal to $\sigma_{ij}^2(0)$). It is assumed that the analyst provides these initial estimates and that the simulation provides i.i.d. samples $\mathbf{X}_i(1), \mathbf{X}_i(2), \ldots,$ from $N(\boldsymbol{\mu}_i, \Sigma_i)$. It is well known [10] that after n samples are drawn from system i,

$$\mu_{ij}|X_{ij}(1), X_{ij}(2), \ldots, X_{ij}(n) \sim N(\mu_{ij}(n), \sigma_{ij}^2(n)), \tag{1}$$

where

$$\mu_{ij}(n) = \frac{\mu_{ij}(0)/\sigma_{ij}^2(0) + n\bar{X}_{ij}(n)/\sigma_{ij}^2}{1/\sigma_{ij}^2(0) + n/\sigma_{ij}^2}, \tag{2}$$

$$\sigma_{ij}^2(n) = \frac{1}{1/\sigma_{ij}^2(0) + n/\sigma_{ij}^2}, \tag{3}$$

$$\bar{X}_{ij}(n) = n^{-1} \sum_k X_{ij}(k). \tag{4}$$

The function $\sigma_{ij}^2(n)$ defined in Equation (3) has the following mathematical properties that are exploited in the subsequent development of the allocation algorithms:

(1) $\sigma_{ij}^2(n)$ monotonically decreases to 0 as $n \to \infty$.
(2) $\sigma_{ij}^2(n)$ and $\sigma_{ij}(n)$ are convex in n on $[0, \infty)$, when n is continuous.

In summary, the two-stage algorithms proposed here use the following general steps for classifying the r systems:

Step 0. Obtain, from the analyst, values $\boldsymbol{\mu}_i(0)$ and $\boldsymbol{\sigma}_i^2(0)$, and the desired value of $1-\beta$. Perform n_0 initial simulation replications for each system i to obtain initial estimates $\boldsymbol{\mu}_i(n_0)$ and $\boldsymbol{\sigma}_i^2(n_0)$ using Equations (2) and (3).

Step 1. Using the current estimates, classify with probability $1-\beta$ as many of the systems as possible, either as in Γ or not in Γ.

Step 2. Determine for which of the remaining systems additional simulation replications will be performed.

Step 3. For each system i identified in Step 2, determine the number of additional simulation replications, Δn_i, to perform. Perform the additional simulation replications for those systems to obtain new estimates of the means and variances in Equations (2) and (3).

Step 4. Classify all of the r systems based on the current estimates of the means and variances.

Note that "correct classification" is defined slightly differently under the Bayesian and frequentist perspectives. In our Bayesian setting, we ensure that the average probability of correct classification over all systems achieves the nominal confidence level. This means that while some systems may be correctly classified at a much higher confidence level than the one specified by the analyst, others may be below the desired confidence level as long as we achieve, on expectation, the confidence level targeted by the analyst. Our frequentist heuristic, however, tries to classify each system with the desired level of confidence. To illustrate this nuance with an example, imagine that we are trying to identify investment opportunities whose payoff exceeds a certain threshold value. In our Bayesian heuristic, the reported portfolio would achieve, on expectation, the nominal probability of correct classification with some investment opportunities clearing the threshold value with a much higher probability while others at a probability of correct classification that is lower than the desired level. The frequentist heuristic, however, would report a portfolio that contains only individual investment opportunities that exceed the threshold value at the desired level of confidence.

In the foregoing algorithm, it is possible to perform Step 1 and Step 3 for each system independently of the other systems—thus allowing for the use of parallel processing. As such, from here on, the subscript i is dropped and the subsequent analysis is performed for one arbitrary system. Thus, for a given system, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_m^2)$ are the true means and variances of that system while $\boldsymbol{\mu}(n) = (\mu_1(n), \ldots, \mu_m(n))$ and $\boldsymbol{\sigma}^2(n) = (\sigma_1^2(n), \ldots, \sigma_m^2(n))$ are the posterior mean and variance in Equations (2) and (3) after obtaining the n simulation replications $\mathbf{X}(1), \ldots, \mathbf{X}(n)$ and $\Sigma(n)$ is the diagonal matrix whose jth diagonal element is $\sigma_j^2(n)$.

After performing n simulation replications to obtain the estimates $\boldsymbol{\mu}(n)$ in Equation (2) and $\boldsymbol{\sigma}^2(n)$ in Equation (3), Step 1 requires determining whether the system can be classified as in Γ or not in Γ with probability $1-\beta$. How this is done is addressed in Section 3. In the event that a system cannot be classified with the desired probability, two different approaches for determining the number of additional simulation replications to perform are presented in Sections 4 and 5, respectively. The complete algorithm is presented in Section 6 and the results of computational experiments are reported in Section 7.

3 CLASSIFYING A SYSTEM

Given the posterior mean, $\boldsymbol{\mu}(n)$, and posterior variance, $\boldsymbol{\sigma}^2(n)$, in Step 1 of the proposed algorithm, it is necessary to determine whether the system can be classified as in Γ or not in Γ with probability $1-\beta$. The way this is done depends on whether $\boldsymbol{\mu}(n) \in \Gamma$ or $\boldsymbol{\mu}(n) \notin \Gamma$, as described now.

3.1 Case 1: $\boldsymbol{\mu}(n) \notin \Gamma$

The system can be classified as not in Γ if $P(\boldsymbol{\mu} \notin \Gamma|\mathbf{X}(1), \ldots, \mathbf{X}(n)) \geq 1 - \beta$, obtained after performing n simulation replications $\mathbf{X}(1), \ldots, \mathbf{X}(n)$. Finding this probability directly is not practical when m is large, because doing so requires finding the area under a multivariate normal density in m dimensions over a half-space defined by a constraint of Γ, which requires significant computational effort (see, for example, Chapter 4 of Traub and Verschultz [22]). A more practical approach is based on the fact that, for *any* hyperplane $H = \{\boldsymbol{x} \in R^m : \boldsymbol{w}^T\boldsymbol{x} = \tilde{w}\}$ separating Γ from $\boldsymbol{\mu}(n)$ with $\Gamma \subset \{\boldsymbol{x} \in R^m : \boldsymbol{w}^T\boldsymbol{x} \leq \tilde{w}\}$ and $\boldsymbol{\mu}(n) \in \{\boldsymbol{x} \in R^m : \boldsymbol{w}^T\boldsymbol{x} > \tilde{w}\}$,

$$P(\boldsymbol{\mu} \notin \Gamma|\mathbf{X(1)}, \ldots, \mathbf{X(n)}) \geq \mathbf{P}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu} > \tilde{\mathbf{w}}|\mathbf{X(1)}, \ldots, \mathbf{X(n)}).$$

Such a hyperplane exists, because Γ, being the intersection of half-spaces, is convex and $\boldsymbol{\mu}(n) \notin \Gamma$. Furthermore, as $\mu_j|\mathbf{X}(1), \ldots, \mathbf{X}(n) \sim N(\mu_j(n), \sigma_j^2(n))$, it follows that the linear combination $\boldsymbol{w}^T\boldsymbol{\mu}|\mathbf{X}(1), \ldots, \mathbf{X}(n) \sim N(\boldsymbol{w}^T\boldsymbol{\mu}(n), \sum_j w_j^2\sigma_j^2(n))$ and so by subtracting this mean and dividing by this standard deviation, the foregoing probability inequality becomes:

$$P(\boldsymbol{\mu} \notin \Gamma|\mathbf{X}(1), \ldots, \mathbf{X}(n)) \geq \Phi(s), \text{ where } s = \frac{\boldsymbol{w}^T\boldsymbol{\mu}(n) - \tilde{w}}{(\sum_j \boldsymbol{w}_j^2\sigma_j^2(n))^{1/2}}. \tag{5}$$

If $\Phi(s) \geq 1 - \beta$, then the system is classified as not in Γ with probability $1 - \beta$.

As any separating hyperplane defined by $(\boldsymbol{w}, \tilde{w})$ might result in classifying the system, it would be desirable to find the *best* such hyperplane, that is, the hyperplane $(\boldsymbol{w}, \tilde{w})$ that maximizes $\Phi(s)$ in Equation (5). Finding this best hyperplane can, in theory, be accomplished by solving the following linearly constrained nonlinear optimization problem in the variables $(\boldsymbol{w}, \tilde{w}, \boldsymbol{v}, \gamma)$:

$$\begin{aligned}
&\max \ \tfrac{\boldsymbol{w}^T\boldsymbol{\mu}(n) - \tilde{w}}{(\sum_j \boldsymbol{w}_j^2\sigma_j^2(n))^{1/2}} \equiv \log(\boldsymbol{w}^T\boldsymbol{\mu}(n) - \tilde{w}) - \tfrac{1}{2}\log[\textstyle\sum_j \boldsymbol{w}_j^2\sigma_j^2(n)] \\
&\text{s.t.} \\
&\qquad \boldsymbol{w}^T = \boldsymbol{v}^T A \\
&\qquad \tilde{w} = \boldsymbol{w}^T\boldsymbol{\mu}(n) - \gamma \\
&\qquad \boldsymbol{w}^T\boldsymbol{\mu}(n) - \boldsymbol{v}^T\boldsymbol{b} = 1 \\
&\qquad \boldsymbol{v} \geq \mathbf{0} \text{ and } 0 \leq \gamma \leq 1.
\end{aligned} \tag{6}$$

Note that any values of $\boldsymbol{w}$ and $\tilde{w}$ that satisfy the constraints of Equation (6) constitute a separating hyperplane for Γ and $\boldsymbol{\mu}(n)$. This is because, for any value of $\boldsymbol{x}$ with $A\boldsymbol{x} \leq \boldsymbol{b}$,

$$\boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{v}^TA\boldsymbol{x} \leq \boldsymbol{v}^T\boldsymbol{b} = \boldsymbol{w}^T\boldsymbol{\mu}(n) - 1 = \tilde{w} + \gamma - 1 \leq \tilde{w}.$$

In contrast,

$$\boldsymbol{w}^T\boldsymbol{\mu}(n) = \tilde{w} + \gamma \geq \tilde{w}.$$

While finding the best separating hyperplane requires solving the nonlinear program in Equation (6), other separating hyperplanes, though not the best one, are readily available. In particular, any constraint of Γ for which $\boldsymbol{\mu}(n)$ is not feasible provides such a separating hyperplane and so can be used to try and classify the system. Specifically, let $L = \{\ell : \boldsymbol{a}_\ell^T\boldsymbol{\mu}(n) > b_\ell\}$ be the set of violated constraints at $\boldsymbol{\mu}(n)$. For any $\ell \in L$, Equation (5) becomes

$$P(\boldsymbol{\mu} \notin \Gamma|\mathbf{X}(1), \ldots, \mathbf{X}(n)) \geq \Phi(s_\ell) \text{ where } s_\ell = \frac{\boldsymbol{a}_\ell^T\boldsymbol{\mu}(n) - b_\ell}{(\sum_j a_{\ell j}^2\sigma_j^2(n))^{1/2}}. \tag{7}$$

Thus, if there exists an $\ell \in L$ for which $\Phi(s_\ell) \geq 1 - \beta$ in Equation (7), then the system is classified as not in Γ with probability $1 - \beta$.

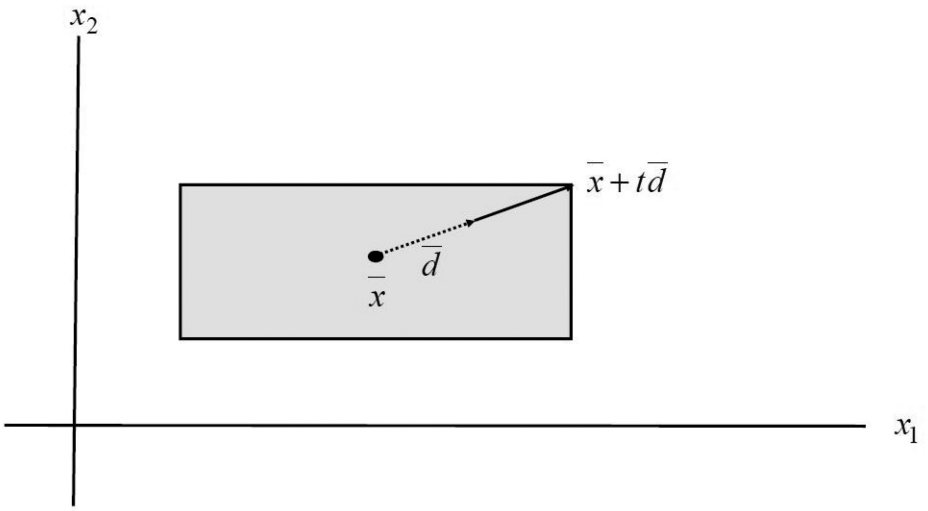

Fig. 2. The rectangle of size t centered at $\bar{\boldsymbol{x}}$ in the direction $\boldsymbol{d}$.

If the system cannot yet be classified as not in Γ with the desired probability using any of the foregoing separating hyperplanes, then it is necessary to determine the number of additional simulation replications, if any, to perform. Two different approaches for doing so are presented in Sections 4 and 5.

3.2 Case 2: $\boldsymbol{\mu}(n) \in \Gamma$

After performing n simulation replications $\mathbf{X}(1), \ldots, \mathbf{X}(n)$, the system can be classified as in Γ if $P(\boldsymbol{\mu} \in \Gamma | \mathbf{X}(1), \ldots, \mathbf{X}(n)) \geq 1 - \beta$. Finding this probability directly is not practical, because doing so requires finding the area under a multivariate normal density in m dimensions over Γ, which requires significant computational effort. Therefore, two alternative approaches for classifying the system are now presented.

3.2.1 Using Confidence Rectangles to Classify a System. The first method for attempting to classify a system is to use a $(1-\beta) * 100\%$ confidence rectangle centered at $\boldsymbol{\mu}(n)$. Given vectors $\bar{\boldsymbol{x}}, \boldsymbol{d} \in R^m$ with $\boldsymbol{d} > \mathbf{0}$, for every real number $t \geq 0$, the m-dimensional rectangle of size t centered at $\bar{\boldsymbol{x}}$ in the direction $\boldsymbol{d}$ is shown in Figure 2 for $m = 2$ and is defined as:

$$R(\bar{\boldsymbol{x}}, \boldsymbol{d}, t) = \{\boldsymbol{x} \in R^m : \bar{\boldsymbol{x}} - t\boldsymbol{d} \leq \boldsymbol{x} \leq \bar{\boldsymbol{x}} + t\boldsymbol{d}\}. \tag{8}$$

Notice that as t increases from 0 to ∞, the rectangle expands from the center $\bar{\boldsymbol{x}}$ to cover all of R^m.

Of particular interest here is a $(1-\beta) * 100\%$ confidence rectangle centered at $\boldsymbol{\mu}(n)$ in the direction $\boldsymbol{\sigma}(n)$ of size z, where z is a number with the property that the rectangle contains $\boldsymbol{\mu}$ with probability $1-\beta$. As $\boldsymbol{\mu}$ follows a normal distribution with parameters given in Equation (1) and the m random variables in the system are independent of each other (by the Independence Assumption in Section 2), the size of the rectangle needs to be $z_\alpha = \Phi^{-1}(1-\alpha/2)$, where $\alpha = 1-(1-\beta)^m$. In summary, the $(1-\beta) * 100\%$ confidence rectangle for $\boldsymbol{\mu}$ is:

$$R(\boldsymbol{\mu}(n), \boldsymbol{\sigma}(n), z_\alpha) = \{\boldsymbol{x} \in R^m : \boldsymbol{\mu}(n) - z_\alpha \boldsymbol{\sigma}(n) \leq \boldsymbol{x} \leq \boldsymbol{\mu}(n) + z_\alpha \boldsymbol{\sigma}(n)\}.$$

In the subsequent analysis, it is often necessary to shrink the confidence rectangle centered at $\boldsymbol{\mu}(n)$, either by decreasing the level of confidence or by increasing the number of simulation

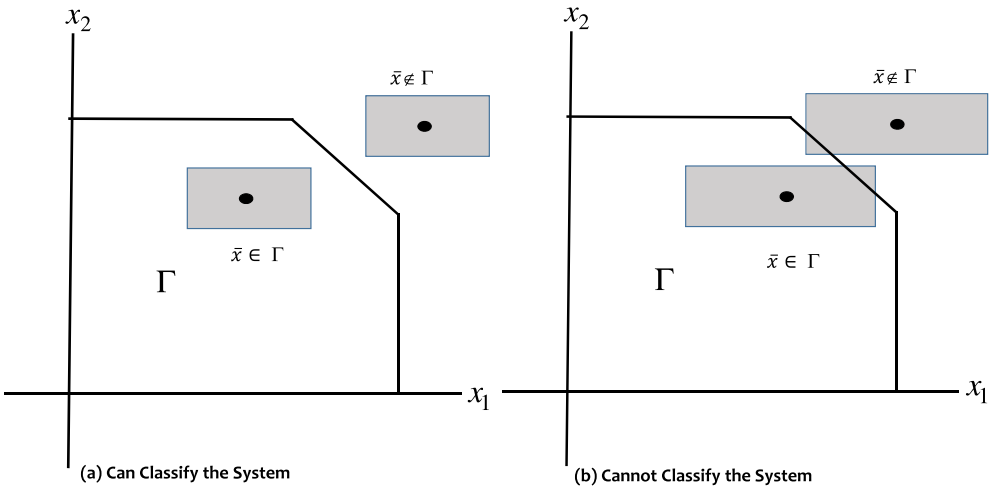

Fig. 3. Confidence rectangles that allow and do not allow a system to be classified.

replications (corresponding to increased sampling while retaining the desired confidence level). In this approach, the center of the rectangle is temporarily fixed with the notation $\boldsymbol{\mu}_{last} = \boldsymbol{\mu}(n)$. Therefore, the following more general notation is used for the rectangle centered at $\boldsymbol{\mu}_{last}$ in the direction $\boldsymbol{\sigma}(\tilde{n})$ of size $t \geq 0$:

$$R_t(\tilde{n}) = R(\boldsymbol{\mu}_{last},\ \boldsymbol{\sigma}(\tilde{n}),\ t) = \{\boldsymbol{x} \in R^m : \boldsymbol{\mu}_{last} - t\boldsymbol{\sigma}(\tilde{n}) \leq \boldsymbol{x} \leq \boldsymbol{\mu}_{last} + t\boldsymbol{\sigma}(\tilde{n})\}. \tag{9}$$

The way the confidence rectangle $R_z(n)$ is used to classify the system when $\boldsymbol{\mu}(n) \in \Gamma$ is illustrated in Figure 3, where it is seen that if the confidence rectangle is contained completely within Γ, then the system is classified (perhaps incorrectly) as in Γ. However, if the confidence rectangle "spills out" of Γ, then the system cannot yet be classified with the desired level of confidence, in which case additional simulation replications are needed to shrink the rectangle. In that regard, as m gets larger, the number of additional replications needed will be overestimated, since $\alpha = 1 - (1 - \beta)^m$.

As seen in Figure 3, if $R_z(n)$ lies entirely in Γ, then the system is classified as in Γ with probability $1 - \beta$. One way to check this is to see whether all 2^m extreme points of the confidence rectangle are in Γ for, if so, $R_z(n) \subset \Gamma$. This approach is computationally practical only for small values of m. When m is large, an alternative approach using the c constraints of Γ is available to determine not only whether $R_z(n) \subset \Gamma$, but also for any sized rectangle centered at $\boldsymbol{\mu}(n)$ in the direction $\boldsymbol{\sigma}(\tilde{n})$, for any integer $\tilde{n} \geq 0$, as shown in the following proposition.

Proposition 3.1. *Given $\boldsymbol{\mu}_{last} = \boldsymbol{\mu}(n)$ and an integer $\tilde{n} \geq 0$, define, for each constraint $k = 1, \ldots, c$ of Γ,*

$$\hat{\tau}_k(\tilde{n}) = \frac{b_k - a_k^T \boldsymbol{\mu}_{last}}{a_k^T \boldsymbol{d}}, \qquad \text{where } d_j = \begin{cases} +\sigma_j(\tilde{n}) & \text{if } a_{kj} \geq 0 \\ -\sigma_j(\tilde{n}) & \text{if } a_{kj} < 0 \end{cases} \tag{10}$$

and

$$\hat{\tau}(\tilde{n}) = \min\{\hat{\tau}_k(\tilde{n}) : k = 1, \ldots, c\}.$$

If $t \leq \hat{\tau}(\tilde{n})$, then $R_t(\tilde{n}) \subset \Gamma$. Furthermore, for any $t > \hat{\tau}(\tilde{n})$, $R_t(\tilde{n}) \not\subset \Gamma$.

As a result of Proposition 3.1, if $z \leq \hat{\tau}(n)$, then the confidence rectangle $R_z(n) \subset \Gamma$ and the system is classified as in Γ with probability $1 - \beta$.

3.2.2 Using Hyperplanes of Γ *to Classify a System.* The second approach for attempting to classify a system is to use the hyperplanes that define Γ, together with the following inequality:

$$\begin{aligned} P(\boldsymbol{\mu} \notin \Gamma | \mathbf{X}(1), \ldots, \mathbf{X}(n)) &= P(\cup_{\ell=1}^{c} \mathbf{a}_\ell^T \boldsymbol{\mu} > b_\ell | \mathbf{X}(1), \ldots, \mathbf{X}(n)) \\ &\leq c \max_{\ell=1,\ldots,c} P(\mathbf{a}_\ell^T \boldsymbol{\mu} > b_\ell | \mathbf{X}(1), \ldots, \mathbf{X}(n)). \end{aligned} \tag{11}$$

The Gaussian conjugate prior assumption (cf. Equations (1)–(4)), together with a Chernoff bound (Dembo and Zeitouni [7]) results in,

$$c \max_{\ell=1,\ldots,c} P(\mathbf{a}_\ell^T \boldsymbol{\mu} > b_\ell | \mathbf{X}(1), \ldots, \mathbf{X}(n)) \leq \exp\left(\log(c) - \min_{\ell=1,\ldots,c} \frac{(\mathbf{a}_\ell^T \boldsymbol{\mu}(n) - b_\ell)^2}{2 \sum_{j=1}^{m} a_{\ell j}^2 \sigma_{\ell j}^2(n)} \right).$$

We conclude that the union bound used in Equation (11) is tight for n of order larger than $\log(c)$.

The system is classified as in Γ with probability $1 - \beta$ if $P(\boldsymbol{\mu} \notin \Gamma | \mathbf{X}(1), \ldots, \mathbf{X}(n)) \leq \beta$. From the inequality in Equation (11) to be satisfied, a sufficient condition is that

$$\max_{\ell=1,\ldots,c} P(\mathbf{a}_\ell^T \boldsymbol{\mu} > b_\ell | \mathbf{X}(1), \ldots, \mathbf{X}(n)) \leq \beta / c. \tag{12}$$

As $\mu_j | \mathbf{X}(1), \ldots, \mathbf{X}(n) \sim N(\mu_j(n), \sigma_j^2(n))$, the linear combination $\boldsymbol{a}_\ell^T \boldsymbol{\mu} | \mathbf{X}(1), \ldots, \mathbf{X}(n) \sim N(\boldsymbol{a}_\ell^T \boldsymbol{\mu}(n), \sum_j a_{\ell j}^2 \sigma_j^2(n))$; hence, by subtracting this mean and dividing by this standard deviation, the inequality in Equation (12) becomes:

$$\max_{\ell=1,\ldots,c} \Phi(s_\ell) \leq \beta / c, \text{ where } s_\ell = \frac{a_\ell^T \boldsymbol{\mu}(n) - b_\ell}{(\sum_j a_{\ell j}^2 \sigma_j^2(n))^{1/2}}. \tag{13}$$

In summary, if Equation (13) is satisfied, then the system is classified as in Γ with probability $1 - \beta$.

When a system cannot be classified as either in Γ or not in Γ at the desired level of confidence, it is necessary to determine the number of additional simulation replications, if any, to perform. Various approaches for doing so in such a way that the system might be classified with probability $1 - \beta$ after performing those additional replications are proposed in Sections 4 and 5, after which the complete algorithm is presented in Section 6.

4 A BAYESIAN APPROACH FOR DETERMINING THE REQUIRED NUMBER OF ADDITIONAL SAMPLES

In this section, we present a Bayesian approach to determine the number of additional replications required to classify a system with probability $1 - \beta$, in expectation. This is an extension of the adaptive algorithms introduced by Szechtman and Yücesan [20] who consider only linear constraints that form an orthant. Another interesting empirical Bayesian method has been proposed by Singham and Szechtman [17] for multiple comparisons with a standard where false discovery rates are controlled. While their methodology is quite distinct, the *standard* in their setting may be interpreted as one of the constraints in our setting. Our method depends on whether the current posterior mean, $\boldsymbol{\mu}(n)$, is in Γ or not.

4.1 Determining the Number of Additional Samples When $\mu(n) \notin \Gamma$

When $\boldsymbol{\mu}(n) \notin \Gamma$, the goal is to determine the smallest number of additional samples, Δn, required to have the posterior probability that a system's mean performance is correctly classified to be not in Γ be at least $1 - \beta$, in expectation. This is accomplished by solving the following optimization problem for each violated constraint $\ell \in L$:

$$\begin{aligned} &\min \Delta n \\ \text{s.t. } &E_{\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)}[P(\boldsymbol{a}_\ell^T \boldsymbol{\mu} > b_\ell | \mathbf{X}(1), \ldots, \mathbf{X}(n + \Delta n))] \geq 1 - \beta \\ &\Delta n \in \mathbb{N}. \end{aligned} \tag{14}$$

The interpretation of Equation (14) is that Δn is the number of additional samples required so the system is classified correctly with probability $1-\beta$, in expectation.

From Equations (1–4) it follows that the distribution of $\boldsymbol{a}_\ell^T\boldsymbol{\mu}|\mathbf{X}(1),\ldots,\mathbf{X}(n)$ is normal with mean

$$\sum_{j=1}^{m} a_{\ell,j}\frac{\frac{\mu_j(0)}{\sigma_j^2(0)}+\frac{n\bar{X}_j(n)}{\sigma_j^2}}{1/\sigma_j^2(0)+n/\sigma_j^2},$$

and variance

$$\sum_{j=1}^{m} a_{\ell,j}^2\frac{1}{1/\sigma_j^2(0)+n/\sigma_j^2}.$$

Likewise, the distribution of $X_j(n+1)|X_j(1),\ldots,X_j(n)$ is normal with mean

$$\gamma_j(n)=\frac{\mu_j(0)/\sigma_j^2(0)+n\bar{X}_j(n)/\sigma_j^2}{1/\sigma_j^2(0)+n/\sigma_j^2} \tag{15}$$

and variance

$$v_j(n)^2=\sigma_j^2+\frac{1}{1/\sigma_j^2(0)+n/\sigma_j^2}. \tag{16}$$

If Δn independent new samples $X_j(n+1),\ldots,X_j(n+\Delta n)\sim N(\gamma_j(n),v_j^2(n))$ are drawn, then

$$\begin{aligned}
&E_{\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)}[P(\boldsymbol{a}_\ell^T\boldsymbol{\mu}>b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n+\Delta n))]\\
&=\int P(\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)\in dx_{n+1},\ldots,dx_{n+\Delta n}|\mathbf{X}(1),\ldots,\mathbf{X}(n))\\
&\quad\times P(\boldsymbol{a}_\ell^T\boldsymbol{\mu}>b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n+\Delta n))\\
&=E_{Z_1,\ldots,Z_m}\left[\Phi\left(\frac{-b_\ell+\sum_{j=1}^m a_{\ell,j}\frac{\frac{\mu_j(0)}{\sigma_j^2(0)}+\frac{n\bar{X}_j(n)+\Delta n\gamma_j(n)+\sqrt{\Delta n}v_j(n)Z_j}{\sigma_j^2}}{1/\sigma_j^2(0)+(n+\Delta n)/\sigma_j^2}}{\left(\sum_{j=1}^m\frac{a_{\ell,j}^2}{1/\sigma_j^2(0)+(n+\Delta n)/\sigma_j^2}\right)^{1/2}}\right)\right]\\
&=P\left(Z_0\le\frac{-b_\ell+\sum_{j=1}^m a_{\ell,j}\frac{\frac{\mu_j(0)}{\sigma_j^2(0)}+\frac{n\bar{X}_j(n)+\Delta n\gamma_j(n)+\sqrt{\Delta n}v_j(n)Z_j}{\sigma_j^2}}{1/\sigma_j^2(0)+(n+\Delta n)/\sigma_j^2}}{\left(\sum_{j=1}^m\frac{a_{\ell,j}^2}{1/\sigma_j^2(0)+(n+\Delta n)/\sigma_j^2}\right)^{1/2}}\right),
\end{aligned}$$

for independent $Z_0,Z_1,\ldots,Z_m\sim N(0,1)$ and a violated constraint $\ell\in L$. Since we end up with a linear combination of independent normal random variables, we conclude that

$$\begin{aligned}
&E_{\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)}[P(\boldsymbol{a}_\ell^T\boldsymbol{\mu}>b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n+\Delta n))]\\
&\qquad=\Phi\left(\frac{-b_\ell+\sum_{j=1}^m a_{\ell,j}\frac{\frac{\mu_j(0)}{\sigma_j^2(0)}+\frac{n\bar{X}_j(n)+\Delta n\gamma_j(n)}{\sigma_j^2}}{1/\sigma_j^2(0)+(n+\Delta n)/\sigma_j^2}}{\left(\sum_{j=1}^m a_{\ell,j}^2\left(\frac{1}{1/\sigma_j^2(0)+(n+\Delta n)/\sigma_j^2}+\frac{\Delta n v_j^2(n)/\sigma_j^4}{(1/\sigma_j^2(0)+(n+\Delta n)/\sigma_j^2)^2}\right)\right)^{1/2}}\right).
\end{aligned}$$

Replacing for the value of γ_j and $v_j^2(n)$ as in Equations (15–16) leads after some algebra to

$$E_{\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)}[P(\boldsymbol{a}_\ell^T\boldsymbol{\mu} > b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n+\Delta n))] = \Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(\sigma_j^2/\sigma_j^2(0)+n+\Delta n)^2}\right)^{1/2}}\right), \tag{17}$$

for $\ell \in L$.

For $\Delta n = 0$ and every violated constraint $\ell \in L$,

$$\Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m \frac{a_{\ell,j}^2}{1/\sigma_j^2(0)+n/\sigma_j^2}\right)^{1/2}}\right) < 1-\beta,$$

while, since $-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j > 0$ for $\boldsymbol{\mu}(n) \notin \Gamma$, it easily follows from Equation (17) that

$$E_{\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)}[P(\boldsymbol{a}_\ell^T\boldsymbol{\mu} > b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n+\Delta n))] \geq 1-\beta$$

for Δn sufficiently large.

Hence, as in Equation (7), a value of $\Delta n > 0$ to classify a system can be found by solving

$$\min_{\Delta n\in\mathbb{N}:\forall \ell\in L} \Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(\sigma_j^2/\sigma_j^2(0)+n+\Delta n)^2}\right)^{1/2}}\right) \geq 1-\beta. \tag{18}$$

A conservative bound for Δn can be computed analytically, since

$$\frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(\sigma_j^2/\sigma_j^2(0)+n+\Delta n)^2} \leq \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(n+\Delta n+\min_{k=1,\ldots,m}\sigma_k^2/\sigma_k^2(0))^2},$$

in the denominator above, meaning that

$$\min_{\Delta n\in\mathbb{N}:\forall \ell\in L} \Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(\sigma_j^2/\sigma_j^2(0)+n+\Delta n)^2}\right)^{1/2}}\right)$$

$$\geq \min_{\Delta n\in\mathbb{N}:\forall \ell\in L} \Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left((n+\Delta n+\min_{k=1,\ldots,m}\{\sigma_k^2/\sigma_k^2(0)\})^{-2}\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)}\right)^{1/2}}\right).$$

Therefore, a conservative value of Δn can be found by solving the quadratic equation

$$\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\Phi^{-1}(1-\beta)}\right)^2 = (n+\Delta n+\min_{k=1,\ldots,m}\sigma_k^2/\sigma_k^2(0))^{-2}\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)}, \tag{19}$$

for each violated constraint $\ell \in L$, and then setting u equal to the smallest positive root of Equation (19) among the violated constraints. This upper bound guarantees that the roots in Equation (18)

can be found using bisection in $O(\log u/\log 2)$ iterations. The benefit of this approach is that it yields a closed-form expression, in terms of a quadratic equation, for the number of extra samples required to classify each violated constraint ℓ. The bound is tight when the variance ratios $\sigma_k^2/\sigma_k^2(0)$ are small in relation to n, or have small variabilty.

4.2 Determining the Number of Additional Samples When $\boldsymbol{\mu}(n) \in \Gamma$

When $\boldsymbol{\mu}(n) \in \Gamma$, with

$$P(\boldsymbol{a}_\ell^T\boldsymbol{\mu} > b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n)) = \Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m \frac{a_{\ell,j}^2}{1/\sigma_j^2(0)+n/\sigma_j^2}\right)^{1/2}}\right) > \beta/c,$$

for $\ell = 1,\ldots,c$, we wish to find the number of additional samples that make the posterior probability that a system's mean performance is correctly classified to be in Γ be at least $1-\beta$, in expectation.

Consider again

$$E_{\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)}[P(\boldsymbol{a}_\ell^T\boldsymbol{\mu} > b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n+\Delta n))]$$
$$= \Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(\sigma_j^2/\sigma_j^2(0)+n+\Delta n)^2}\right)^{1/2}}\right),$$

for any constraint $\ell = 1,\ldots,c$. The numerator of the expression in the parentheses on the right-hand side is negative when the posterior mean is in Γ, ensuring that $E_{\mathbf{X}(n+1),\ldots,\mathbf{X}(n+\Delta n)}[P(\boldsymbol{a}_\ell^T\boldsymbol{\mu} > b_\ell|\mathbf{X}(1),\ldots,\mathbf{X}(n+\Delta n))] \to 0$ as Δn increases. Hence, as in Equation (13), we solve for the integer value of Δn for which

$$\min_{\Delta n} \max_{\ell=1,\ldots,c} \Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(\sigma_j^2/\sigma_j^2(0)+n+\Delta n)^2}\right)^{1/2}}\right) \le \beta/c. \tag{20}$$

This problem can be solved by finding a root of

$$\Phi\left(\frac{-b_\ell + \sum_{j=1}^m a_{\ell,j}\gamma_j}{\left(\sum_{j=1}^m a_{\ell,j}^2 \frac{(\sigma_j^2/\sigma_j^2(0)+n)^2+\Delta n(2\sigma_j^2/\sigma_j^2(0)+2n+1)}{(1/\sigma_j^2(0)+n/\sigma_j^2)(\sigma_j^2/\sigma_j^2(0)+n+\Delta n)^2}\right)^{1/2}}\right) = \beta/c,$$

for $\ell = 1,\ldots,c$, and then returning the largest of the c roots.

5 A NON-BAYESIAN APPROACH FOR DETERMINING THE REQUIRED NUMBER OF ADDITIONAL SAMPLES

When additional simulation replications are performed, the estimates of the means may change. However, for the approach proposed now, it is assumed that those estimates remain fixed for a given system at $\boldsymbol{\mu}(n) = \boldsymbol{\mu}_{last}$.

As in the Bayesian case, determining the number of additional simulations to perform when a system cannot be classified with probability $1-\beta$ depends on whether the current estimate of the mean, namely, $\boldsymbol{\mu}_{last}$, is in or not in Γ. Non-Bayesian approaches for doing so in each of these cases are now proposed.

5.1 Determining the Number of Additional Samples When $\boldsymbol{\mu}_{last} \notin \Gamma$

Recall from Section 3.1 that when a system for which $\boldsymbol{\mu}_{last} \notin \Gamma$ is not classified at the desired level of confidence, then for every violated constraint $\ell \in L = \{\ell : \boldsymbol{a}_\ell^T \boldsymbol{\mu}_{last} > b_\ell\}$, $\Phi(s_\ell) < 1 - \beta$, where s_ℓ is defined in Equation (7). The approach proposed now for determining the number of additional simulation replications is based on the observation that, for any such violated constraint, had n been large enough, the value of $\boldsymbol{\sigma}(n)$ from Equation (3) would have been sufficiently small so Equation (5) would allow the system to be classified as not in Γ with probability $1 - \beta$. Thus, the goal is to determine $n^* > n$ so $\boldsymbol{\sigma}(n^*)$ in Equation (5) would allow the system to be classified as not in Γ with probability $1 - \beta$, for then the number of additional simulation replications to be performed using this approach is $\Delta n = n^* - n$.

To that end, if the violated constraints at $\boldsymbol{\mu}_{last}$ do not allow the system to be classified, then each such constraint $\ell \in L$ provides a separating hyperplane for which one can compute the smallest number of samples, n_ℓ, that would have provided a value of $\boldsymbol{\sigma}(n_\ell)$ in Equation (3) for which $\Phi(s_\ell) \geq 1 - \beta$, thus allowing the system to be classified as not in Γ using Equation (7). Thinking of $\tilde{n}$ as a continuous variable, the desired value of n_ℓ is the smallest value of $\tilde{n} > n$ for which $\boldsymbol{\sigma}(n_\ell)$ results in $\Phi(s_\ell) = 1 - \beta$, that is, n_ℓ is the smallest value of $\tilde{n} > n$ for which

$$f(\tilde{n}) = \boldsymbol{a}_\ell^{2T} \boldsymbol{\sigma}^2(\tilde{n}) - \left(\frac{\boldsymbol{a}_\ell^T \boldsymbol{\mu}_{last} - b_\ell}{\Phi^{-1}(1 - \beta)} \right)^2 = 0. \tag{21}$$

Assuming that n is continuous, the function $f : [0, \infty) \to R^1$ in (21) is convex and $f(n) > 0$ while $\lim_{\tilde{n} \to \infty} f(\tilde{n}) < 0$; hence, one can use the approach described in Appendix A to find a zero of f, namely, n_ℓ. The number of additional samples to collect for this system is then

$$\Delta n = n^* - n, \quad \text{where } n^* = \min\{\lceil n_\ell \rceil\}. \tag{22}$$

Note that an even smaller value for Δn can be obtained by using the best separating hyperplane, defined in Equation (6), although finding this separating hyperplane requires solving the associated nonlinear program.

5.2 Determining the Number of Additional Samples When $\boldsymbol{\mu}_{last} \in \Gamma$

When the current estimate of the mean $\boldsymbol{\mu}_{last}$, obtained after performing n simulation replications, is in Γ and the system cannot be classified with probability $1 - \beta$, determining the number of additional simulation replications depends on whether confidence rectangles (see Section 3.2.1) or hyperplanes (see Section 3.2.2) are used to classify the system.

5.2.1 Using Confidence Rectangles. The following heuristics are based on the observation that, had n been large enough, the value of $\boldsymbol{\sigma}(n)$ would have been sufficiently small so the confidence rectangle centered at $\boldsymbol{\mu}_{last}$ lies entirely in Γ, thus allowing the system to be classified as in Γ with probability $1 - \beta$.

Heuristic 1: The objective of this first heuristic is to determine the minimum number of samples, n^*, that would shrink the confidence rectangle so $R_z(n^*) \subset \Gamma$. From Proposition 3.1, the desired value for n^* is the smallest value of $\tilde{n}$ for which $\hat{\tau}(\tilde{n}) \geq z$. There is such a value for n^*, because $\hat{\tau}(n) < z$ and $\hat{\tau}(\tilde{n}) \to \infty$ as $\tilde{n} \to \infty$, since $\boldsymbol{\sigma}(\tilde{n}) \to \mathbf{0}$ monotonically as $\tilde{n} \to \infty$. As a result, it is possible to use bisection on the interval $[n, \infty)$ to find n^*. However, because of the mathematical properties of the functions $\hat{\tau}_k(\tilde{n})$ in Equation (10), a more efficient alternative is available. To this end, consider the following functions $g_k : [0, \infty) \to R^1$, for each $k = 1, \ldots, c$:

$$g_k(\tilde{n}) = \frac{1}{\hat{\tau}_k(\tilde{n})} = \frac{a_k^T \boldsymbol{d}}{b_k - a_k^T \boldsymbol{\mu}_{last}} = \frac{\sum_{j=1}^m a_{kj}^T d_j}{b_k - a_k^T \boldsymbol{\mu}_{last}} = \frac{\sum_{j:a_{kj} \geq 0}^m a_{kj} \sigma_j(\tilde{n}) - \sum_{j:a_{kj} < 0}^m a_{kj} \sigma_j(\tilde{n})}{b_k - a_k^T \boldsymbol{\mu}_{last}}. \tag{23}$$

The function in Equation (23), being the sum of nonnegative constants times the convex functions $\sigma_j(\tilde{n})$, is also convex. Finally, finding the value of $\tilde{n}$ for which $\hat{\tau}(\tilde{n}) = z$ is equivalent to finding the 0 of the following function $f : [0, \infty) \to R^1$:

$$f(\tilde{n}) = \max\{g_k(\tilde{n}) : k = 1, \ldots, c\} - \frac{1}{z}. \tag{24}$$

Since the function f in Equation (24) is the maximum of convex functions, it is also convex and has a zero, because $f(n) > 0$ while $\lim_{\tilde{n}\to\infty} f(\tilde{n}) < 0$. Assuming that n is continuous, this zero can be found using the approach described in Appendix A.

In summary, when $R_z(n)$ is not contained inside Γ, it is possible to use a subgradient Newton method on the function f in Equation (24) to find $n^* = \min\{\tilde{n} > n : \hat{\tau}(\tilde{n}) \geq z\}$ for then $R_z(n^*) \subset \Gamma$. Then the number of additional simulation replications to be performed is $\Delta n = n^* - n$.

Heuristic 2: The number of additional simulation replications under Heuristic 1 requires performing a subgradient Newton algorithm. The heuristic proposed here avoids this need. Rather than finding n^* so $R_z(n^*) \subset \Gamma$, Heuristic 2 attempts to find first the largest value of $t > 0$ so $R_t(n) \subset \Gamma$, which, according to Proposition 3.1, is $t = \hat{\tau}(n)$. The desired value for n^* is the smallest integer $\tilde{n} > n$ so $R_z(\tilde{n}) \subset R_t(n) \subset \Gamma$. The value of n^* can be found using the formula for $\sigma_j^2(n)$ in Equation (3) to compute, for each coordinate $j = 1, \ldots, m$, the value, $\hat{n}_j$, so

$$\sigma_j(\hat{n}_j) = \frac{t}{z}\sigma_j(n).$$

Using Equation (3) and solving for $\hat{n}_j$ results in the following value:

$$\hat{n}_j = \left[\left(\frac{z}{t}\right)^2 - 1\right]\left(\frac{\sigma_j^2(n)}{\sigma_j^2(0)}\right) + \left(\frac{z}{t}\right)^2 n \quad \text{and so} \quad n^* = \max\{\lceil \hat{n}_j \rceil\}. \tag{25}$$

The number of additional simulation replications to be performed using this heuristic is then $\Delta n = n^* - n$.

5.2.2 Using Hyperplanes. Determining the number of additional simulation replications when hyperplanes are used to classify the system is based on the observation that, had n been large enough, the value of $\boldsymbol{\sigma}(n)$ from Equation (3) would have been sufficiently small so Equation (13) would allow the system to be classified as in Γ with probability $1 - \beta$. Thus, the goal is to determine $n^* > n$ so $\boldsymbol{\sigma}(n^*)$ in Equation (13) would allow the system to be classified as in Γ with probability $1 - \beta$, for then the number of additional simulation replications to be performed is $\Delta n = n^* - n$.

If the constraints of Γ do not allow the system to be classified, then each constraint ℓ provides the ability to compute the smallest number of samples, n_ℓ, that would have provided a value of $\boldsymbol{\sigma}(n_\ell)$ in Equation (3) for which $\Phi(s_\ell) \leq \beta/c$. Defining $n^* = \max_\ell\{n_\ell\}$ would then allow the system to be classified as in Γ using Equation (13). Thinking of $\tilde{n}$ as a continuous variable, the desired value of n_ℓ is the smallest value of $\tilde{n} > n$ for which $\boldsymbol{\sigma}(n_\ell)$ results in $\Phi(s_\ell) = \beta/c$, that is, n_ℓ is the smallest value of $\tilde{n} > n$ for which

$$f(\tilde{n}) = \boldsymbol{a}_\ell^{2T}\boldsymbol{\sigma}^2(\tilde{n}) - \left(\frac{\boldsymbol{a}_\ell^T\boldsymbol{\mu}_{last} - b_\ell}{\Phi^{-1}(\beta/c)}\right)^2 = 0. \tag{26}$$

Assuming that n is continuous, the function $f : [0, \infty) \to R^1$ in Equation (26) is convex and $f(n) > 0$ while $\lim_{\tilde{n}\to\infty} f(\tilde{n}) < 0$; hence, one can again use the approach described in Appendix A to find a zero of f, namely, n_ℓ. The number of additional samples to perform for this system is then given in Equation (22).

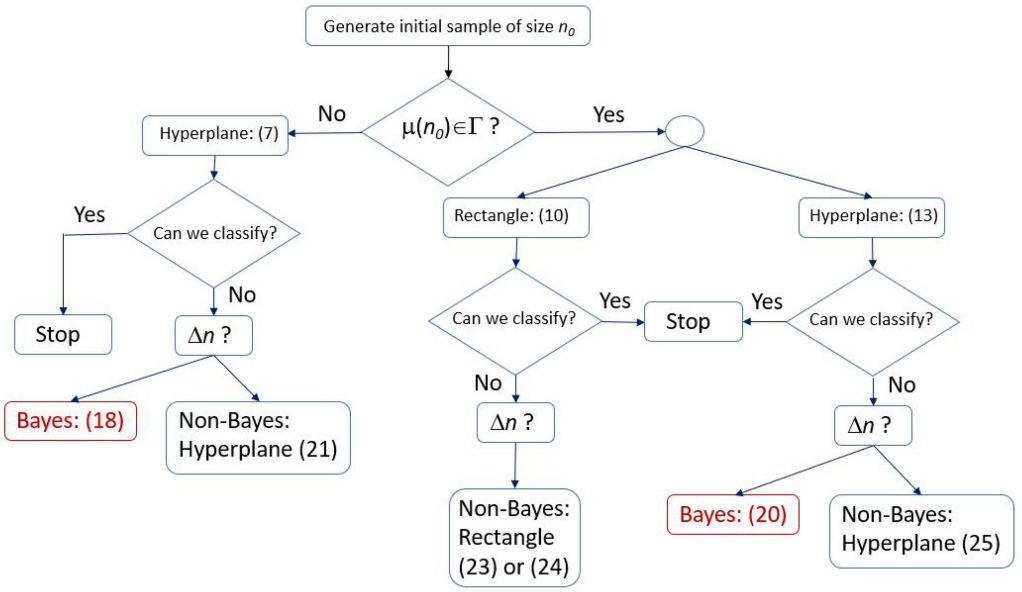

Fig. 4. The flow logic of the Bayesian and frequentist algorithms.

From here on, $\Delta n(\beta)$ refers to the number of additional simulation replications to perform for a system—in addition to the n that have already been performed—given a desired probability $1 - \beta$ of correct classification.

6 THE COMPLETE ALGORITHM

The algorithms for classifying the r systems as in or not in Γ when Γ is defined by a finite collection of linear inequalities are summarized in Figure 4. All three algorithms start by taking an initial sample of $n(0)$ from each system to assess whether the estimated mean is in Γ. If the sample average is not in Γ , a separating hyperplane is used to determine whether the system can be classified with the desired (user-specified) level of confidence. If such a classification is possible, the algorithm stops. Otherwise, there are two possible approaches for determining the additional number of samples that would enable classification: a Bayesian approach using Equation (18) or a frequentist approach using Equation (21). After the additional $\Delta(n)$ samples are collected, the system is classified based on the updated estimate of the sample mean. However, if the estimated mean based on the initial sample is in Γ, then there are several possible ways to proceed. In a purely frequentist approach, one can construct a rectangle centered at the sample average using Equation (10). If this rectangle is not fully contained in the feasible region, one can determine the additional number of samples using Equations (23) or (24) to "shrink" the confidence rectangle so it is completely contained within Γ. Alternatively, one can find a separating hyperplane to determine whether the system can be classified with the desired (user-specified) level of confidence. If such a classification is possible, the algorithm stops. Otherwise, there are two possible approaches for determining the additional number of samples that would enable classification: a Bayesian approach using Equation (20) or a frequentist approach using Equation (25). After the additional $\Delta(n)$ samples are collected, the system is classified based on the updated estimate of the sample mean. Results of computational experiments with these algorithms are presented in Section 7.

Table 1. Parameter Settings for the Experiments

Level	n_0	$\mu_{ij}(0)$	$\sigma_{ij}(0)$	σ_{ij}	σ_{ij}	ρ_{ij}
HIGH	5	$(-1, -1)$	10	10	unknown	0.7
LOW	10	$(1, 1)$	1	1	known	0.3

7 COMPUTATIONAL RESULTS

In this section, we illustrate the performance of the algorithms through simple numerical examples. First, we demonstrate the validity of the algorithms (i.e., their ability to deliver the desired probability of correct classification) under various experimental settings, including the cases with unknown variances and correlated performance metrics. We then compare the performance of our algorithms to that of the feasibility determination algorithm proposed by Batur and Kim [3]. Two of the algorithms are not only effective in their performance, but, compared with other feasibility determination approaches, they also appear to be competitive.

7.1 Validity

In this subsection, the performance of the three algorithms, namely, Bayes, separating hyperplanes, and rectangles as summarized in Figure 4, is illustrated through small examples. In Figure 4, the numbers in parentheses refer to the equations used to determine the additional number of simulation replications needed for correct classification. In the examples, the feasible region, Γ, is defined to be the negative quadrant in $m = 2$ dimensions. To demonstrate the validity and the robustness of the algorithms, five key parameters associated with classifying $r = 1,000$ systems with confidence $1 - \beta = 0.95$ are varied: the number of initial samples, n_0; the prior value for the mean, $\mu_{ij}(0)$; the prior value for the variance, $\sigma^2_{ij}(0)$, both specified by the analyst; and the true variance, σ^2_{ij}, whereby we also investigate the setting where the actual variance is not known, but must be estimated in the first stage of the algorithms using the initial sample.

The parameter settings are shown in Table 1. Experimenting with two levels for each parameter results in 32 design points. The **HIGH** settings, i.e., the parameter value with an H, are meant to depict more challenging conditions for the three algorithms. For instance, the design point with **HHHHH** represents the most challenging setting for the algorithms with a small initial sample of $n_0 = 5$, the prior mean, $\mu_{ij}(0) = (-1, -1)$, in Γ, a high prior value for $\sigma_{ij}(0) = 10$, and a high value for σ_{ij}, which, if unknown, would be estimated from the initial samples, based on the values in Table 1. The **LOW** settings are more forgiving.

The observed average confidence level at each of the design points for all three algorithms is shown in Figure 5, while the average performance of the three algorithms is summarized in Table 2, where the observed confidence level, the average number of samples required to achieve the desired confidence level, the median number of samples required to achieve the desired confidence level, and the average number of systems that are incorrectly classified (all averaged over 10 independent macro replications at each design point) are tabulated for the desired confidence level, $1 - \beta = 0.95$.

Several observations are in order. First, the Bayesian algorithm appears to be the most robust approach for feasibility determination in the sense that it delivers the desired confidence level under *all* experimental settings—including the one with unknown variances. Unfortunately, there is no "free lunch": The Bayesian algorithm achieves this robust performance by requiring a large sample size to consistently deliver the desired level of confidence. Second, while the non-Bayesian algorithm based on hyperplanes requires three to ten times fewer samples than the Bayesian approach, its performance deteriorates in settings where the prior value of the mean is in Γ. This is

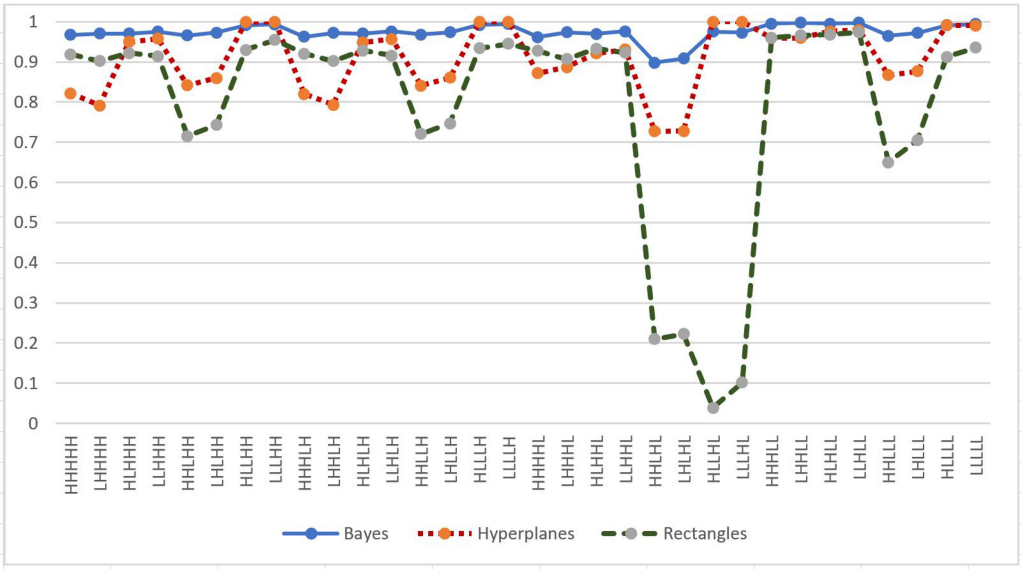

Fig. 5. Observed average confidence levels for the algorithms.

Table 2. Observed Average Performance (Standard Error) of the Three Algorithms

Bayes	**Known Variance**	**Unknown Variance**	**Overall**
$\hat{\beta}$	0.9715 (0.0073)	0.9760 (0.0026)	**0.9737**
Avg Sample Size	19,284 (8868)	67,297 (22676)	**43,291**
Median Sample Size	150.46 (75.64)	9.16 (0.65)	**79.81**
Avg Incorrectly Classified	2.5 (0.43)	4.7 (0.55)	**3.6**
Hyperplanes			
$\hat{\beta}$	0.9165 (0.0218)	0.9025 (0.0201)	**0.9095**
Avg Sample Size	5,670 (3032)	6,374 (3930)	**6,022**
Median Sample Size	14.73 (2.03)	29.18 (6.05)	**19.22**
Avg Incorrectly Classified	2.11 (0.67)	8.57 (2.19)	**5.34**
Rectangles			
$\hat{\beta}$	0.7083 (0.0877)	0.8756 (0.0218)	**0.7920**
Avg Sample Size	18,446 (7428)	50,475 (21303)	**34,461**
Median Sample Size	13.59 (2.78)	10.15 (1.41)	**11.87**
Avg Incorrectly Classified	2.03 (0.46)	4.85 (0.66)	**3.44**

a challenging setting where all constraints must be checked to ensure correct classification. The performance further degrades in settings where the initial sample size is small and the true (known or unknown) variance is large. Third, the non-Bayesian algorithm based on rectangles is the most vulnerable. While it is mostly dominated by the first two algorithms, its performance is anemic in settings with a small initial sample and large variances, where the true variance is estimated from the data. In fact, the most challenging setting for all three algorithms is the one where the variances, σ_{ij}^2, are high, for which the achieved estimated confidence level is as low as 0.8978 for the Bayesian algorithm, 0.7270 for the hyperplane, and an anemic 0.0391 for the rectangles. A larger value for the initial sample size, n_0, offers only partial relief. Overall, we recommend our Bayesian

Table 3. Observed Average Performance (Standard Error) of the Three Algorithms in Correlated Settings

Alg.	Setting	Correlation	$\hat{\beta}$	Avg Sample Size	Med Sample Size	Avg Incorr Class'd
Bayes	HLLHL	HIGH	0.9447 (0.0021)	511.02 (7.11)	393.4 (4.42)	5 (1.26)
		LOW	0.9590 (0.0010)	527 (5.77)	400.7 (3.24)	2.6 (0.22)
		UNCORR	**0.9755** (0.0013)	**305** (2.10)	**270.8** (3.51)	**2.2** (0.37)
	HHLHL	HIGH	0.8778 (0.0025)	911.3 (23.35)	666.4 (48.97)	2.6 (0.6)
		LOW	0.9152 (0.0035)	1,211.9 (272.27)	719.1 (3.15)	3 (1.05)
		UNCORR	**0.8978** (0.0018)	**1,166.8** (115.84)	**836** (8.09)	**2.8** (0.73)
Hyper-planes	HLLHL	HIGH	1.0 (0.0)	20.1 (0.05)	20.2 (0.04)	0.0 (0.0)
		LOW	1.0 (0.0)	20.4 (0.04)	20.1 (0.03)	0.0 (0.0)
		UNCORR	**1.0** (0.0)	**20.8** (0.06)	**20.0** (0.04)	**0.0** (0.0)
	HHLHL	HIGH	0.7794 (0.0029)	21.2 (2.14)	19.5 (1.67)	0.0 (0.0)
		LOW	0.7799 (0.0015)	21.2 (1.67)	20.3 (1.32)	0.0 (0.0)
		UNCORR	**0.7270** (0.0013)	**20.7** (0.97)	**20.0** (0.66)	**0.0** (0.0)

algorithm provided that the execution cost of each simulation replication is not prohibitively high. In that case, the non-Bayesian approach with the hyperplanes may provide a rough-cut solution. In fact, if the prior value of the variance provided by the analyst is not too high, the hyperplanes approach delivers the desired confidence level provided that the initial sample size is not too low.

Finally, we have relaxed Assumption 2 by considering settings where performance metrics are correlated. As described in Section 2, this is done through a transformation of the coordinate axes for the correlated random variables. The results are summarized in Table 3 where the Bayes and the separating hyperplanes heuristics are tested at two design points: HLLHL (i.e., $n_0 = 5$, $\mu_0 = (1, 1)$, $\sigma_0 = 1$, $\sigma = 10$, and known variance), where both algorithms have exceeded the desired probability of correct determination, and HHLHL (i.e., $n_0 = 5$, $\mu_0 = (-1, -1)$, $\sigma_0 = 1$, $\sigma = 10$, and known variance), where both algorithms failed to achieve the desired probability of correct determination in the uncorrelated setting. The performance of the hyperplanes heuristic appears to be robust at both design points and insensitive to the level of correlation. The Bayesian heuristic maintains its robustness in the setting with low correlation, but suffers a slight performance degradation when the correlation is high. Overall, their performance is consistent.

The settings where the covariance matrix is unknown and must therefore be estimated from the samples are a bit more complicated. The estimator of the matrix is *asymptotically* consistent; in fact, it is well known that the covariance matrix can be estimated with relative error ϵ in $O((t/\epsilon)^2 m)$ replications with probability $1 - 2\exp(-t^2 m)$; see Vershynin [23]. Hence, in a finite-sample setting, the estimated covariance matrix may not even be positive definite, which would create a fatal flaw in the algorithm.

7.2 Comparison with the Method of Batur and Kim [3]

In this section, we compare the performance of the Bayesian and the frequentist heuristics using separating hyperplanes with the procedures proposed by Batur and Kim [3]. In particular, our heuristics are applied to the setting Kim and Batur describe in their Table 1, which is reproduced below as Table 4 for completeness, and contrasted with the experimental results they report in Table 3 of their paper, where the feasible region is the negative quadrant and the tolerance level for each constraint is given by $\epsilon = 1/\sqrt{n_0}$. Recall that the tolerance level is a parameter, which is analogous to the indifference zone parameter in classical R&S, representing the minimum difference that is worth detecting. As shown in Figure 6, the systems therefore fall into three categories: "desirable,"

			constant variance			increasing variance			decreasing variance		
			mean	median	PCD	mean	median	PCD	mean	median	PCD
Batur Kim	**D1**		**153**			**268**			**270**		
Bayes	$-\epsilon, -\epsilon$	n_0=1	23.44	2	0.8705	29.89	2	0.8831	37.55	2	0.8808
		n_0=10	25.97	10	0.8744	36.22	10	0.8743	49.74	10	0.8772
Hyperplane		n_0=1	1184.3	10.4	1	1267.5	9	1	1233.3	9.6	1
		n_0=10	1087.9	21.4	1	1159.1	11	1	1114.5	10.8	1
Batur Kim	**D2**		**101**			**117**			**219**		
Bayes	$-\epsilon, -2\epsilon$	n_0=1	11.17	2	0.9342	30.58	2	0.8813	25.45	2	0.9435
		n_0=10	37.64	10	0.8852	40.93	10	0.8792	34.27	10	0.8875
Hyperplane		n_0=1	2090.8	7.1	1	1338	7.1	1	1357.7	4.4	1
		n_0=10	1459	20.2	1	1437.6	9.6	1	1395.6	9.2	1
Batur Kim	**D3**		**21**			**38**			**38**		
Bayes	$-10\epsilon, -10\epsilon$	n_0=1	11.17	2	1	34	2	1	34.29	2	1
		n_0=10	11.14	10	0.9138	33.73	10	0.9827	33.94	10	0.9771
Hyperplane		n_0=1	1	1	1	1	1	1	1	1	1
		n_0=10	0.066	10	1	1.34	10	1	53.18	10	1
Batur Kim	**A1**		**215**			**376**			**375**		
Bayes	$-2\epsilon, -\epsilon/2$	n_0=1	8.93	2	0.9462	15.14	2	0.9472	33.97	2	0.8955
		n_0=10	31.7	10	0.8891	31.06	10	0.8932	41.49	10	0.8841
Hyperplane		n_0=1	1477.9	11.1	1	1071.6	6.8	1	1114.1	14.2	1
		n_0=10	1106.6	11.4	1	1449.9	9	1	1189	10.6	1
Batur Kim	**A3**		**28**			**44**			**44**		
Bayes	$\epsilon/2, \epsilon/2$	n_0=1	42.82	2.54	0.8639	75.01	3.21	0.8676	59.08	4.92	0.8703
		n_0=10	36.34	10	0.8747	62.19	10	0.8758	56.73	10	0.8747
Hyperplane		n_0=1	1117	14.8	1	1137	14.2	1	1176.5	13.8	1
		n_0=10	1149.6	12.5	1	1411.6	10.6	1	1415.6	10.8	1
Batur Kim	**U1**		**51**			**96**			**65**		
Bayes	$-2\epsilon, \epsilon$	n_0=1	15.41	2	0.9351	28.82	2	0.9452	32.15	2	0.8773
		n_0=10	67.77	10	0.8886	50.48	10	0.8866	37.66	10	0.8797
Hyperplane		n_0=1	1816.5	8.8	1	2426.7	6	1	1957.3	11	1
		n_0=10	1738.1	10.6	1	1161.4	9.2	1	2848.5	9.9	1
Batur Kim	**U2**		**19**			**30**			**30**		
Bayes	ϵ, ϵ	n_0=1	37.03	2	0.8685	42.32	2	0.8826	28.24	2	0.8823
		n_0=10	55.06	10	0.8779	44.12	10	0.8762	42.43	10	0.8752
Hyperplane		n_0=1	1149.4	9.4	1	1444.6	9	1	1347.1	9	1
		n_0=10	1194.3	11	1	1086.7	10.6	1	1217.4	10	1
Batur Kim	**U3**		**11**			**14**			**14**		
Bayes	$\epsilon, 2\epsilon$	n_0=1	11.91	2	0.9371	26.62	2	0.8772	18.81	2	0.9441
		n_0=10	32.53	10	0.8896	41.97	10	0.8806	30.12	10	0.8889
Hyperplane		n_0=1	1745.9	5	1	1200.2	6.6	1	1157.4	4.2	1
		n_0=10	1475.4	9.8	1	1155.1	10.8	1	1348.1	9.4	1

Fig. 6. Comparison with the Method of Batur and Kim [3].

Table 4. Parameter Settings from Table 1 of Batur and Kim [3]

	desirable
D1	$\mu_j = -\epsilon, j = 1, \ldots, m$
D2	$\mu_j = -j\epsilon, j = 1, \ldots, m$
D3	$\mu_j = -10\epsilon, j = 1, \ldots, m$
	acceptable
A1	$\mu_1 = \mu_2 = -2\epsilon, \mu_j = -\epsilon/2, j = 3, 4, \ldots, m$
A2	$\mu_j = 0, j = 1, \ldots, m$
A3	$\mu_j = \epsilon/2, j = 1, \ldots, m$
	unacceptable
U1	$\mu_1 = \mu_2 = -2\epsilon, \mu_j = \epsilon, j = 3, 4, \ldots, m$
U2	$\mu_j = \epsilon, j = 1, \ldots, m$
U3	$\mu_j = j\epsilon, j = 1, \ldots, m$

denoted by $D1$, $D2$, and $D3$; "acceptable," denoted by $A1$ and $A3$; "undesirable," denoted by $U1$, $U2$, and $U3$, designating feasible, infeasible but acceptable, and infeasible systems, respectively.

We note that this is not an "apples-to-apples" comparison, as the performance metrics used in the two papers are slightly different. In particular, Batur and Kim [3] define a *correct decision* as the event whereby the set of feasible systems reported by their algorithms not only contains all desirable (truly feasible) systems, but may also contain some of the acceptable ones with the probability of a correct decision at or above the nominal confidence level. In our Bayesian setting, we try to ensure that the average probability of correct classification over all systems achieves the nominal confidence level. This means that while some systems may be correctly classified at a much higher confidence level than the one specified by the analyst, others may be below the desired confidence level as long as we achieve, in expectation, the confidence level targeted by the analyst. Our frequentist heuristic using hyperplanes, however, tries to classify each system with the desired level of confidence. To illustrate with an example, imagine that we are trying to identify investment opportunities whose payoff exceeds a certain threshold value. The algorithms of Batur and Kim [3] would report, with the desired level of confidence, a portfolio that includes all investment opportunities that exceed the threshold value (i.e., desirable ones) as well as some opportunities that fall into the indifference zone (i.e., acceptable ones). In our Bayesian heuristic, however, the reported portfolio would achieve, in expectation, the nominal probability of correct classification with some investment opportunities clearing the threshold value with a much higher probability while others at a probability of correct classification that is lower than the desired level. Finally, the frequentist heuristic will report a portfolio that contains only the desirable investment opportunities that exceed the threshold value at the postulated level of confidence. However, as they present one of the best feasibility determination approaches in the literature, Batur and Kim [3] provide a natural benchmark.

Even though we had developed our algorithms under the assumption of known variances, in our experiments, we have estimated the variances based on the initial sample. As expected, for the Bayesian setting, violating this assumption resulted in an increase in the average total sample size as well as a slight degradation in the performance of the algorithm. The frequentist heuristic, however, has been quite robust in achieving perfect probability of correct classification; however, this performance is obtained with significantly higher average sample sizes. Given that Batur and Kim [3] do not report variances in their results, we cannot directly compare the volatility in the

performance of these procedures. Note that all of the algorithms had to spend considerably more effort in correctly classifying a system inside the feasible region.

As is the case with other multi-stage algorithms that rely on Bayesian updating, if the analyst does not have an accurate estimate of the prior values of the system parameters, it is advisable to initialize the procedure with a small initial sample size, say, $n_0 = 1$. An inaccurate prior value coupled with a large initial sample size, say, $n_0 = 10$, anchors the procedure on the wrong point, necessitating a larger number of additional samples for correct classification.

8 CONCLUSIONS AND FUTURE RESEARCH

This work was concerned with determining whether a number of systems—each characterized by the same number of random variables—belongs to a set Γ defined by a finite collection of linear inequalities. Specifically, a system is in (not in) Γ if the mean of its random variables is in (not in) Γ. Unfortunately, the means of the systems are unknown; they must therefore be estimated using Monte Carlo simulation. Various heuristics have been proposed for classifying a number of the systems with a user-specified level of confidence, $1 - \beta$, with the minimum number of simulation replications. The heuristics are based on a Bayesian approach in which the user specifies initial estimates of the means and standard deviations of the random variables in each system. After performing some initial simulation replications, the current estimates of the means and standard deviations are used to determine which systems, if any, can be classified at the user's level of confidence. For each of the remaining unclassified systems, heuristics are used to estimate the number of additional simulation replications to perform to hopefully classify the system at the desired level of confidence. The way this is done depends on whether the current estimate of the mean of a system is, or is not, in Γ. Computational experiments with the algorithms indicate that they are easy to implement and efficient with respect to the average number of replications needed to achieve the desired performance level.

While the algorithms were originally developed under the assumptions that system variance is known and the random variables are independent, their performance remains satisfactory when those assumptions are relaxed *individually*. It would be interesting to relax the two assumptions *simultaneously,* as we may run into issues of invetibility with the estimated covariance matrix. This is the subject of current investigation.

Another possible measure of performance is to classify a user-specific fraction of the r systems, each with a user-specified minimum probability of correct classification. This is the focus of our current work. Additionally, we are investigating the setting where the set Γ is defined by a finite set of differentiable nonlinear inequalities.

APPENDIX

Appendix A: Finding a Zero of a Convex Function

In the algorithms developed here, it is sometimes necessary to solve a slightly modified version of the following zero-finding problem:

Problem 1: Given a convex function $h : R^1 \to R^1$, find $x^* \in R^1$ such that $h(x^*) = 0$.

Solow and Li [18] developed a subgradient Newton method to solve Problem 1 that has some very desirable properties as a result of h being convex. In particular, starting anywhere, the algorithm either terminates finitely—in which case, a zero of h is found or else there is no zero of h. Alternatively, if the algorithm generates an infinite sequence of points, then that sequence either converges to a zero of h or diverges to $\pm\infty$, in which case, h has no zero.

Turning to the application at hand, it is sometimes necessary to solve the following variation of Problem 1:

Problem 2: Given a convex function $h : [0, \infty) \rightarrow R^1$, find a point $x^* \in [0, \infty)$ such that $h(x^*) = 0$.

The next proposition provides conditions under which the algorithm in Reference [18] will find such an x^* (see Appendix B for the proof).

Proposition 1. *If $h : [0, \infty) \rightarrow R^1$ is a convex function on $[0, \infty)$ that has a zero, then the algorithm in Equation [18] will find such a point provided that the algorithm is started at a point $x \in [0, \infty)$ for which $h(x) \geq 0$.*

Appendix B: Proofs of Propositions

This Appendix contains proofs for all of the propositions.

Proof of Proposition 3.1. For an integer $\tilde{n} > 0$ it is first shown that if $t \leq \hat{\tau}(\tilde{n})$, then $R_t(\tilde{n}) \subset \Gamma$, in which for each $k = 1, \ldots, c$:

$$\hat{\tau}_k(\tilde{n}) = \frac{b_k - \boldsymbol{a}_k^T \boldsymbol{\mu}_{last}}{\boldsymbol{a}_k^T \boldsymbol{d}}, \quad \text{where } d_j = \begin{cases} +\sigma_j(\tilde{n}) & \text{if } a_{kj} \geq 0, \\ -\sigma_j(\tilde{n}) & \text{if } a_{kj} < 0, \end{cases} \tag{27}$$

and

$$\hat{\tau}(\tilde{n}) = \min\{\hat{\tau}_k(\tilde{n}) : k = 1, \ldots, c\}.$$

For any constraint k of Γ, consider the following feasible and bounded linear programming problem and optimal solution:

$$\begin{array}{ll} \max & \boldsymbol{a}_k^T \boldsymbol{y} \\ \text{s.t.} & \boldsymbol{y} \in R(\boldsymbol{\mu}_{last}, \boldsymbol{\sigma}(\tilde{n}), \hat{\tau}_k(\tilde{n})) \end{array} \quad y_j^* = \begin{cases} \boldsymbol{\mu}_{last} + \hat{\tau}_k(\tilde{n})\sigma_j(\tilde{n}) & \text{if } a_{kj} \geq 0, \\ \boldsymbol{\mu}_{last} - \hat{\tau}_k(\tilde{n})\sigma_j(\tilde{n}) & \text{if } a_{kj} < 0. \end{cases} \tag{28}$$

From the definition of $\boldsymbol{d}$ and $\hat{\tau}_k(\tilde{n})$ in Equation (27), it follows that $\boldsymbol{a}_k^T \boldsymbol{y}^* = \boldsymbol{a}_k^T(\boldsymbol{\mu}_{last} + \hat{\tau}_k(\tilde{n})\boldsymbol{d}) = b_k$.

To show that $R_t(\tilde{n}) \subset \Gamma$, let $\boldsymbol{y} \in R_t(\tilde{n}) = R(\boldsymbol{\mu}_{last}, \boldsymbol{\sigma}(\tilde{n}), t) \subset R(\boldsymbol{\mu}_{last}, \boldsymbol{\sigma}(\tilde{n}), \hat{\tau}_k(\tilde{n}))$ (because $t \leq \hat{\tau}_k(\tilde{n})$), and so $\boldsymbol{y}$ is feasible for the LP in Equation (28). As such, $\boldsymbol{a}_k^T \boldsymbol{y} \leq \boldsymbol{a}_k^T \boldsymbol{y}^* = b_k$. Thus, $\boldsymbol{y}$ is feasible for each constraint k of Γ and so $R_t(\tilde{n}) \subset \Gamma$.

It remains to show that if $t > \hat{\tau}(\tilde{n})$, then $R_t(\tilde{n}) \not\subset \Gamma$. As $t > \hat{\tau}(\tilde{n})$, there is a constraint k of Γ for which $t > \hat{\tau}_k(\tilde{n})$ and consider the following feasible and bounded linear programming problem and optimal solution:

$$\begin{array}{ll} \max & \boldsymbol{a}_k^T \boldsymbol{y} \\ \text{s.t.} & \boldsymbol{y} \in R_t(\tilde{n}) \end{array} \quad y_j^* = \begin{cases} \boldsymbol{\mu}_{last} + t\sigma_j(\tilde{n}) & \text{if } a_{kj} \geq 0, \\ \boldsymbol{\mu}_{last} - t\sigma_j(\tilde{n}) & \text{if } a_{kj} < 0. \end{cases} \tag{29}$$

But then the point $\boldsymbol{y}^* \in R_t(\tilde{n})$ while, from the definition of $\boldsymbol{d}$ and $\hat{\tau}_k(\tilde{n})$ in Equation (27), it follows that

$$\boldsymbol{a}_k^T \boldsymbol{y}^* = \boldsymbol{a}_k^T(\boldsymbol{\mu}_{last} + t\boldsymbol{d}) > \boldsymbol{a}_k^T(\boldsymbol{\mu}_{last} + \hat{\tau}_k(\tilde{n})\boldsymbol{d}) = b_k.$$

This means that $\boldsymbol{y}^* \in R_t(\tilde{n})$ but $\boldsymbol{y}^* \notin \Gamma$ and so the proof is now complete.

Proof of Proposition 1 in Appendix A. Suppose Solow's subgradient algorithm is started at a point $x^0 \in [0, \infty)$ for which $h(x^0) > 0$ and assume that h has a zero. The result is proved for the case when h is differentiable on $[0, \infty)$. Suppose first that $h'(x^0) < 0$. In this case, the algorithm generates a monotonically increasing sequence of points, so each $x^k \in [0, \infty)$. If for any k, $h'(x^k) \geq 0$ then Reference [18] showed that the function h has no zero, which contradicts the assumption that h has a zero. Thus, if the sequence generated by the algorithm is finite, then, as shown in Reference [18], the point at which the algorithm stops is a zero of h. However, if the sequence generated by the algorithm is infinite, then the monotonically increasing sequence must be bounded above for, if not, as shown in Reference [18], the function h has no zero. As such, the monotonically

increasing and bounded sequence generated by the algorithm must converge to a point x^* and, again, as shown in Reference [18], $h(x^*) = 0$.

Turning now to the case when $h'(x^0) > 0$, as shown in Reference [18], the sequence of points generated by the algorithm is monotonically decreasing with the function value at each such point x^k being strictly positive. Furthermore, $h'(x^k) > 0$ for, if not, then, as shown in Reference [18], the function h has no zero, which contradicts the assumption that h has a zero.

If, for some first value of k, $x^{k+1} < 0$ and hence outside the domain of h, then the function has no zero, as is now shown. From the gradient inequality for a convex function, it follows that for all $x > x^k$,

$$h(x) \geq h(x^k) + h'(x^k)(x - x^k) > h(x^k) > 0.$$

Thus, there is no value of $x > x^k$ for which $h(x) = 0$. It remains to show that there is no value of $x \in [0, x^k)$ with $h(x) = 0$, so suppose there is such a value for x. As $x^{k+1} = x^k - h(x^k)/h'(x^k) < 0$ and $h'(x^k) > 0$, it follows that $h(x^k) - h'(x^k)x^k > 0$. From the gradient inequality for a convex function, it follows that

$$h(x) \geq h(x^k) + h'(x^k)(x - x^k) = h(x^k) - h'(x^k)x^k + h'(x^k)x > h'(x^k)x \geq 0.$$

This means that, if $h'(x^0) > 0$, then the sequence of points generated by the algorithm is within the domain of h and is monotonically decreasing and bounded below by 0. Thus, if the sequence generated by the algorithm is finite, then, as shown in Reference [18], the point at which the algorithm stops is a zero of h. However, if the sequence generated by the algorithm is infinite, then the monotonically decreasing sequence is bounded below by 0 and must converge to a point x^*, which, as shown in Reference [18], satisfies $h(x^*) = 0$. This completes the proof.

ACKNOWLEDGMENTS

The authors wish to thank the Associate Editor and the three referees for their detailed guidance that led to a much improved manuscript.

REFERENCES

[1] S. Andradottir, D. Goldsman, and S.-H. Kim. 2005. Finding the best in the presence of stochastic constraints. In *Proceedings of the Winter Simulation Conference*, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines (Eds.). Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 732–738.

[2] S. Andradottir and S. H. Kim. 2010. Fully sequential procedures for comparing constrained systems via simulation. *Nav. Res. Log.* 57 (2010), 403–421.

[3] D. Batur and S. H. Kim. 2010. Finding feasible systems in the presence of constraints on multiple performance measures. *ACM Trans. Model. Comput. Simul.* 20 (2010), 118–130.

[4] H. C. Chen, C. H. Chen, and E. Yücesan. 2000. Computing efforts allocation for ordinal optimization and discrete event simulation. *IEEE Trans. Automat. Control* 45 (2000), 960–964.

[5] S. Chick and K. Inoue. 2001. New two-stage and sequential procedures for selecting the best simulated system. *Oper. Res.* 49 (2001), 1609–1624.

[6] S. E. Chick, J. Branke, and C. Schmidt. 2010. Sequential sampling to myopically maximize the expected value of information. *INFORMS J. Comput.* 22 (2010), 71–80.

[7] A. Dembo and O. Zeitouni. 1998. Large deviations techniques and applications. In *Applications of Mathematics, Stochastic Modeling and Applied Probabaility*. Volume 38. Springer-Verlag, New York.

[8] P. I. Frazier, W. B. Powell, and S. Dayanik. 2008. A knowledge-gradient policy for sequential information collection. *SIAM J. Contr. Optim.* 47 (2008), 2410–2439.

[9] S. Gao and W. Chen. 2017. Efficient feasibility determination with multiple performance measure constraints. *IEEE Trans. Automat. Control* 62 (2017), 113–122.

[10] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. 2013. *Bayesian Data Analysis* (3rd ed.). CRC Press, Boca Raton, FL.

[11] S. R. Hunter and B. McClosky. 2016. Maximizing quantitative traits in the mating design problem via simulation-based Pareto estimation. *IISE Trans.* 48 (2016), 565–578.

[12] S. R. Hunter and R. Pasupathy. 2013. Optimal sampling laws for stochastically constrained simulation optimization on finite sets. *INFORMS J. Comput.* 25, 3 (2013), 527–542.
[13] L. H. Lee, N. A. Pujowidianto, C. H. Chen, and C. M. Yap. 2012. Asymptotic simulation budget allocation for selecting the best design in the presence of stochastic constraints. *IEEE Trans. Automat. Control* 57 (2012), 2940–2945.
[14] B. Libai, E. Muller, and R. Peres. 2005. The role of seeding in multi-market entry. *J. Res. Market.* 22 (2005), 375–393.
[15] I. O. Ryzhov. 2016. On the convergence rates of expected improvement methods. *Oper. Res.* 64 (2016), 1515–1528.
[16] T. J. Santner and A. C. Tamhane. 1984. Designing experiments for selecting a normal population with a large mean and a small variance. In *Design of Experiments – Ranking and Selection: Essays in Honor of Robert E. Bechhoffer*, T. J. Santner and A. C. Tamhane (Eds.). Marcel-Dekker, New York, 179–198.
[17] D. I. Singham and R. Szechtman. 2016. Multiple comparisons with a standard using false discovery rates. In *Proceedings of the Winter Simulation Conference*, T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick (Eds.). Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 501–511.
[18] D. Solow and H. Li. 2007. A constrained optimization approach to solving certain systems of convex equations. *Eur. J. Oper. Res.* 176 (2007), 1334–1347.
[19] R. Szechtman and E. Yücesan. 2008. A new perspective on feasibility determination. In *Proceedings of the Winter Simulation Conference*, S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler (Eds.). Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 273–280.
[20] R. Szechtman and E. Yücesan. 2016. A Bayesian approach to feasibility determination. In *Proceedings of the Winter Simulation Conference*, T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick (Eds.). Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 782–790.
[21] J. Telgen. 1983. Identifying redundant constraints and implicit equalities in systems of linear constraints. *Manag. Sci.* 29, 10 (1983), 1209–1222.
[22] J. F. Traub and A. G. Verschultz. 2000. *Complexity and Information* (3rd ed.). Cambridge University Press, Cambridge, UK.
[23] R. Vershynin. 2010. Introduction to the non-asymptotic analysis of random matrices. *Arxiv Preprint Arxiv:1011.3027* (2010).
[24] J. Xie and P. I. Frazier. 2013. Sequential Bayes-optimal policies for multiple comparisons with a known standard. *Oper. Res.* 61 (2013), 1174–1189.

Received August 2018; revised June 2020; accepted July 2020