Efficient Computing Budget Allocation for Finding Simplest Good Designs

Qing-Shan Jia, Enlu Zhou, Chun-Hung Chen

Abstract

In many applications some designs are easier to implement, require less training data and shorter training time, and consume less storage than the others. Such designs are called simple designs, and are usually preferred over complex ones when they all have good performance. Despite the abundant existing studies on how to find good designs in simulation-based optimization (SBO), there exist few studies on finding simplest good designs. We consider this important problem in this paper, and make the following contributions. First, we provide lower bounds for the probabilities of correctly selecting the m simplest designs with top performance, and selecting the best m such simplest good designs, respectively. Second, we develop two efficient computing budget allocation methods to find m simplest good designs and to find the best m such designs, respectively; and show their asymptotic optimalities. Third, we compare the performance of the two methods with equal allocations over 6 academic examples and a smoke detection problem in wireless sensor networks. We hope that this work brings insight to finding the simplest good designs in general.

This work has been supported in part by National Natural Science Foundation of China under grant (Nos. 60704008, 60736027, 61174072, and 90924001), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20070003110), the Tsinghua National Laboratory for Information Science and Technology (TNLIST) Cross-Discipline Foundation, the National 111 International Collaboration Project (No. B06002), National Science Foundation under Grants ECCS-0901543 and CMMI-1130273, the Air Force Office of Scientific Research under Grant FA-9550-12-1-0250, Department of Energy under Award DE-SC0002223, NIH under Grant 1R21DK088368-01, and National Science Council of Taiwan under Award NSC-100-2218-E-002-027-MY3.

Qing-Shan Jia is with Center for Intelligent and Networked Systems (CFINS), Department of Automation, TNLIST, Tsinghua University, Beijing 100084, China (Tel.: +86-10-62773006, fax: +86-10-62796115, email: jiaqs@tsinghua.edu.cn).

Enlu Zhou is with Department of Industrial & Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, IL, 61801, USA (enluzhou@illinois.edu).

Chun-Hung Chen is with Department of Electrical Engineering and Institute of Industrial Engineering, National Taiwan University, Taipei, Taiwan (cchen9@cc.ee.ntu.edu.tw).

Keywords: **Simulation-based optimization, complexity preference, optimal computing budget allocation, wireless sensor network.**

I. Introduction

Many systems nowadays follow not only physical laws but also man-made rules. These systems are called discrete-event dynamic systems (DEDS's) and the optimization of their performance enters the realm of simulation-based optimization (SBO). In many applications some designs are easier to implement, require less training data and shorter training time, and consume less storage than the others. Such designs are called simple designs, and are usually preferred over complex ones when they all have good performance. For example, a wireless sensor network can be used to detect smoke. While a larger sensing radius allows to detect the smoke faster, it also increases the power consumption and shortens the lifetime of the network. Thus, when the detection is fast enough, a small sensing radius is preferred over a large one.

There are abundant existing studies on SBO. Ranking and selection (R&S) procedures are well-known procedures to combine simulation and optimization to improve the efficiency. The origins can be traced back to two papers, namely Bechhofer [1] for the indifference-zone formulation and Gupta [2], [3] for the subset selection formulation. Excellent surveys on R&S can be found in [4]–[6]. Chen et al. [7] and Chen and Yücesan [8] developed the optimal computing budget allocation (OCBA) procedure to asymptotically maximize a lower bound of the probability of correctly selecting the best solution candidate. OCBA was later extended to select the best several solution candidates [9], to handle stochastic constraints [10], multiple objective functions [11], correlated observation noises [12], and opportunity cost [13]. A comprehensive introduction to OCBA is recently available in [14]. Recent surveys on other methods for SBO can be found in [15]–[19].

Despite the abundance of existing studies on finding good designs in SBO, there exist few studies on finding simplest good designs. This problem is challenging due to the following difficulties. First, simulation-based performance evaluation is usually time-consuming and provides only noisy performance estimation. The second difficulty is randomness. Due to the pervasive randomness in the system dynamics, usually multiple runs are required in order to obtain accurate estimation. The last difficulty is complexity preference for simple designs which requires one to consider both the performance and the complexity simultaneously. Existing studies on SBO usually handle the first two difficulties well, but very few consider the complexity preference, which is the main contribution of this paper.

The preference for simple designs has been mathematically described by the Kolmogorov complexity [20], the Solomonoff's universal prior [21], [22], the Levin complexity [23], and the AIXI model [24], just to name a few. Most of these formulations assume that the performance of designs can be easily evaluated, and thus do not address the unique difficulties of simulation-based performance evaluations. The study on SBO with complexity preference has only been recently considered. Jia and Zhao [25] studied the relationship between performance and complexity of different policies in Markov decision processes (MDPs). Jia [26] showed that it is possible to obtain simple policies with good performance in MDPs. Jia [27] and Jia [28] considered SBO with descriptive complexity preference and developed an adaptive sampling algorithm to find the simplest design with bounded cardinal performance. Later on, Yan et al. [29], [30] developed two algorithms (OCBAmSG and OCBAbSG) to find m simplest designs with bounded cardinal performance and to find the best m such designs, respectively. The above methods suit the applications when there are clear bounds on the cardinal performance of designs. However, in many applications it is difficult to estimate the performance of the best design a priori, which makes it difficult to identify "good" designs in a cardinal sense. In [31], Ho et al. showed that the probability for correctly identifying the relative order among two designs converges to 1 exponentially fast with respect to (w.r.t.) the number of observations that are taken for each design. Note that the standard deviation of cardinal performance estimation using Monte Carlo simulation only converges in the rate of $1/\sqrt{n}$, where n is the number of observations. So in comparison, one finds that the ordinal values converge much faster than the cardinal ones. Since in many applications we want to find simple designs with top performance, we focus this paper on finding simplest good designs in the ordinal sense.

In this paper, we consider the important problem of how to allocate the computing budget so that the simplest good designs can be found with high probability, and make the following major contributions. First, we mathematically formulate two related problems. One is how to find m simplest designs that have top-g performance. When $g > m$ there could be multiple choices for m such designs. For example, suppose θ_1, θ_2, and θ_3 are the best, the second best, and the third best designs, respectively. And suppose their complexities are the same. When $g = 3$ and $m = 2$, there are three choices for m simplest designs with top-g performance, namely $\{\theta_1, \theta_2\}, \{\theta_1, \theta_3\}$, and $\{\theta_2, \theta_3\}$. Clearly, the choice of $\{\theta_1, \theta_2\}$ has the best performance. So another related problem is how to find m simplest top-g designs that have the best performance among all the choices. We develop lower bounds for the probabilities of correctly selecting the two subsets of m simplest good designs, which are denoted as PCSm and PCSb, respectively. Second, we develop efficient computing budget allocation methods to asymptotically optimize the two PCS's, respectively. The two methods are called optimal computing budget allocation for m simplest good designs in the ordinal sense (OCBAmSGO) and optimal computing budget allocation for the best m simplest good designs in the ordinal sense (OCBAbSGO), respectively. Then we numerically compare their performance with equal allocation on academic examples and a smoke detection problem in wireless sensor networks (WSNs).

The rest of this paper is organized as follows. We mathematically formulate the two problems in section II, present the main results in section III, show the experimental results in section IV, and briefly conclude in section V.

II. Problem Formulation

In this section we define the m simplest good designs (or mSG for short) and the best m simplest good designs (or bSG for short) using the true performance of the designs in subsection II-A, and define the probabilities of correct selection based on Bayesian model in subsection II-B.

A. Definitions of mSG and bSG

Consider a search space of k competing designs $\Theta = \{\theta_1, \ldots, \theta_k\}$. Let $J(\theta)$ be the performance of θ, which can be accurately evaluated only through an infinite number of replications, i.e.,

$$J(\theta) \overset{a.s.}{=\!=} \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \hat{J}(\theta, \xi_i), \tag{1}$$

where

$$\hat{J}(\theta, \xi_i) = J(\theta) + w(\theta, \xi_i) \tag{2}$$

is the observation, $w(\theta, \xi_i)$ is the observation noise which has independent Gaussian distribution $N(0, \sigma^2(\theta))$, and ξ_i represents the randomness in the i-th sample path. Suppose we are considering minimization problems. Sort all the designs from the smallest to the largest according to $J(\theta)$.

Let G be the set of the top-g ($g < k$) designs, i.e., $G = \{\theta_{i_1}, \ldots, \theta_{i_g}\}$, where θ_{i_j} is the top j-th design. Let $C(\theta)$ be the complexity of design θ. Without loss of generality, we assume M different complexities, i.e., $\{C(\theta), \theta \in \Theta\} = \{1, \ldots, M\}$. Let Θ_i denote designs with complexity i, i.e., $\Theta_i = \{\theta \in \Theta, C(\theta) = i\}$. Then each Θ_i can be divided into two subsets, namely $G_i = \Theta_i \cap G$ that contains all the good designs with complexity i and $D_i = \Theta_i \setminus G_i$ that contains all the rest of the designs with complexity i, where $\setminus$ represents set minus. We then have $\bigcup_{i=1}^{M} G_i = G$ and $D = \Theta \backslash G$. It is possible that $G_i = \varnothing$ or $D_i = \varnothing$ for some i. Throughout this paper, we assume

Assumption 1: The complexities of all the designs are known, i.e., $C(\theta)$ is known for all $\theta \in \Theta$.

Assumption 2: When selecting among good designs, complexity has priority over performance.

A set of designs S is called the m ($m < g$) simplest good designs (or mSG for short) if all the following conditions are satisfied.

(1) $|S| = m$,

(2) $S \subset G$,

(3) $\max_{\theta \in S} C(\theta) \leq \min_{\theta \in G \backslash S} C(\theta)$.

Furthermore, a set of designs S is called the best mSG (or bSG for short) if all the following conditions are satisfied.

(1) S is mSG,

(2) if there exists $\theta \in S$ and $\theta' \in G \setminus S$ s.t. $C(\theta) = C(\theta')$, then $J(\theta) \leq J(\theta')$.

Note that the first condition above clearly shows that a bSG must be an mSG. The second condition shows that a bSG has the best performance among all mSG's.

B. Definitions of Probabilities of Correct Selection

Note that in the above definitions of G, D, mSG, and bSG the true performance $J(\theta)$ is used, which can be obtained only when an infinite number of replications are used. In practice only a finite number of replications are affordable, i.e.,

$$\bar{J}(\theta) = \frac{1}{n(\theta)} \sum_{i=1}^{n(\theta)} \hat{J}(\theta, \xi_i). \tag{3}$$

We follow the Bayesian model as in [32], [7], and [9]. The mean of the simulation output for each design, $J(\theta)$, is assumed unknown and treated as a random variable. After the simulation is performed, a posterior distribution for the unknown mean $J(\theta)$, $p(J(\theta)|\hat{J}(\theta, \xi_i), i = 1, \ldots, n(\theta))$, is constructed based on two pieces of information: (i) prior knowledge of the system's performance, and (ii) current simulation output. As in [32], we assume that the unknown mean $J(\theta)$ has a conjugate normal prior distribution and consider noninformative prior distributions, which implies that no prior knowledge is available about the performance of any design before conducting the simulations, in which case the posterior distribution of $J(\theta)$ is (c.f. [33])

$$\tilde{J}(\theta) \sim N(\bar{J}(\theta), \sigma^2(\theta)/n(\theta)). \tag{4}$$

Given $\bar{J}$, then G and D are both random sets. Thus, we define the probability of correctly selecting an mSG as

$$\begin{aligned} PCSm &\equiv \Pr\{S \text{ is mSG}\} \\ &= \Pr\left\{|S| = m, S \subset G, \max_{\theta\in S} C(\theta) \le \min_{\theta'\in G\setminus S} C(\theta')\right\}, \end{aligned} \tag{5}$$

and define the probability of correctly selecting a bSG as

$$\begin{aligned} PCSb &\equiv \Pr\{S \text{ is bSG}\} \\ &= \Pr\left\{|S| = m, S \subset G, \max_{\theta\in S} C(\theta) \le \min_{\theta'\in G\setminus S} C(\theta'), \max_{\theta\in\Theta_{\max_{\theta'\in S} C(\theta')}\cap S} \tilde{J}(\theta) \le \min_{\theta''\in\Theta_{\max_{\theta'\in S} C(\theta')}\setminus S} \tilde{J}(\theta'')\right\}. \end{aligned} \tag{6}$$

Now we can mathematically formulate the following two problems.

(P1) $\max_{n(\theta_1),\ldots,n(\theta_k)} PCSm$ s.t. $\sum_{i=1}^{k} n(\theta_i) = T$,

(P2) $\max_{n(\theta_1),\ldots,n(\theta_k)} PCSb$ s.t. $\sum_{i=1}^{k} n(\theta_i) = T$,

where $n(\theta_i)$ is the number of replications allocated to design θ_i. We will provide methods to address the above two problems in the next section.

III. Main Results

In this section we address problems P1 and P2 in subsections III-A and III-B, respectively.

A. Selecting an mSG

Given the observed performance of all the designs, we can divide the entire design space Θ into at most $2M$ subsets, namely $\bar{G}_1, \ldots, \bar{G}_M$ and $\bar{D}_1, \ldots, \bar{D}_M$, where $\bar{G}$ and $\bar{D}$ represent the observed top-g designs and the rest designs, respectively; $\bar{G}_i = \Theta_i \cap \bar{G}$, and $\bar{D}_i = \Theta_i \setminus \bar{G}_i$. Though there are multiple choices of an mSG, we are specifically interested in the following procedure to select an mSG. Initially, set $S = \varnothing$. We start from $\bar{G}_1$ and add designs from $\bar{G}_1$ to S from the smallest to the largest according to their observed performance $\bar{J}(\theta)$. When all the designs in $\bar{G}_i$ have been added to S and $|S| < m$, we move on to $\bar{G}_{i+1}$ and continue the above procedure until $|S| = m$. Suppose t satisfies that $\sum_{i=1}^{t} |\bar{G}_i| < m \leq \sum_{i=1}^{t+1} |\bar{G}_i|$. Let $\theta_{i,j}$ be the observed jth best design in Θ_i. Then we have

$$S = \left(\bigcup_{i=1}^{t} \bar{G}_i \right) \cup \left\{ \theta_{t+1,1}, \ldots, \theta_{t+1,m-\sum_{i=1}^{t} |\bar{G}_i|} \right\}. \tag{7}$$

This specific choice of S contains the best simplest m designs in G. It is not only an estimate of mSG under the given $\bar{J}(\theta)$'s but also is an estimate of bSG. This latter point will be explored later in subsection III-B.

Note that complexity has been considered in the choice of S in Eq. (7). We have $|S| = m$, $S \subset \bar{G}$, and $\max_{\theta \in S} C(\theta) \leq \min_{\theta' \in \bar{G} \setminus S} C(\theta')$. Then following Eq. (5) we have

$$\begin{aligned}
PCSm &= \Pr\{S \text{ is mSG}\} \\
&= \Pr\left\{ |S| = m, S \subset G, \max_{\theta \in S} C(\theta) \leq \min_{\theta' \subset G \setminus S} C(\theta') \right\} \\
&= \Pr\left\{ S \subset G, \max_{\theta \in S} C(\theta) \leq \min_{\theta' \in G \setminus S} C(\theta'), G = \bar{G} \right\} \\
&\quad + \Pr\left\{ S \subset G, \max_{\theta \in S} C(\theta) \leq \min_{\theta' \in G \setminus S} C(\theta'), G \neq \bar{G} \right\} \\
&\geq \Pr\left\{ S \subset G, \max_{\theta \in S} C(\theta) \leq \min_{\theta' \in G \setminus S} C(\theta') | G = \bar{G} \right\} \Pr\left\{ G = \bar{G} \right\} \\
&= \Pr\left\{ G = \bar{G} \right\} \\
&= \Pr\left\{ \tilde{J}(\theta) < \tilde{J}(\theta') \text{ for all } \theta \in \bar{G} \text{ and } \theta' \in \bar{D} \right\} \\
&\geq \Pr\left\{ \tilde{J}(\theta) \leq \mu_1 \text{ and } \tilde{J}(\theta') > \mu_1 \text{ for all } \theta \in \bar{G} \text{ and } \theta' \in \bar{D} \right\},
\end{aligned} \tag{8}$$

where μ_1 is a given constant. Due to the independence between $\tilde{J}(\theta)$'s, we have

$$\begin{aligned}
&\Pr\left\{ \tilde{J}(\theta) \leq \mu_1 \text{ and } \tilde{J}(\theta') > \mu_1 \text{ for all } \theta \in \bar{G} \text{ and } \theta' \in \bar{D} \right\} \\
= \; &\Pr\left\{ \tilde{J}(\theta) \leq \mu_1 \text{ for all } \theta \in \bar{G} \right\} \Pr\left\{ \tilde{J}(\theta) > \mu_1 \text{ for all } \theta' \in \bar{D} \right\} \\
= \; &\left(\Pi_{\theta \in \bar{G}} \Pr\left\{ \tilde{J}(\theta) \leq \mu_1 \right\} \right) \left(\Pi_{\theta' \in \bar{D}} \Pr\left\{ \tilde{J}(\theta') > \mu_1 \right\} \right).
\end{aligned} \tag{9}$$

Following Eqs. (8) and (9), we have the following approximate PCSm (APCSm),

$$APCSm \equiv \left(\Pi_{\theta \in \bar{G}} \Pr\left\{ \tilde{J}(\theta) \leq \mu_1 \right\} \right) \left(\Pi_{\theta \in \bar{D}} \Pr\left\{ \tilde{J}(\theta) > \mu_1 \right\} \right). \tag{10}$$

Then an approximate version of P1 is

(AP1) $\max_{n(\theta_1),\ldots,n(\theta_k)} APCSm$ s.t. $\sum_{i=1}^{k} n(\theta_i) = T$.

Note that the idea of APCSm is that PCSm is lower bounded by the probability that all observed good enough designs are truly good enough, which can be asymptotically maximized if we follow the allocation procedure in [9]. This explains why the following Theorem 1 leads to a similar allocation procedure as in [9]. We briefly provide the analysis as follows.

Let L_m be the Lagrangian relaxation of AP1,

$$\begin{aligned} L_m &\equiv APCSm + \lambda_m \left(\sum_{i=1}^{k} n(\theta_i) - T \right) \\ &= \left(\Pi_{\theta \in \bar{G}} \Pr\left\{ \tilde{J}(\theta) \le \mu_1 \right\} \right) \left(\Pi_{\theta \in \bar{D}} \Pr\left\{ \tilde{J}(\theta) > \mu_1 \right\} \right) + \lambda_m \left(\sum_{i=1}^{k} n(\theta_i) - T \right). \end{aligned} \tag{11}$$

The Karush-Kuhn-Tucker (KKT) conditions (c.f. [34]) of AP1 are as follows.

For $\theta \in \bar{G}$,

$$\frac{\partial L_m}{\partial n(\theta)} = \left(\Pi_{\theta' \in \bar{G}, \theta' \neq \theta} \Pr\left\{ \tilde{J}(\theta') \le \mu_1 \right\} \right) \left(\Pi_{\theta' \in \bar{D}} \Pr\left\{ \tilde{J}(\theta') > \mu_1 \right\} \right) \phi\left(\frac{\mu_1 - \bar{J}(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}} \right) \frac{\mu_1 - \bar{J}(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}} + \lambda_m = 0; \tag{12}$$

for $\theta \in \bar{D}$,

$$\frac{\partial L_m}{\partial n(\theta)} = \left(\Pi_{\theta' \in \bar{G}} \Pr\left\{ \tilde{J}(\theta') \le \mu_1 \right\} \right) \left(\Pi_{\theta' \in \bar{D}, \theta' \neq \theta} \Pr\left\{ \tilde{J}(\theta') > \mu_1 \right\} \right) \phi\left(\frac{\bar{J}(\theta) - \mu_1}{\sigma(\theta)/\sqrt{n(\theta)}} \right) \frac{\bar{J}(\theta) - \mu_1}{2\sigma(\theta)\sqrt{n(\theta)}} + \lambda_m = 0; \tag{13}$$

and

$$\frac{\partial L_m}{\partial \lambda_m} = \sum_{i=1}^{k} n(\theta_i) - T = 0, \tag{14}$$

where $\phi(\cdot)$ is the probability density function of the standard Gaussian distribution. We now analyze the relationship between $n(\theta)$ and $n(\theta')$. First, we consider the case that θ and $\theta' \in \bar{G}$ and $\theta \neq \theta'$. Following Eq. (12) we have

$$\begin{aligned} &\left(\Pi_{\theta'' \in \bar{G}, \theta'' \neq \theta} \Pr\left\{ \tilde{J}(\theta'') \le \mu_1 \right\} \right) \left(\Pi_{\theta'' \in \bar{D}} \Pr\left\{ \tilde{J}(\theta'') > \mu_1 \right\} \right) \phi\left(-\frac{\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}} \right) \frac{-\delta_1(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}} \\ = &\left(\Pi_{\theta'' \in \bar{G}, \theta'' \neq \theta'} \Pr\left\{ \tilde{J}(\theta'') \le \mu_1 \right\} \right) \left(\Pi_{\theta'' \in \bar{D}} \Pr\left\{ \tilde{J}(\theta'' > \mu_1) \right\} \right) \phi\left(-\frac{\delta_1(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}} \right) \frac{-\delta_1(\theta')}{2\sigma(\theta')\sqrt{n(\theta')}}, \end{aligned} \tag{15}$$

where $\delta_1(\theta) \equiv \bar{J}(\theta) - \mu_1$ for all $\theta \in \Theta$. Simplifying Eq. (15), then we have

$$\Pr\left\{ \tilde{J}(\theta') \le \mu_1 \right\} \phi\left(-\frac{\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}} \right) \frac{\delta_1(\theta)}{\sigma(\theta)\sqrt{n(\theta)}} = \Pr\left\{ \tilde{J}(\theta) \le \mu_1 \right\} \phi\left(-\frac{\delta_1(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}} \right) \frac{\delta_1(\theta')}{\sigma(\theta')\sqrt{n(\theta')}}. \tag{16}$$

Taking natural log of both sides, then we have

$$\begin{aligned} &\log \Pr\left\{\tilde{J}(\theta') \leq \mu_1\right\} + \log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta)}{\sigma^2(\theta)} n(\theta) + \log \frac{\delta_1(\theta)}{\sigma(\theta)} - \frac{1}{2}\log n(\theta) \\ = \;& \log \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\} + \log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta')}{2\sigma^2(\theta')} n(\theta') + \log \frac{\delta_1(\theta')}{\sigma(\theta')} - \frac{1}{2}\log n(\theta'). \end{aligned} \tag{17}$$

Letting $n(\theta) = \alpha(\theta)T$ and $n(\theta') = \alpha(\theta')T$, and dividing both sides by T, then we have

$$\begin{aligned} &\frac{1}{T}\log \Pr\left\{\tilde{J}(\theta') \leq \mu_1\right\} + \frac{1}{T}\log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta)}{2\sigma^2(\theta)}\alpha(\theta) + \frac{1}{T}\log \frac{\delta_1(\theta)}{\sigma(\theta)} - \frac{1}{2T}\left(\log \alpha(\theta) + \log T\right) \\ = \;& \frac{1}{T}\log \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\} + \frac{1}{T}\log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta') + \frac{1}{T}\log \frac{\delta_1(\theta')}{\sigma(\theta')} - \frac{1}{2T}\left(\log \alpha(\theta') + \log T\right). \end{aligned} \tag{18}$$

Letting $T \to \infty$, we have

$$\frac{\delta_1^2(\theta)}{\sigma^2(\theta)}\alpha(\theta) = \frac{\delta_1^2(\theta')}{\sigma^2(\theta')}\alpha(\theta'). \tag{19}$$

Rearranging the terms, we have

$$\frac{n(\theta)}{n(\theta')} = \frac{\alpha(\theta)}{\alpha(\theta')} = \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_1(\theta')}\right)^2. \tag{20}$$

For the other choices of θ and θ', it is tedious but straightforward to show that we have

$$\frac{n(\theta)}{n(\theta')} = \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_1(\theta')}\right)^2. \tag{21}$$

Also note that Eq. (14) requires that $n(\theta) \geq 0$ for all $\theta \in \Theta$. Thus Eq. (20) provides an asymptotically optimal allocation for AP1. We summarize the above results into the following theorem.

Theorem 1: PCSm is asymptotically maximized when

$$\frac{n(\theta)}{n(\theta')} = \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_1(\theta')}\right)^2$$

for all $\theta, \theta' \in \Theta$.

Note that the difference between APCSm and PCSm depends on μ_1, which is a boundary separating the top-g designs from the rest of the designs. In order to minimize this difference we should pick μ_1 such that APCSm is maximized. Following similar analysis as in [9], we have

$$\mu_1 = \frac{\sigma(\theta_{i_{g+1}})\bar{J}(\theta_{i_g})/\sqrt{n(\theta_{i_{g+1}})} + \sigma(\theta_{i_g})\bar{J}(\theta_{i_{g+1}})/\sqrt{n(\theta_{i_g})}}{\sigma(\theta_{i_g})/\sqrt{n(\theta_{i_g})} + \sigma(\theta_{i_{g+1}})/\sqrt{n(\theta_{i_{g+1}})}}. \tag{22}$$

Following Theorem 1, we then have Algorithm 1.

Algorithm 1 Optimal computing budget allocation for m simplest good designs in the ordinal sense (OCBAmSGO)

Step 0: Simulate each design by n_0 replications; $l \leftarrow 0$; $n^l(\theta_1) = n^l(\theta_2) = \cdots = n^l(\theta_k) = n_0$.

Step 1: If $\sum_{i=1}^{k} n(\theta_i) \geq T$, stop.

Step 2: Increase the total simulation time by Δ and compute the new budget allocation $n^{l+1}(\theta_1), \ldots, n^{l+1}(\theta_k)$ using Theorem 1.

Step 3: Simulate design i for additional $\max\left(0, n^{l+1}(\theta_i) - n^l(\theta_i)\right)$ time, $i = 1, \ldots, k$; $l \leftarrow l + 1$. Go to step 1.

B. Selecting a bSG

The choice of S in Eq. (7) also provides an estimate of the bSG under the given observed performance. We can divide the entire design space into 4 subsets.

$$S_1 = \bigcup_{i=1, i\neq t+1}^{M} \bar{G}_i, \tag{23}$$

$$S_2 = \bigcup_{i=1}^{M} \bar{D}_i, \tag{24}$$

$$S_3 = \left\{\theta_{t+1,1}, \ldots, \theta_{t+1, m-\sum_{i=1}^{t} |\bar{G}_i|}\right\}, \tag{25}$$

$$S_4 = \bar{G}_{t+1} \setminus S_3. \tag{26}$$

In other words, S_1 contains all the observed good designs except those in $\bar{G}_{t+1}$; S_2 contains all the observed bad designs, including those in $\bar{D}_{t+1}$; $S_3 = S \cap \bar{G}_{t+1}$; S_4 contains the designs in $\bar{G}_{t+1}$ other than S_3. Then we have

$$\begin{aligned}
& PCSb \\
= \ & \Pr\{S \text{ is bSG}\} \\
\geq \ & \Pr\{S_1 \cup S_3 \cup S_4 \text{ is good, } S_2 \text{ is not good}, S_3 \text{ is better than } S_4\} \\
\geq \ & \Pr\left\{\tilde{J}(\theta) \leq \mu_1, \theta \in S_1; \tilde{J}(\theta) > \mu_1, \theta \in S_2; \tilde{J}(\theta) \leq \mu_2, \theta \in S_3; \mu_2 < \tilde{J}(\theta) \leq \mu_1, \theta \in S_4\right\} \\
= \ & \left(\Pi_{\theta\in S_1} \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\}\right)\left(\Pi_{\theta\in S_2} \Pr\left\{\tilde{J}(\theta) > \mu_1\right\}\right)\left(\Pi_{\theta\in S_3} \Pr\left\{\tilde{J}(\theta) \leq \mu_2\right\}\right)\left(\Pi_{\theta\in S_4} \Pr\left\{\mu_2 < \tilde{J}(\theta) \leq \mu_1\right\}\right) \\
\equiv \ & APCSb,
\end{aligned} \tag{27}$$

where μ_2 is a boundary separating the designs in S_3 from the designs in S_4. The difference between APCSb and PCSb depends on μ_1 and μ_2. In order to minimize this difference we should pick μ_1 and μ_2 such that APCSb is maximized. Following similar analysis as in [9], we use Eq. (22) to determine the value of μ_1, and we have

$$\mu_2 = \min\left(\mu_1, \frac{\sigma(\theta_{t+1,r+1})\bar{J}(\theta_{t+1,r})/\sqrt{n(\theta_{t+1,r+1})} + \sigma(\theta_{t+1,r})\bar{J}(\theta_{t+1,r+1})/\sqrt{n(\theta_{t+1,r})}}{\sigma(\theta_{t+1,r})/\sqrt{n(\theta_{t+1,r})} + \sigma(\theta_{t+1,r+1})/\sqrt{n(\theta_{t+1,r+1})}}\right), \tag{28}$$

where $r = m - \sum_{i=1}^{t} |\bar{G}_i|$. Then an approximate version of P2 is

(AP2) $\max_{n(\theta),\theta\in\Theta} APCSb$ s.t. $\sum_{i=1}^{k} n(\theta_i) = T$.

Let L_b be the Lagrangian relaxation of AP2.

$$L_b \equiv APCSb + \lambda_b\left(\sum_{i=1}^{k} n(\theta_i) - T\right). \tag{29}$$

The KKT conditions of AP2 are as follows.

For $\theta \in S_1$,

$$\begin{aligned}\frac{\partial L_b}{\partial n(\theta)} &= \left(\Pi_{\theta\in S_1,\theta'\neq\theta} \Pr\left\{\tilde{J}(\theta') \le \mu_1\right\}\right)\left(\Pi_{\theta'\in S_2} \Pr\left\{\tilde{J}(\theta') > \mu_1\right\}\right)\left(\Pi_{\theta'\in S_3} \Pr\left\{\tilde{J}(\theta') \le \mu_2\right\}\right) \\ &\quad \left(\Pi_{\theta'\in S_4} \Pr\left\{\mu_2 < \tilde{J}(\theta') \le \mu_1\right\}\right)\phi\left(\frac{-\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right)\frac{-\delta_1(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}} + \lambda_b = 0;\end{aligned} \tag{30}$$

for $\theta \in S_2$,

$$\begin{aligned}\frac{\partial L_b}{\partial n(\theta)} &= \left(\Pi_{\theta'\in S_1} \Pr\left\{\tilde{J}(\theta') \le \mu_1\right\}\right)\left(\Pi_{\theta'\in S_2,\theta'\neq\theta} \Pr\left\{\tilde{J}(\theta') > \mu_1\right\}\right)\left(\Pi_{\theta'\in S_3} \Pr\left\{\tilde{J}(\theta') \le \mu_2\right\}\right) \\ &\quad \left(\Pi_{\theta'\in S_4} \Pr\left\{\mu_2 < \tilde{J}(\theta') \le \mu_1\right\}\right)\phi\left(\frac{\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right)\frac{\delta_1(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}} + \lambda_b = 0;\end{aligned} \tag{31}$$

for $\theta \in S_3$,

$$\begin{aligned}\frac{\partial L_b}{\partial n(\theta)} &= \left(\Pi_{\theta'\in S_1} \Pr\left\{\tilde{J}(\theta') \le \mu_1\right\}\right)\left(\Pi_{\theta'\in S_2} \Pr\left\{\tilde{J}(\theta') > \mu_1\right\}\right)\left(\Pi_{\theta'\in S_3,\theta'\neq\theta} \Pr\left\{\tilde{J}(\theta') \le \mu_2\right\}\right) \\ &\quad \left(\Pi_{\theta'\in S_4} \Pr\left\{\mu_2 < \tilde{J}(\theta') \le \mu_1\right\}\right)\phi\left(\frac{-\delta_2(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right)\frac{-\delta_2(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}} + \lambda_b = 0;\end{aligned} \tag{32}$$

for $\theta \in S_4$,

$$\begin{aligned}\frac{\partial L_b}{\partial n(\theta)} &= \left(\Pi_{\theta'\in S_1} \Pr\left\{\tilde{J}(\theta') \le \mu_1\right\}\right)\left(\Pi_{\theta'\in S_2} \Pr\left\{\tilde{J}(\theta') > \mu_1\right\}\right)\left(\Pi_{\theta'\in S_3} \Pr\left\{\tilde{J}(\theta') \le \mu_2\right\}\right) \\ &\quad \left(\Pi_{\theta'\in S_4,\theta'\neq\theta} \Pr\left\{\mu_2 < \tilde{J}(\theta') \le \mu_1\right\}\right) \\ &\quad \left[\phi\left(\frac{-\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right)\frac{-\delta_1(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}} - \phi\left(\frac{-\delta_2(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right)\frac{-\delta_2(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}}\right] + \lambda_b = 0;\end{aligned} \tag{33}$$

and

$$\frac{\partial L_b}{\partial \lambda_b} = \sum_{i=1}^{k} n(\theta_i) - T = 0, \tag{34}$$

where $\delta_2(\theta) = \bar{J}(\theta) - \mu_2$.

We now analyze the relationship among $n(\theta)$'s, $\theta \in \Theta$.

Case 1. $\theta, \theta' \in S_1$. Following Eq. (30), we have

$$\begin{aligned}
& \left(\Pi_{\theta'' \in S_1, \theta'' \neq \theta} \Pr\left\{\tilde{J}(\theta'') \leq \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_2} \Pr\left\{\tilde{J}(\theta'') > \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_3} \Pr\left\{\tilde{J}(\theta'') \leq \mu_2\right\}\right) \\
& \left(\Pi_{\theta'' \in S_4} \Pr\left\{\mu_2 < \tilde{J}(\theta'') \leq \mu_1\right\}\right) \phi\left(\frac{-\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right) \frac{-\delta_1(\theta)}{\sigma(\theta)\sqrt{n(\theta)}} \\
= & \left(\Pi_{\theta'' \in S_1, \theta'' \neq \theta'} \Pr\left\{\tilde{J}(\theta'') \leq \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_2} \Pr\left\{\tilde{J}(\theta'') > \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_3} \Pr\left\{\tilde{J}(\theta'') \leq \mu_2\right\}\right) \\
& \left(\Pi_{\theta'' \in S_4} \Pr\left\{\mu_2 < \tilde{J}(\theta'') \leq \mu_1\right\}\right) \phi\left(\frac{-\delta_1(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}}\right) \frac{-\delta_1(\theta')}{\sigma(\theta')\sqrt{n(\theta')}}.
\end{aligned} \tag{35}$$

Simplifying Eq. (35), then we have

$$\Pr\left\{\tilde{J}(\theta') \leq \mu_1\right\} \phi\left(\frac{-\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right) \frac{\delta_1(\theta)}{\sigma(\theta)\sqrt{n(\theta)}} = \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\} \phi\left(\frac{-\delta_1(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}}\right) \frac{\delta_1(\theta')}{\sigma(\theta')\sqrt{n(\theta')}}. \tag{36}$$

Taking natural log of both sides, we have

$$\begin{aligned}
& \log \Pr\left\{\tilde{J}(\theta') \leq \mu_1\right\} + \log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta)}{2\sigma^2(\theta)} n(\theta) + \log \frac{\delta_1(\theta)}{\sigma(\theta)} - \frac{1}{2} \log n(\theta) \\
= & \log \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\} + \log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta')}{2\sigma^2(\theta')} n(\theta') + \log \frac{\delta_1(\theta')}{\sigma(\theta')} - \frac{1}{2} \log n(\theta').
\end{aligned} \tag{37}$$

Let $n(\theta) = \alpha(\theta)T$ and $n(\theta') = \alpha(\theta')T$. Dividing both sides by T, we have

$$\begin{aligned}
& \frac{1}{T} \log \Pr\left\{\tilde{J}(\theta') \leq \mu_1\right\} + \frac{1}{T} \log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta)}{2\sigma^2(\theta)} \alpha(\theta) + \frac{1}{T} \log \frac{\delta_1(\theta)}{\sigma(\theta)} - \frac{1}{2T} \log \alpha(\theta)T \\
= & \frac{1}{T} \log \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\} + \frac{1}{T} \log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta')}{2\sigma^2(\theta')} \alpha(\theta') + \frac{1}{T} \log \frac{\delta_1(\theta')}{\sigma(\theta')} - \frac{1}{2T} \log \alpha(\theta')T.
\end{aligned} \tag{38}$$

Letting $T \to \infty$, we have

$$\frac{\delta_1^2(\theta)}{\sigma^2(\theta)} \alpha(\theta) = \frac{\delta_1^2(\theta')}{\sigma^2(\theta')} \alpha(\theta'). \tag{39}$$

Rearranging the terms, then we have

$$\frac{n(\theta)}{n(\theta')} = \frac{\alpha(\theta)}{\alpha(\theta')} = \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_1(\theta')}\right)^2. \tag{40}$$

Following similar analysis we have that for $\theta, \theta' \in S_1 \cup S_2 \cup S_3$,

$$\frac{n(\theta)}{n(\theta')} = \left(\frac{\sigma(\theta)/\nu(\theta)}{\sigma(\theta')/\nu(\theta')}\right)^2, \tag{41}$$

where

$$\nu(\theta) = \begin{cases} \delta_1(\theta), & \theta \in S_1 \cup S_2, \\ \delta_2(\theta), & \theta \in S_3. \end{cases}$$

Case 2. $\theta \in S_1, \theta' \in S_4$. Following Eqs. (30) and (33), we have

$$\begin{aligned} & \left(\Pi_{\theta'' \in S_1, \theta'' \neq \theta} \Pr\left\{\tilde{J}(\theta'') \leq \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_2} \Pr\left\{\tilde{J}(\theta'') > \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_3} \Pr\left\{\tilde{J}(\theta'') \leq \mu_2\right\}\right) \\ & \left(\Pi_{\theta'' \in S_4} \Pr\left\{\mu_2 < \tilde{J}(\theta'') \leq \mu_1\right\}\right) \phi\left(\frac{-\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right) \frac{-\delta_1(\theta)}{2\sigma(\theta)\sqrt{n(\theta)}} \\ = & \left(\Pi_{\theta'' \in S_1} \Pr\left\{\tilde{J}(\theta'') \leq \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_2} \Pr\left\{\tilde{J}(\theta'') > \mu_1\right\}\right)\left(\Pi_{\theta'' \in S_3} \Pr\left\{\tilde{J}(\theta'') \leq \mu_2\right\}\right) \\ & \left(\Pi_{\theta'' \in S_4, \theta'' \neq \theta'} \Pr\left\{\mu_2 < \tilde{J}(\theta'') \leq \mu_1\right\}\right) \\ & \left[\phi\left(\frac{-\delta_1(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}}\right) \frac{-\delta_1(\theta')}{2\sigma(\theta')\sqrt{n(\theta')}} - \phi\left(\frac{-\delta_2(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}}\right) \frac{-\delta_2(\theta')}{2\sigma(\theta')\sqrt{n(\theta')}}\right]. \end{aligned} \tag{42}$$

Simplifying Eq. (42), then we have

$$\Pr\left\{\mu_2 < \tilde{J}(\theta') \leq \mu_1\right\} \phi\left(\frac{-\delta_1(\theta)}{\sigma(\theta)/\sqrt{n(\theta)}}\right) \frac{-\delta_1(\theta)}{\sigma(\theta)\sqrt{n(\theta)}} = \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\} A, \tag{43}$$

where

$$A = \phi\left(\frac{-\delta_1(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}}\right) \frac{-\delta_1(\theta')}{\sigma(\theta')\sqrt{n(\theta')}} - \phi\left(\frac{-\delta_2(\theta')}{\sigma(\theta')/\sqrt{n(\theta')}}\right) \frac{-\delta_2(\theta')}{\sigma(\theta')\sqrt{n(\theta')}}. \tag{44}$$

Taking natural log of both sides, we have

$$\log \Pr\left\{\mu_2 < \tilde{J}(\theta') \leq \mu_1\right\} + \log \frac{1}{\sqrt{2\pi}} - \frac{\delta_1^2(\theta)}{2\sigma^2(\theta)} n(\theta) + \log \frac{-\delta_1(\theta)}{\sigma(\theta)} - \frac{1}{2} \log n(\theta) = \log \Pr\left\{\tilde{J}(\theta) \leq \mu_1\right\} + \log A. \tag{45}$$

Rearranging the terms, we have

$$-\frac{\delta_1^2(\theta)\alpha(\theta)}{2\sigma^2(\theta)} = \lim_{T\to\infty} \frac{1}{T} \log A. \tag{46}$$

Note that by L'Hôpital's rule we have

$$\lim_{T\to\infty} \frac{1}{T} \log A = \lim_{T\to\infty} \frac{dA/dT}{A}, \tag{47}$$

where

$$\frac{dA/dT}{A} = \frac{\exp\left\{-\frac{\delta_1^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T\right\}\left[\frac{\delta_1^3(\theta')}{2\sigma^3(\theta')}\sqrt{\frac{\alpha(\theta')}{T}} + \frac{\delta_1(\theta')}{2\sigma(\theta')}\frac{1}{\sqrt{\alpha(\theta)T^3}}\right] - \exp\left\{-\frac{\delta_2^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T\right\}\left[\frac{\delta_2^3(\theta')}{2\sigma^3(\theta')}\sqrt{\frac{\alpha(\theta')}{T}} + \frac{\delta_2(\theta')}{2\sigma(\theta')}\frac{1}{\sqrt{\alpha(\theta')T^3}}\right]}{\exp\left\{-\frac{\delta_1^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T\right\}\frac{-\delta_1(\theta')}{\sigma(\theta')\sqrt{\alpha(\theta')T}} - \exp\left\{-\frac{\delta_2^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T\right\}\frac{-\delta_2(\theta')}{\sigma(\theta')\sqrt{\alpha(\theta')T}}}. \tag{48}$$

Dividing both the numerator and the denominator on the right-hand-side (RHS) of Eq. (48) by $\exp\left\{-\frac{\delta_2^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T\right\}$, we have

$$\frac{dA/dT}{A} = \frac{\exp\left\{-\frac{\delta_1^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T - \frac{-\delta_2^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T\right\}\left[\frac{\delta_1^3(\theta')}{2\sigma^3(\theta')}\sqrt{\frac{\alpha(\theta')}{T}} + \frac{\delta_1(\theta')}{2\sigma(\theta')}\sqrt{\frac{1}{\alpha(\theta')T^3}}\right] - \left[\frac{\delta_2^3(\theta')}{2\sigma^3(\theta')}\sqrt{\frac{\alpha(\theta')}{T}} + \frac{\delta_2(\theta')}{2\sigma(\theta')}\sqrt{\frac{1}{\alpha(\theta')T^3}}\right]}{\exp\left\{-\frac{\delta_1^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T - \frac{-\delta_2^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta')T\right\}\frac{-\delta_1(\theta')}{\sigma(\theta')\sqrt{\alpha(\theta')T}} + \frac{\delta_2(\theta')}{\sigma(\theta')\sqrt{\alpha(\theta')T}}}. \tag{49}$$

If $|\delta_2(\theta')| > |\delta_1(\theta')|$, i.e., $\bar{J}(\theta') > (\mu_1 + \mu_2)/2$, then we have

$$\lim_{T\to\infty}\frac{dA/dT}{A} = -\frac{\delta_1^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta'). \tag{50}$$

Combining Eqs. (46), (47), and (50), we then have

$$\frac{\alpha(\theta)}{\alpha(\theta')} = \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_1(\theta')}\right)^2. \tag{51}$$

If $|\delta_2(\theta')| < |\delta_1(\theta')|$, i.e., $\bar{J}(\theta') \leq (\mu_1 + \mu_2)/2$, then we have

$$\lim_{T\to\infty}\frac{dA/dT}{A} = -\frac{\delta_2^2(\theta')}{2\sigma^2(\theta')}\alpha(\theta'). \tag{52}$$

Combining Eqs. (46), (47), and (52), we then have

$$\frac{\alpha(\theta)}{\alpha(\theta')} = \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_2(\theta')}\right)^2. \tag{53}$$

Combining Eqs. (51) and (53), we have

$$\frac{n(\theta)}{n(\theta')} = \begin{cases} \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_1(\theta')}\right)^2, & \text{if } \bar{J}(\theta') > \frac{\mu_1+\mu_2}{2}, \\ \left(\frac{\sigma(\theta)/\delta_1(\theta)}{\sigma(\theta')/\delta_2(\theta')}\right)^2, & \text{if } \bar{J}(\theta') \leq \frac{\mu_1+\mu_2}{2}. \end{cases} \tag{54}$$

Similarly, for $\theta \in S_1 \cup S_2 \cup S_3$ and $\theta' \in S_4$, we have

$$\frac{n(\theta)}{n(\theta')} = \begin{cases} \left(\frac{\sigma(\theta)/\nu(\theta)}{\sigma(\theta')/\delta_1(\theta')}\right)^2, & \text{if } \bar{J}(\theta') > \frac{\mu_1+\mu_2}{2}, \\ \left(\frac{\sigma(\theta)/\nu(\theta)}{\sigma(\theta')/\delta_2(\theta')}\right)^2, & \text{if } \bar{J}(\theta') \leq \frac{\mu_1+\mu_2}{2}. \end{cases} \tag{55}$$

In other words, we can further split S_4 into two subsets

$$S_{41} = \left\{\theta \in S_4 \text{ and } \bar{J}(\theta) > \frac{\mu_1 + \mu_2}{2}\right\}, \tag{56}$$

$$S_{42} = \left\{\theta \in S_4 \text{ and } \bar{J}(\theta) \leq \frac{\mu_1 + \mu_2}{2}\right\}. \tag{57}$$

Algorithm 2 Optimal computing budget allocation for the best m simplest good designs in the ordinal sense (OCBAbSGO)

Step 0: Simulate each design by n_0 replications; $l \leftarrow 0$; $n^l(\theta_1) = n^l(\theta_2) = \cdots = n^l(\theta_k) = n_0$.

Step 1: If $\sum_{i=1}^{k} n(\theta_i) \geq T$, stop.

Step 2: Increase the total simulation time by Δ and compute the new budget allocation $n^{l+1}(\theta_1), \ldots, n^{l+1}(\theta_k)$ using Theorem 2.

Step 3: Simulate design i for additional $\max\left(0, n^{l+1}(\theta_i) - n^l(\theta_i)\right)$ time, $i = 1, \ldots, k$; $l \leftarrow l + 1$. Go to step 1.

Combining Eqs. (41), (54), and (55) together, we have

$$\left.\frac{n(\theta)}{(\sigma(\theta)/\delta_1(\theta))^2}\right|_{\theta \in S_1 \cup S_2 \cup S_{41}} = \left.\frac{n(\theta)}{(\sigma(\theta)/\delta_2(\theta))^2}\right|_{\theta \in S_3 \cup S_{42}}. \tag{58}$$

Then we have the following theorem.

Theorem 2: PCSb is asymptotically maximized when

$$\left.\frac{n(\theta)}{(\sigma(\theta)/\delta_1(\theta))^2}\right|_{\theta \in S_1 \cup S_2 \cup S_{41}} = \left.\frac{n(\theta)}{(\sigma(\theta)/\delta_2(\theta))^2}\right|_{\theta \in S_3 \cup S_{42}}. \tag{59}$$

We then have Algorithm 2.

Note that though the above choices of S in mSG and bSG are the same, their allocations are clearly different. OCBAmSGO tries to make sure that $\bar{G}$ are truly top-g. Then the choice of S in Eq. (7) will make sure that the simplest m designs in $\bar{G}$ are picked. OCBAbSGO tries to furthermore make sure that designs in S_3 are better than designs in S_4.

IV. Numerical Results

We compare OCBAmSGO (Algorithm 1, or A1 for short) and OCBAbSGO (Algorithm 2, or A2 for short) with equal allocation (EA) over two groups of examples. The first group includes academic examples. The second group includes smoke detection problems in wireless sensor networks (WSNs).

A. Academic Examples

In order to capture different relationships between cardinal performance and ordinal indexes of designs, the following three types of ordered performance are considered.

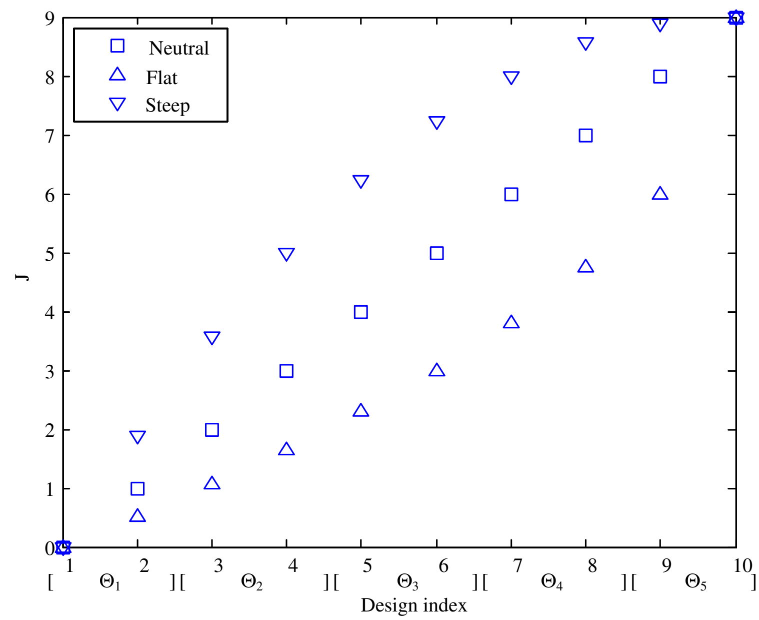

Fig. 1. The three examples where simpler is better.

1) Neutral, $J(\theta_{[i]}) = i - 1, i = 1, \ldots, 10$;

2) Flat, $J(\theta_{[i]}) = 9 - 3\sqrt{10 - i}, i = 1, \ldots, 10$;

3) Steep, $J(\theta_{[i]}) = 9 - \left(\frac{10-i}{3}\right)^2, i = 1, \ldots, 10$,

where $\theta_{[i]}$ represents the top i-th design. In the neutral type of problems, the performance difference among neighboring designs are equal. In the flat type of problems, most designs have good performance. On the contrary, in the steep type of problems, most designs have bad performance. The following two types of relationship between performance and complexity are considered.

1) Simpler is better, i.e., if $C(\theta) < C(\theta')$, then $J(\theta) < J(\theta')$;

2) Simpler is worse, i.e., if $C(\theta) < C(\theta')$, then $J(\theta) > J(\theta')$.

When simpler is better, we have $\theta_{[i]} = \theta_i, i = 1, \ldots, 10$. When simpler is worse, we have $\theta_{[i]} = \theta_{11-i}, i = 1, \ldots, 10$.

Combining the above two considerations, we then have 6 types of problems. In each problem, $\Theta_1 = \{\theta_1, \theta_2\}$, $\Theta_2 = \{\theta_3, \theta_4\}$, $\Theta_3 = \{\theta_5, \theta_6\}$, $\Theta_4 = \{\theta_7, \theta_8\}$, and $\Theta_5 = \{\theta_9, \theta_{10}\}$. The performance of the 6 problems are shown in Figs. 1 and 2.

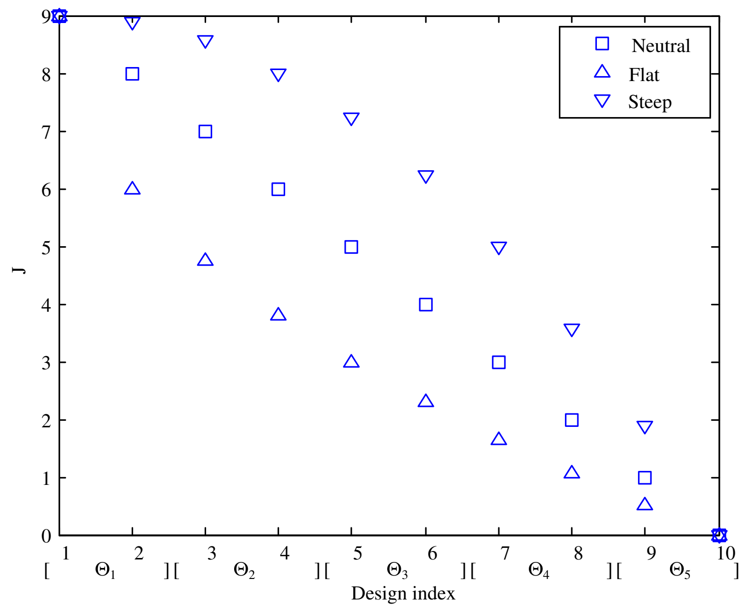

Fig. 2. The three examples where simpler is worse.

Regard the top-3 designs as good, i.e., $g = 3$. We are interested in the m-simplest good designs, where $m = 2$. Suppose the observation noise for each design is i.i.d. Gaussian $N(0, 6^2)$. We use OCBAmSGO, OCBAbSGO, and EA to find the mSG and bSG, respectively, where $n_0 = 30$ and $\Delta = 10$. Their PCSm and PCSb for different T's are estimated over 100,000 independent runs and shown in Figs. 3-8, respectively. We make the following remarks.

Remark 1. In all of the 6 problems, OCBAmSGO (Algorithm 1) achieves higher PCSm's than EA; and OCBAbSGO (Algorithm 2) achieves higher PCSb's than EA.

Remark 2. In all of the 6 problems, when T is fixed, the PCSm that is achieved by OCBAmSGO is higher than the PCSb that is achieved by OCBAbSGO. Similarly, EA achieves higher PCSm's than PCSb's. This is consistent with the fact that a bSG must be an mSG but an mSG is not necessarily a bSG.

Remark 3. When T increases, OCBAmSGO, OCBAbSGO, and EA all achieve higher PCS's. This is consistent with intuition because more computing budget should lead to higher PCS's.

Remark 4. For a given T, OCBAmSGO, OCBAbSGO, and EA achieve the highest PCS's in the steep problems, achieve lower PCS's in the neutral problems, and achieve the lowest PCS's in the flat problems. This is consistent with intuition because the performance differences among

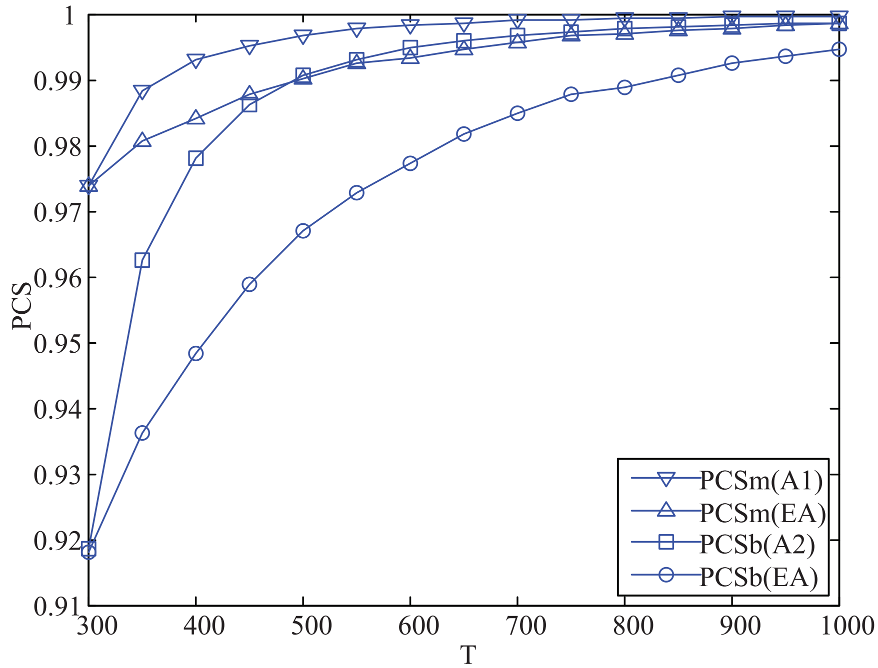

Fig. 3. The neutral problem where simpler is better.

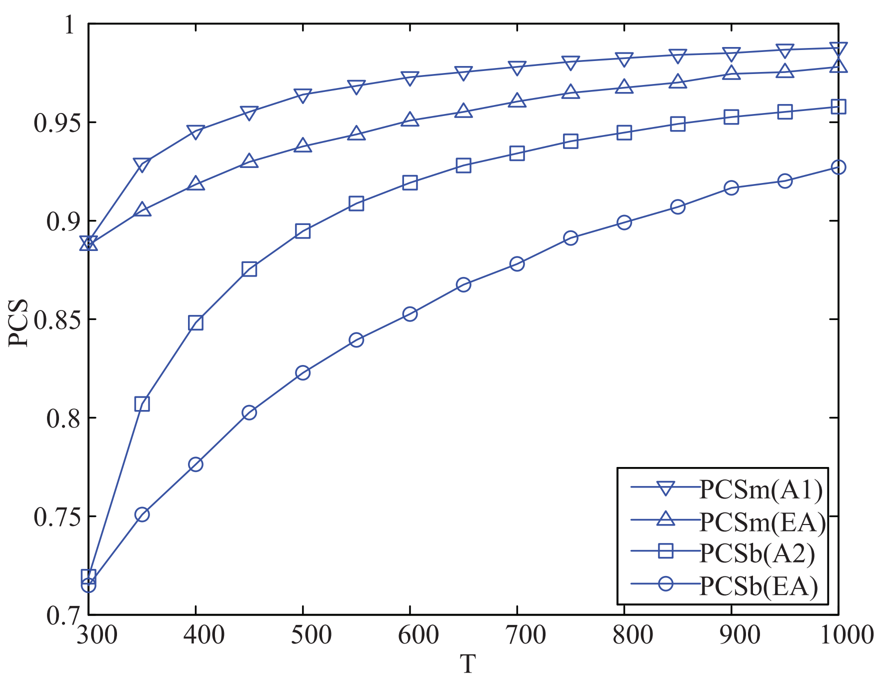

Fig. 4. The flat problem where simpler is better.

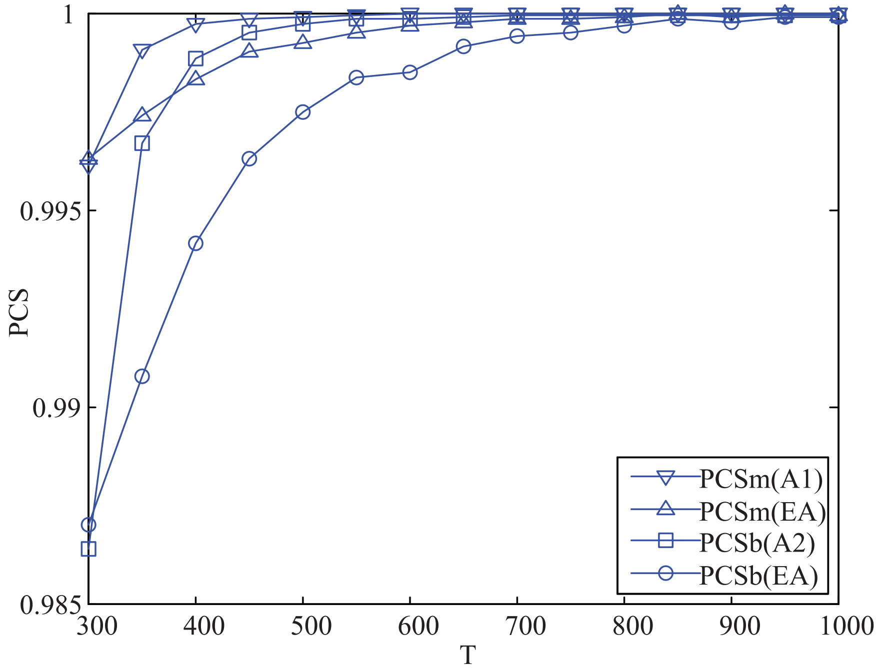

Fig. 5. The steep problem where simpler is better.

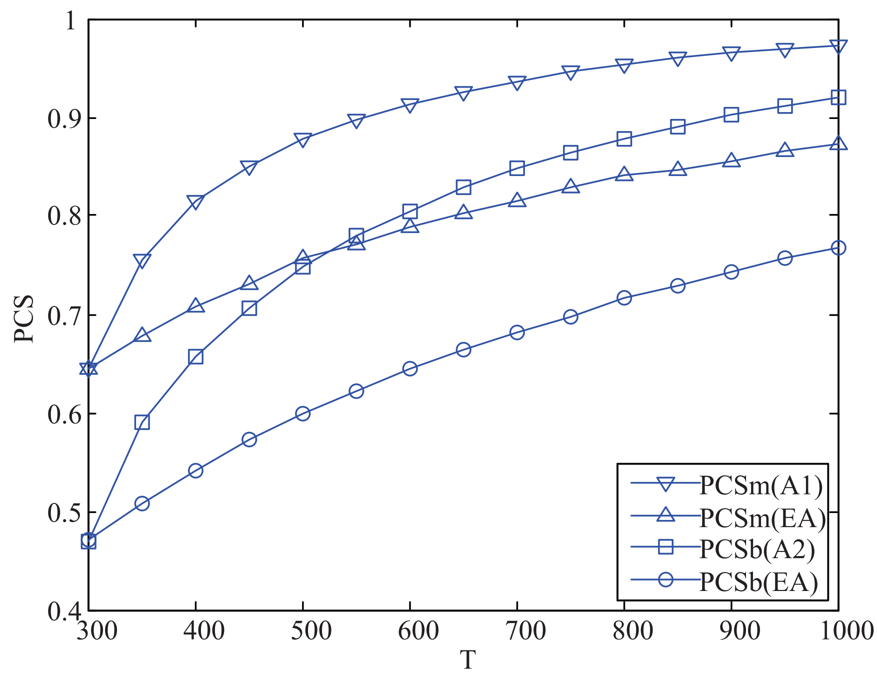

Fig. 6. The neutral problem where simpler is worse.

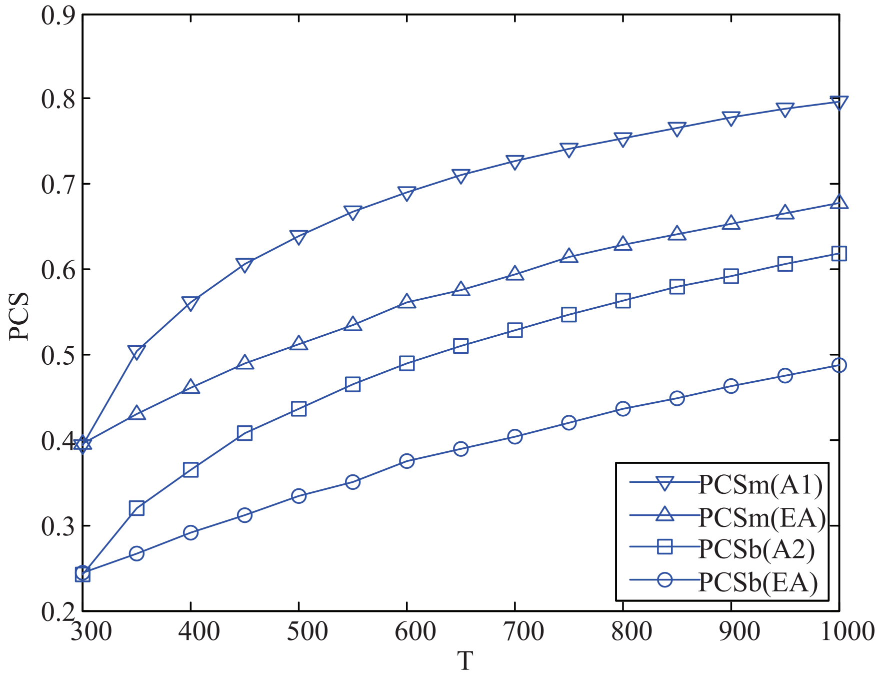

Fig. 7. The flat problem where simpler is worse.

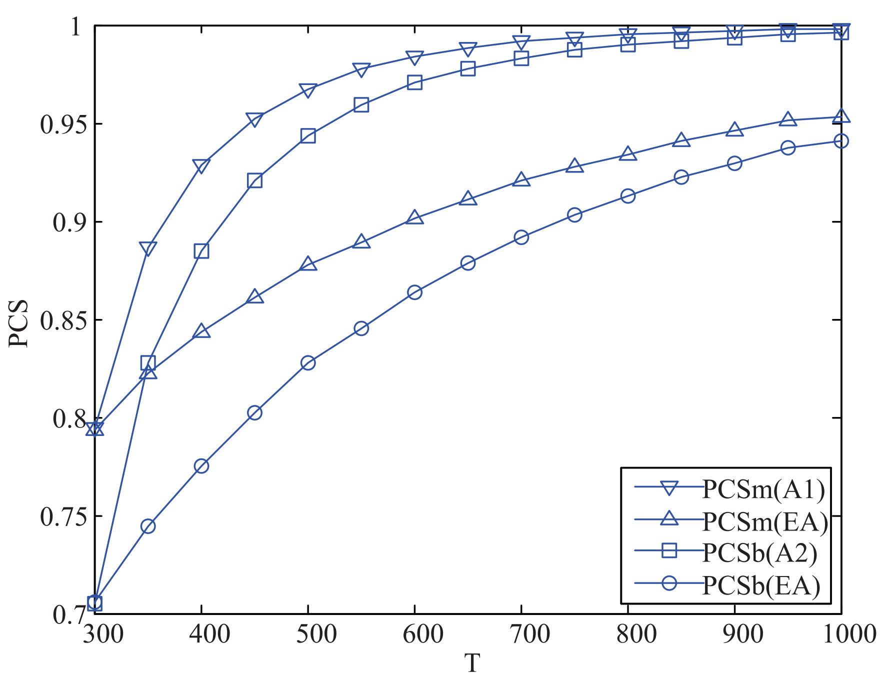

Fig. 8. The steep problem where simpler is worse.

the good designs in the steep problems are larger than that in the neutral problems, which in turn are larger than that in the flat problems.

Remark 5. Note that in our academic examples where simpler is better PCSm only requires 2 of the actual top 3 designs to be within the observed top 3. But when simpler is worse PCSm requires all of the actual top 3 designs to be observed as top 3. Similarly, when simpler is better PCSb only requires the actual top 2 designs to be observed as top 2. But when simpler is worse PCSb requires all of the actual top 3 designs to be observed as top 3, and furthermore requires the truly best design to be observed as better than the truly second best design. This explains why the PCS's in Figs. 3-5 are higher than the PCS's in Figs. 6-8. Because the performance differences between the good and bad designs in the steep problems are larger than those in the neutral problems, which in turn are larger than those in the flat problems, the flat problem where simpler is worse (Fig. 7) has significantly lower PCS than where simpler is better (Fig. 4). Note that though the PCS's are lower in problems where simpler is worse, they still converge to 1 when there is an infinite computing budget. It is an interesting future research topic to explore the knowledge of "simpler is worse" or "simpler is better" to develop a better allocation procedure. It is also interesting to study how to change the complexities of the designs so that a higher PCS may be achieved under a given computing budget.

B. Smoke Detection in WSN

Consider a WSN with 3 nodes that are used to detect smoke in an area of interest (AoI). AoI is discretized into 10×10 grids in Fig. 9. A fire may be set up at any point on the grid in AoI with equal probability. Once set up, a smoke particle will be generated at the fire source at any unit of time. An existing smoke particle will walk randomly to a neighboring point along the grid with a positive probability. There are four possible directions to walk. Each direction is taken with some probability, i.e.,

$$\Pr\{x_{t+1} = x_t + 1, y_{t+1} = y_t\} \quad \propto \quad d((x_t + 1, y_t), (x_0, y_0)), \tag{60}$$

$$\Pr\{x_{t+1} = x_t - 1, y_{t+1} = y_t\} \quad \propto \quad d((x_t - 1, y_t), (x_0, y_0)), \tag{61}$$

$$\Pr\{x_{t+1} = x_t, y_{t+1} = y_t + 1\} \quad \propto \quad d((x_t, y_t + 1), (x_0, y_0)), \tag{62}$$

$$\Pr\{x_{t+1} = x_t, y_{t+1} = y_t - 1\} \quad \propto \quad d((x_t, y_t - 1), (x_0, y_0)), \tag{63}$$

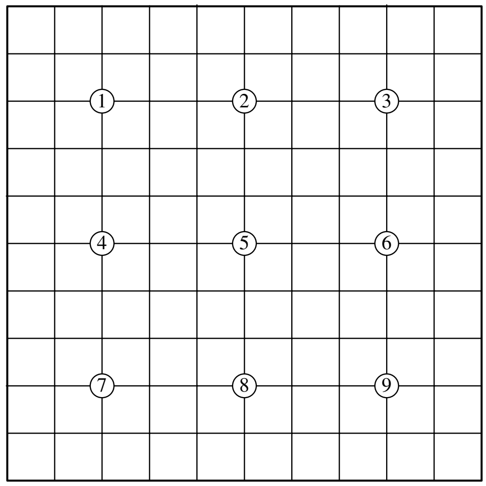

Fig. 9. A smoke detection problem in wireless sensor network.

where (x_t, y_t) is the position of the smoke particle at time t, (x_0, y_0) is the position of the fire source, and $d(\cdot, \cdot)$ is the distance between two positions. In other words, the fire is pushing the smoke away with the above specified probability. The sensors can be deployed to 3 of the 9 positions. Once deployed, the sensors use an identical active sensing radius $r \in \{0, 1, 2, 3\}$, where $r = 0$ means that the sensor detects the smoke (and thus triggers the alarm) only if a smoke particle arrives at the sensor (This is purely passive detection.); $r > 1$ means that the sensor detects the smoke if a smoke particle is within r grids away from the sensor. Power consumption increases as the active sensing radius increases. Thus, we are interested in good deployment with short sensing radius. When we pick out m simplest good deployments, it is up to the final user to determine which deployment to use. Usually only one deployment is eventually selected and implemented. There are 84 possible ways of deployment of the 3 sensors, but taking out all of the symmetric possibilities leaves us with only 16 to analyze, as seen in Fig. 10. Let r be the complexity measure. Thus $|\Theta_i| = 16, i = 1, 2, 3, 4$ and $|\Theta| = 64$. For each deployment and sensing radius, we are interested in the probability that a smoke is detected within T_0 time, where $T_0 = 5$ in our numerical experiments. We evaluate the performance of the 64 designs over 10,000 independent simulations, which are shown in Fig. 11 together with the standard deviations of their performance observations. We regard the top-20 designs as good enough, i.e., $g = 20$, which are denoted by big circles in Fig. 11. For different T's and m's, we apply OCBAmSGO,

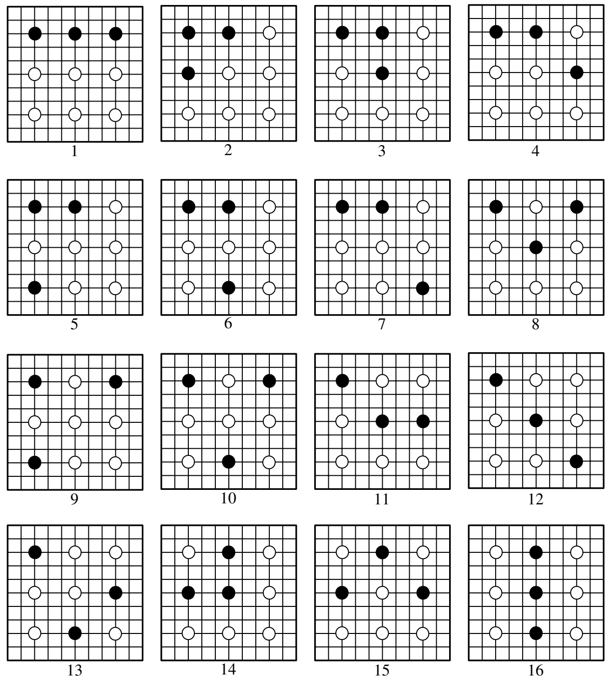

Fig. 10. The 16 deployments of the 3 sensors.

OCBAbSGO, and EA to get mSG's and bSG's, estimate their PCSm's and PCSb's using 100,000 independent runs of the algorithms, and show the results in Figs. 12 and 13, respectively, where $n_0 = 20$ and $\Delta = 10$. We make the following remarks.

Remark 6. OCBAmSGO (A1) achieves higher PCSm's than EA for both values of m and for all the computing budgets T. This clearly demonstrates the advantage of OCBAmSGO over EA even when the observation noise does not have Gaussian distribution and when there are only finite computing budgets.

Remark 7. OCBAbSGO (A2) achieves higher PCSb's than EA for both values of m and for all the computing budgets T. Similarly, this demonstrates the advantage of OCBAbSGO over EA even when the observation noise does not have Gaussian distribution and when there are only finite computing budgets.

Remark 8. When m increases from 19 to 20, both OCBAmSGO and EA achieve lower PCSm's. Similarly, both OCBAbSGO and EA achieve lower PCSb's. However, this does not imply that PCS is a monotonic function of m. For example, suppose $\Theta = \{\theta_1, \theta_2, \theta_3\}, \Theta_1 = \{\theta_1\}, \Theta_2 =$

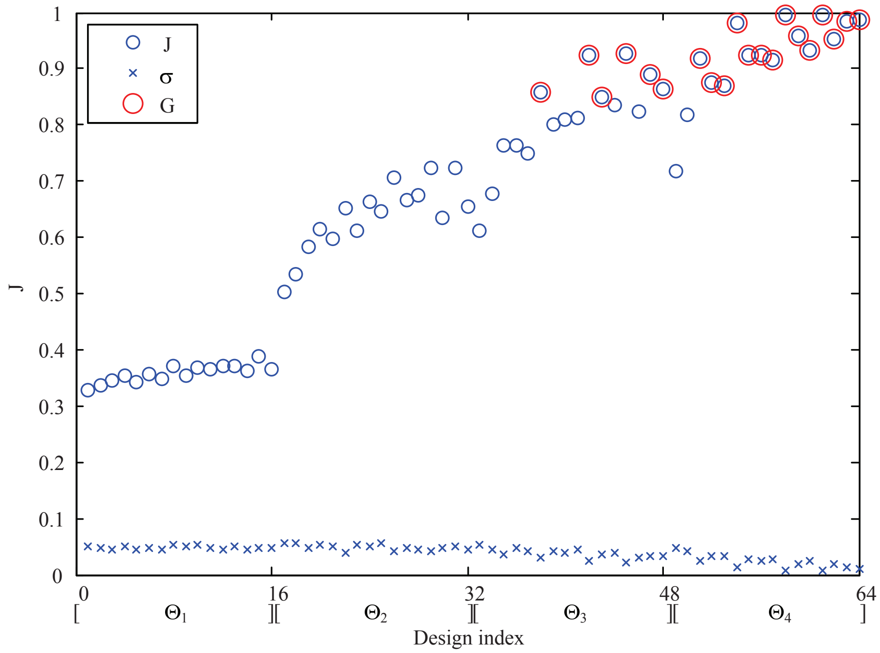

Fig. 11. The performance of the 64 designs.

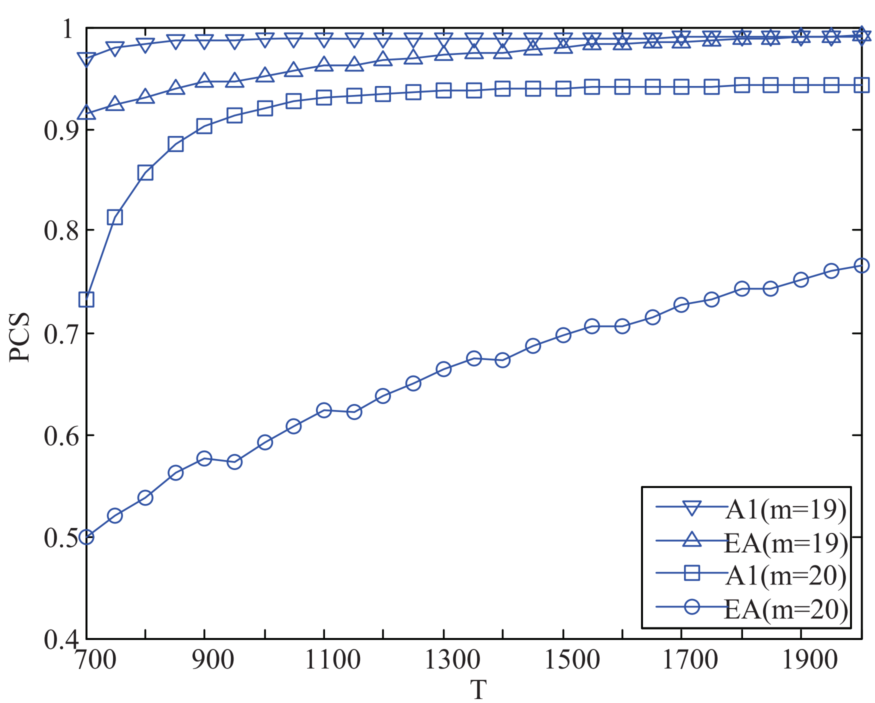

Fig. 12. PCSm's of EA and A1 ($m = 19, 20$).

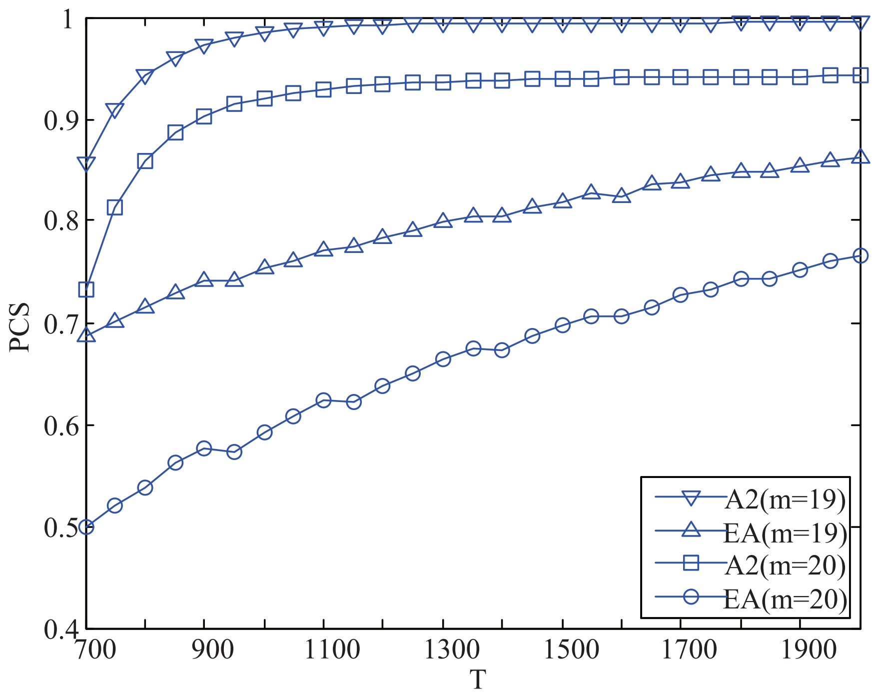

Fig. 13. PCSb's of EA and A2 ($m = 19, 20$).

$\{\theta_2\}, \Theta_3 = \{\theta_3\}, J(\theta_i) = i, i = 1, 2, 3$. Let $g = 3$. When $m = 2$, as long as there is observation noise, $PCSm < 1$ and $PCSb < 1$. But when $m = 3$, $PCSm = 1$ and $PCSb = 1$. This clearly shows that both PCSm and PCSb are not necessarily decreasing functions w.r.t. m.

Remark 9. When $m = 19$, the PCSm's achieved by OCBAmSGO and EA are both higher than the PCSb's achieved by OCBAbSGO and EA, respectively. This is intuitively reasonable because a bSG is an mSG while the reverse is not necessarily true. When $m = 20$, the PCSm's achieved by OCBAmSGO and EA are equal to the PCSb's achieved by OCBAbSGO and EA, respectively, because in this case the bSG coincides with the mSG.

V. Conclusion

Simple designs with good ordinal performance are of special interest in many applications, especially when it is difficult to specify what cardinal values are good. In this paper, we consider the computing budget allocation problem in selecting m simplest good designs and develop OCBAmSGO and OCBAbSGO to asymptotically maximize the probabilities for correctly selecting such an mSG and the best mSG, respectively. We compare their performance with equal allocation on several numerical examples. Though we assume Gaussian observation noise in this paper, the numerical results indicate that both OCBAmSGO and OCBAbSGO have good performance when the observation noise is not Gaussian and when the computing budget is finite. Note that APCSm and APCSb are lower bounds for PCSm and PCSb, respectively. They are derived by considering only the case that the observed top g designs are truly top g. It is possible for the choice of S in Eq. (7) to be an mSG (or bSG) even though $\bar{G} \neq G$, i.e., when an observed top g design is not truly top g. Exploring this case may lead to tighter lower bounds and better allocation procedures, which is an important further research direction. Also note that only a single objective function with no simulation-based constraints is considered in this paper. It is an interesting future research topic to extend the work in this paper to problems with multiple objective functions and simulation-based constraints. We hope this work brings more insight to finding simple and good designs in general.

Acknowledgements

The authors would like to thank Mr. Grover Laporte, the editor, and the anonymous reviewers for their constructive comments on previous versions of this paper.

References

[1] R. E. Bechhofer, “A single-sample multiple decision procedure for ranking means of normal populations with known variances,” *Annals of Mathematical Statistics*, vol. 25, pp. 16–39, 1954.

[2] S. S. Gupta, “On a decision rule for a problem in ranking means,” Ph.D. dissertation, Institute of Statistics, University of North Carolina, Chapel Hill, NC, 1956.

[3] ——, “On some multiple decision (ranking and selection) rules,” *Technometrics*, vol. 7, pp. 225–245, 1965.

[4] R. E. Bechhofer, T. J. Santner, and D. Goldsman, *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*, New York, NY: John Wiley & Sons, 1995.

[5] J. R. Swisher, S. H. Jacobson, and E. Yücesan, “Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey,” *ACM Transactions on Modeling and Computer Simulation*, vol. 13, pp. 134–154, 2003.

[6] S.-H. Kim and B. L. Nelson, “Selecting the best system: Theory and methods,” in *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, Eds., Piscataway, New Jersey, 2003, pp. 101–112.

[7] C.-H. Chen, J. Lin, E. Yücesan, and S. E. Chick, “Simulation budget allocation for further enhancing the efficiency of ordinal optimization,” *Discrete Event Dynamic Systems: Theory and Applications*, vol. 10, pp. 251–270, 2000.

[8] C. H. Chen and E. Yücesan, “An alternative simulation budget allocation scheme for efficient simulation,” *International Journal of Simulation and Process Modeling*, vol. 1, pp. 49–57, 2005.

[9] C.-H. Chen, D. He, M. Fu, and L. H. Lee, "Efficient simulation budget allocation for selecting an optimal subset," *INFORMS Journal on Computing*, vol. 20, no. 4, pp. 579–595, 2008.

[10] N. A. Pujowidianto, L. H. Lee, C.-H. Chen, and C. M. Yap, "Optimal computing budget allocation for constrained optimization," in *Proceedings of the 2009 Winter Simulation Conference*, M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, Eds., 2009, pp. 584–589.

[11] L. H. Lee, E. P. Chew, S. Teng, and D. Goldsman, "Optimal computing budget allocation for multi-objective simulation models," in *Proceedings of the 2004 Winter Simulation Conference*, R. G. Ingalls, M. D. Rosetti, J. S. Smith, and B. A. Peters, Eds. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc., 2004, pp. 586–594.

[12] M. C. Fu, J. H. Hu, C. H. Chen, and X. Xiong, "Simulation allocation for determining the best design in the presence of correlated sampling," *INFORMS Journal on Computing*, vol. 19, no. 1, pp. 101–111, 2007.

[13] D. He, S. E. Chick, and C.-H. Chen, "Opportunity cost and ocba selection procedures in ordinal optimization for a fixed number of alternative systems," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 37, no. 5, pp. 951–961, 2007.

[14] C.-H. Chen and L.-H. Lee, *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*. Hackensack, NJ: World Scientific, 2011.

[15] S. Andradóttir, "Simulation optimization," in *Handbook on Simulation*, J. Banks, Ed. New York, NY: John Wiley and Sons, 1998, pp. 307–333.

[16] M. C. Fu, "Optimization for simulation: Theory vs. practice," *INFORMS Journal on Computing*, vol. 14, pp. 192–215, 2002.

[17] J. R. Swisher, P. D. Hyden, S. H. Jacobson, and L. W. Schruben, "A survey of recent advances in discrete input parameter discrete-event simulation optimization," *IIE Transactions*, vol. 36, pp. 591–600, 2004.

[18] E. Tekin and I. Sabuncuoglu, "Simulation optimization: A comprehensive review on theory and applications," *IIE Transactions*, vol. 36, pp. 1067–1081, 2004.

[19] M. C. Fu, C.-H. Chen, and L. Shi, "Some topics for simulation optimization," in *Proceedings of the 2008 Winter Simulation Conference*, S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, Eds., 2008, pp. 27–38.

[20] A. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems of Information Transmission*, vol. 1, no. 1, pp. 1–7, 1965.

[21] R. J. Solomonoff, "A formal theory of inductive inference: Parts 1 and 2," *Information and Control*, vol. 7, no. 1-22 and 224-254, 1964.

[22] ——, "Complexity-based induction systems: Comparisons and convergence theorems," *IEEE Transactions on Information Theory*, vol. 24, pp. 422–432, 1978.

[23] L. A. Levin, "Universal sequential search problems," *Problems of Information Transmission*, vol. 9, pp. 265–266, 1973.

[24] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions Based On Algorithmic Probability*. Berlin Heidelberg: Springer, 2005.

[25] Q.-S. Jia and Q.-C. Zhao, "Strategy optimization for controlled markov process with descriptive complexity constraint," *Science in China Series F: Information Sciences*, vol. 52, no. 11, pp. 1993–2005, 2009.

[26] Q.-S. Jia, "On state aggregation to approximate complex value functions in large-scale markov decision processes," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 333–344, Feb. 2011.

[27] ——, "An adaptive sampling algorithm for simulation-based optimization with descriptive complexity constraints," in

Proceedings of the 1st IEEE Youth Conference on Information, Computing, and Telecommunications (YC-ICT2009), Beijing, China, 2009, pp. 118–121.

[28] ——, "An adaptive sampling algorithm for simulation-based optimization with descriptive complexity preference," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 4, pp. 720–731, 2011.

[29] S. Yan, E. Zhou, and C.-H. Chen, "Efficient simulation budget allocation for selecting the best set of simplest good enough designs," in *Proceedings of the 2010 Winter Simulation Conference*, B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, Eds., 2010, pp. 1152–1159.

[30] ——, "Efficient selection of a set of good enough designs with complexity preference," *IEEE Transactions on Automation Science and Engineering*, forthcoming.

[31] Y.-C. Ho, Q.-C. Zhao, and Q.-S. Jia, *Ordinal Optimization: Soft Optimization for Hard Problems*. New York, NY: Springer, 2007.

[32] C. H. Chen, "A lower bound for the correct subset-selection probability and its application to discrete event system simulations," *IEEE Transactions on Automatic Control*, vol. 41, pp. 1227–1231, 1996.

[33] M. H. DeGroot, *Optimal Statistical Decisions*. New York, NY: McGraw-Hill, 1970.

[34] R. C. Walker, *Introduction to Mathematical Programming*. Upper Saddle River, NJ: Prentice Hall, 1999.

Qing-Shan Jia received the B.E. degree in automation in July 2002 and the Ph.D. degree in control science and engineering in July 2006, both from Tsinghua University, Beijing, China. He is an Associate Professor at the Center for Intelligent and Networked Systems (CFINS), Department of Automation, TNLIST, Tsinghua University, Beijing, China. He was a Visiting Scholar at Harvard University in 2006, and a Visiting Scholar at the Hong Kong University of Science and Technology in 2010. His research interests include theories and applications of discrete event dynamic systems (DEDS's) and simulation-based performance evaluation and optimization of complex systems. He is a senior member of IEEE and serves as a co-guest editor and an associate editor of Discrete Event Dynamic Systems: Theory and Applications, co-guest editor and area editor of IIE Transactions, and an associate editor of IEEE Conference on Automation Science and Engineering.

Enlu Zhou received the B.S. degree with highest honors in electrical engineering from Zhejiang University, China, in 2004, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2009. Since then she has been an Assistant Professor at the Industrial & Enterprise Systems Engineering Department at the University of Illinois Urbana-Champaign. Her research interests include Markov decision processes, stochastic control, and simulation optimization. She is a recipient of the Best Theoretical Paper award at the 2009 Winter Simulation Conference and the 2012 AFOSR Young Investigator award.

Chun-Hung Chen received his Ph.D. degree in Engineering Sciences from Harvard University in 1994. He is a Professor of Systems Engineering & Operations Research at George Mason University and is also affiliated with National Taiwan University. Dr. Chen was an Assistant Professor of Systems Engineering at the University of Pennsylvania before joining GMU. Sponsored by NSF, NIH, DOE, NASA, MDA, and FAA, he has worked on the development of very efficient methodology for stochastic simulation optimization and its applications to air transportation system, semiconductor manufacturing, healthcare, security network, power grids, and missile defense system. Dr. Chen received “National Thousand Talents” Award from the central government of China in 2011, the Best Automation Paper Award from the 2003 IEEE International Conference on Robotics and Automation, 1994 Eliahu I. Jury Award from Harvard University, and the 1992 MasPar Parallel Computer Challenge Award. Dr. Chen has served as Co-Editor of the Proceedings of the 2002 Winter Simulation Conference and Program Co-Chair for 2007 Informs Simulation Society Workshop. He has served as a department editor for IIE Transactions, associate editor of IEEE Transactions on Automatic Control, area editor of Journal of Simulation Modeling Practice and Theory, associate editor of International Journal of Simulation and Process Modeling, and associate editor of IEEE Conference on Automation Science and Engineering.