

# Quantifying Heuristics in the Ordinal Optimization Framework

Zhen Shen · Qian-Chuan Zhao · Qing-Shan Jia

Received: 17 March 2008 / Accepted: 26 October 2009  
© Springer Science + Business Media, LLC 2009

**Abstract** Finding the optimal design for a discrete event dynamic system (DEDS) is in general difficult due to the large search space and the simulation-based performance evaluation. Various heuristics have been developed to find good designs. An important question is how to quantify the goodness of the heuristic designs. Inspired by the Ordinal Optimization, which has become an important tool for optimizing DEDS, we provide a method which can quantify the goodness of the design. By comparing with a set of designs that are uniformly sampled, we measure the ordinal performances of heuristic designs, i.e., we quantify the ranks of all (or some of) the heuristic designs among all the designs in the entire search space. The mathematical tool we use is the Hypothesis Testing, and the probability of making Type II error in the quantification is controlled to be under a very low level. The method can be used both when the performances of the designs can be accurately evaluated and when such performances are estimated by a crude but computationally easy model. The method can quantify both heuristics that output a single design and that output a set of designs. The method is demonstrated through numerical examples.

---

This work was supported by NSFC under Grants (60574067, 60704008, 60721003, 60736027, and 90924001), the Specialized Research Fund for the Doctoral Program of Higher Education (20070003110), the Program of Introducing Talents of Discipline to Universities (National 111 International Collaboration Project, B06002) and the high-level graduate student scholarship 2007 of China Scholarship Council. Part of the content of the paper was developed by Zhen Shen when he visited Boston University. Zhen Shen thanks Boston University for the hospitality. And, the content of this paper has been included in Zhen Shen's Ph.D. thesis at Tsinghua University.

---

Z. Shen (✉)

Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China  
e-mail: zhenshentu@gmail.com

Z. Shen · Q.-C. Zhao · Q.-S. Jia

Center for Intelligent and Networked Systems (CFINS), Department of Automation, TNLIST, Tsinghua University, Beijing, 100084, China

**Keywords** Discrete Event Dynamic Systems · Ordinal Optimization · Heuristic · Hypothesis Testing

## Key notations

Symbol	Meaning
$\Theta$	The search space
$\theta$	An element of the search space
$N$	Set of designs uniformly sampled from $\Theta$
$\theta_H$	A heuristic design
$\theta_{n\%}$	The design ranking exactly at top $n\%$ of $\Theta$
$\theta_{N,i}$	The $i$ -th design in $N$
$J(\cdot)$	The true performance of a design
$\hat{J}(\cdot)$	Observed performance of a design
$\hat{J}(\theta_{N,[t]})$	The $t$ -th order statistic of $\hat{J}(\theta_{N,i}), i = 1, 2, \dots,  N $
$\beta_0$	Bounding level for the probability of making the Type II error
$R_{\Theta}(\cdot)$	The rank of a design in $\Theta$

## 1 Introduction

Nowadays we have many complex man-made systems which are not easily described by differential equations. Examples of such systems include production and assembly lines, traffic systems, computer/communication networks, etc. These systems are called as Discrete Event Dynamic Systems (DEDS's). The evolution of DEDS depends on the complex interactions of timing of various discrete events as well as man-made rules of operation (Ho 1989). These systems are complex because they usually can be described only by computer simulation models. These systems are difficult to describe and optimize due to the uncertainty and the curse of dimensionality. In general, the optimization of DEDS can be modeled as follows,

$$\min_{\theta \in \Theta} J(\theta) = E[L(x(t; \theta, \xi))], \quad (1)$$

where  $\theta$  stands for the various system parameters that may subject to design choices,  $x(t; \theta, \xi)$  is the sample path obtained in simulation when the parameter is  $\theta$  and the randomness in the simulation is  $\xi$ ,  $L$  is the performance function defined on sample path and  $\Theta$  is the search space for the optimization variable  $\theta$ . Usually  $\Theta$  is a very large but finite set.

Heuristics are usually used to attack such analytically intractable problems. Heuristic, also called rule of thumb, refers to the method reasonably designed based on the human knowledge about the problem. Examples of the heuristic methods include the nearest neighbor method for Traveling Salesman Problem (TSP), Earliest Due Date first (EDD), Shortest Processing Time first (SPT), Shortest Setup Time first (SST) for Job Shop Problem (JSP) (Pinedo 2002), just to name a few. We also regard methods such as Genetic Algorithms (GA) (Holland 1975), Ant Colony Optimization (ACO) (Dorigo and Gambardella 1999; Dorigo et al. 1999), Particle Swarm Intelligence (PSO) (Kennedy and Eberhart 1995) as heuristics. All heuristic methods aim at finding good enough solutions. This is the main idea of soft computing. The

presence of problem information, expert knowledge, and human experience make soft computing easily accepted and applied to many practical problems. However, a common and natural question about all soft computing tools is: are designs found by the tool good enough? In this paper, we develop a systematic way to quantify the global ordinal performances of the designs found by the soft computing tools, including heuristics. In this paper, actually we do not care about the mechanism of the heuristics. The designs can be obtained using rules, search methods, algorithms, or the combination of several methods. As long as the problem can be modeled as Eq. 1, and the method outputs a design  $\theta$  or a set of designs from the search space  $\Theta$ , in principle we can apply the quantification method in this paper. We use the word “heuristics” as a representative of the large set of methods that can output designs from the search space. The basic idea is to compare the uniformly sampled designs of the search space with the heuristic design(s) in order to obtain the ordinal performance(s) of the heuristic design(s). This idea is inspired by the Ordinal Optimization (OO). OO takes out uniform samples from the search space to compose the sample space  $N$ . Then the samples are evaluated by a crude model and the top  $|S|$  ones are selected to compose the selected set  $S$ , where  $|S|$  represents the size of the set  $S$ . Here by “crude model” we refer to the model that is computationally easy but can give a performance estimate of a design. For example, we use a short simulation or a simulation with only a few replications. Different from many other soft computing tools, OO can guarantee that  $S$  contains top  $n\%$  (say 5%) designs of the search space with a high probability (say, no less than 95%). So, the goodness of the results of OO is quantified. In this paper, we quantify the performances of heuristic designs as follows. We compare the heuristic designs with the designs that are uniformly sampled. Then we use Hypothesis Testing (HT) method to test how many of the heuristic designs should be regarded as good enough. The probability of making the Type II error should be less than a given level (e.g., 0.05).

It should be noted that OO has been used together with other soft computing methods such as Simulated Annealing (SA) (Yen et al. 2004), Tabu Search (TS) (Mori and Tani 2003), and Nested Partitions (NP) (Shi and Ólafsson 2000). They do not provide a method to quantify the performances of heuristic designs in general. In this paper, we provide a method to quantify this performance in general. Our work in this paper is also different from OO. OO is an optimization technique. Given  $k$ , OO finds a set  $S$  that contains at least  $k$  good enough designs with a high probability. In this paper, for the given heuristic designs, our method quantifies the ordinal performances of these designs, with a small probability to make mistakes.

The rest of this paper is organized as follows. In Section 2 we briefly review OO and the hypothesis testing method. In Section 3 we formulate our problem in a Hypothesis Testing framework. In Section 4 we present the quantification method. In Section 5 we give two numerical examples. In Section 6 we conclude the paper.

## 2 Preliminary

### 2.1 Brief overview of Ordinal Optimization

Since the invention of OO in 1992 (Ho et al. 1992), there have been hundreds of papers related to OO (Shen et al. 2005). There are two tenets in OO. The first tenet

is ordinal comparison. Instead of using the accurate performances which presumably take a long time to obtain if the simulation model is used, one can use the relative order of noisy performance estimates as a basis for comparing and choosing designs. “Order” is easier to ascertain than “value”. The second tenet is goal softening. Instead of only caring about the single optimal design in the extremely large design space, which is improbable in the presence of large observation noise, one can pick a subset in which some “good enough” designs are guaranteed to be contained with a large probability. In OO, the top  $n\%$  (e.g., 5%) of the search space  $\Theta$  is defined as the good enough set of  $\Theta$ , and denoted as  $G_\Theta$ . OO is a two-stage optimization technique. At the first stage, usually OO uniformly samples a given number (say, 1,000) of designs from the search space  $\Theta$  to compose a sample space  $N$ . The top  $n\%$  of  $N$  is defined as the good enough set of  $N$  and denoted as  $G_N$ . At the second stage, OO uses a crude model to estimate the performances of the designs in  $N$  and selects the observed top  $|S|$  ones to compose the selected set  $S$ . OO has the virtue that it can guarantee that  $S$  contains at least  $k$  (called the alignment level in the jargon of OO) good enough designs of  $N$  with a high probability (say, no less than 0.95). In this way, OO obtains designs with quantifiable performances. In Lin and Ho (2002) and Shen et al. (2009a), it is justified that  $S$  found by OO can contain  $k^*$  ( $k^* \leq k$ ) good enough designs with a high probability, with  $k^*$  properly chosen. So, the global goodness, i.e., the goodness of  $S$  in the search space is also quantified. OO has two stages, at the first stage, OO samples from  $\Theta$  into  $N$  by the uniform sampling. At the second stage, OO obtains the selected set  $S$  from  $N$ . The first stage is easy to understand. Now we introduce how OO does in the second stage. For any design  $\theta$  of  $\Theta$ , the true performance is denoted by  $J(\theta)$ . By ordering  $J(\theta)$ , a non-decreasing curve can be obtained. The  $x$ -coordinate is the index of  $\theta$ , and the  $y$ -coordinate is  $J(\theta)$ . This curve is named as Ordered Performance Curve (OPC) and can reflect the internal relative relations among the designs. The OPC can be sorted into 5 types:

1. lots of good designs, Flat type
2. lots of good and lots of bad designs, but few intermediate ones, U-Shaped type
3. equally distributed good, bad and intermediate designs, Neutral type
4. lots of intermediate designs, but few good and bad designs, Bell type
5. lots of bad designs, Steep type

Categories 1 and 2 “represent problems where a good design will be relatively easy to find”, and category 5 and to a lesser extent category 4 “have a paucity of good designs”, as commented in Ho (1999). If the slope in category 5 becomes very sharp, it becomes the problem of “needle in the haystack”, which generally is difficult for any optimization technique. In Ho et al. (2007), the authors made comments that, “despite its (OO’s) many successes, OO is not at all useful for the needle-in-the-haystack type of problems where nothing but the best will do”. But, there are indeed many successful stories of OO. Please see Shen et al. (2005) for the references.

If we normalize the OPC into  $[0, 1] \times [0, 1]$  square, we obtain the standardized OPC. The standardized OPC can be specified by an incomplete beta function, with two shape parameters  $\alpha$  and  $\beta$  (Lau and Ho 1997). This OPC is the OPC of the whole search space and is denoted as  $OPC_\Theta$ . Similarly, we have the OPC of the sample space  $N$ . It is denoted as  $OPC_N$ .

There are many methods to obtain the selected set  $S$  (Ho et al. 2007). Two are popular. One is Blind Picking (BP), i.e.,  $S$  is uniformly sampled from  $N$ . Another is

Horse Racing. It is assumed that there is a crude but computationally easy model to evaluate the designs and then the computation burden shall be reduced. For example, we can use a short simulation or a simulation with only a few replications as a crude model. For design  $\theta$ , the estimated performance (also called “observed performance”) is denoted as  $\hat{J}(\theta)$ . The difference between  $\hat{J}(\theta)$  and  $J(\theta)$  is noise (or error), i.e.,

$$\hat{J}(\theta) = J(\theta) + W, \quad \theta \in N. \quad (2)$$

The noise  $W$  in Eq. 2 consists of both model noise and observation noise. Usually the noise is divided into three levels, i.e., the small noise, the medium noise and the large noise. The level could be determined by comparing the standard deviation of the noise with the value range of  $J(\theta)$ . We can order the designs in  $N$  according to the observed performance  $\hat{J}(\theta)$  from small to large. We can select out the best observed designs and then use them to constitute  $S$ . This is what “racing” in “Horse Racing” means. To measure the goodness of the selected set, the Alignment Probability (AP) is used,

$$P_A \equiv P\{|G_N \cap S| \geq k\}. \quad (3)$$

$P_A$  describes the probability that the number of truly (not estimated or observed) good enough designs in  $S$  is no smaller than  $k$ .  $k$  is called the alignment level.

The virtue of OO is that given the good enough set  $G_N$ , the size of sample  $N$  (usually 1,000), the noise level, the type of  $OPC_N$ , the selection rule, the alignment level  $k$  and the required alignment probability (e.g.,  $> 0.95$ ), we can know how large a selected set  $S$  should be. AP defined in Eq. 3 is also called the Universal Alignment Probability (UAP), since no specific knowledge about the problem is needed. This above work was mainly reported in (Lau and Ho 1997). Further, in the paper Lin and Ho (2002), the following definition is given,

$$P_A^* \equiv P\{|G_\Theta \cap S| \geq k^*\}. \quad (4)$$

This is the alignment probability between  $G_\Theta$  and  $S$ , and  $k^*$  is the alignment level between  $G_\Theta$  and  $S$ . What we are actually concerned with is  $P_A^*$ , not  $P_A$ . In Lin and Ho (2002), it was shown that, when we have properly set  $N$  (usually 1,000) and  $n\%$  (usually 5%), by properly selecting  $k^*$ , which should be equal to or slightly smaller than  $k$ , we can have a  $P_A^*$  very close to  $P_A$ . Thus, the good enough designs of  $N$  are probably the good enough designs of  $\Theta$ , i.e.,  $N$  can represent  $\Theta$ . In Shen et al. (2009a) we provide a new proof to show that  $N$  can represent  $\Theta$ .

In the standard OO, every design in the sample space  $N$  is allocated with the same computation budget to estimate the performance. In the papers (Chen et al. 2000a, b) the authors give an improving method called Optimal Computing Budget Allocation (OCBA). Concisely speaking, the improvement from OO to OCBA is that “instead of equally simulating all designs”, OCBA determines “the best numbers of simulation samples for each designs” (Chen et al. 2000a). The authors gave one possible definition of the probability of correct selection,  $P\{CS\}$ , as the probability that the observed best design is actually the best designs. It is a kind of alignment probability. And, to compute easily and quickly, the authors give the Approximate Probability of Correct Selection (APCS), which is a lower bound of  $P\{CS\}$ . Based on the above concepts, the authors provide the OCBA algorithm to allocate the computation budgets and show its advantage by numerical tests. And, in Chen et al.

(1999) the authors show how to use OCBA to compare different settings for a heuristic and then select the most promising ones. Different from OCBA which is based on the approximation *APCS*, we will use an alternative framework to quantify heuristic designs and hope new insights can be obtained.

## 2.2 Brief overview of Hypothesis Testing

Usually we are interested in whether a statement is correct (for example, the selected set contains at least  $k$  good enough designs) based on noised observation. We set the statement as the null hypothesis  $H_0$ . And usually, the opposite of  $H_0$  is set as  $H_1$  which is called the alternative hypothesis. The test is formed as a statement  $D_0$  constructed based on observation which contains noises. The negation of  $D_0$  is defined as  $D_1$ . Whenever we observe that  $D_0$  is true, we make the judgment to accept  $H_0$  (reject  $H_1$  at the same time). Whenever we observe  $D_1$  is true, we accept  $H_1$  (reject  $H_0$  at the same time). Thus, given the observation, four possibilities exist,

- (a)  $H_0$  is true, and is judged to be true;
- (b)  $H_0$  is true, but is judged to be false;
- (c)  $H_1$  is true, but is judged to be false;
- (d)  $H_1$  is true, and is judged to be true.

(a) and (d) are correct judgments while (b) and (c) are wrong. So, there are two types of errors,

Type I:  $H_0$  is true, but is judged to be false,  $P\{D_1|H_0\}$

Type II:  $H_1$  is true, but is judged to be false,  $P\{D_0|H_1\}$

Type I error means that  $H_0$  is true but we make the decision that we accept  $H_1$ .  $P\{D_1|H_0\}$  is a conditional probability meaning that  $H_0$  is true but we judge to accept  $H_1$ . Type II error means that  $H_1$  is true but we make the decision to accept  $H_0$ . One minus type II error probability is called the power of the test. As by the hypothesis testing method usually we cannot limit the probabilities of the two types of errors at the same time.<sup>1</sup>

## 3 Problem formulation

In order to quantify the ordinal quality of heuristics, we introduce a unified framework based on the two stages of OO. Without loss of generality, we view a heuristic  $H$  as a sampling mechanism having two stages. The first stage is to sample a subset  $N_H$  from the search space  $\Theta$ , the second stage is to select and output a set of designs  $S_H$  from  $N_H$ .  $S_H$  will also be called the selected set. We can also define the alignment probability (AP)

$$P_{A,H}^* \equiv P\{|G_\Theta \cap S_H| \geq k\} \quad (5)$$

for the heuristic  $H$ , where  $k$  is still called the alignment level. It is clear that the standard OO can be viewed as a special heuristic where the selected set  $S$  is generated

<sup>1</sup>Sequential Probability Ratio Test (SPRT) can limit the two types of errors at the same time. This can be a future topic following the work in this paper.

by BP or HR from a uniformly sampled subset  $N$  of  $\Theta$ . The introduction to BP and HR has been given in Section 2.1. Our description of heuristic is quite general. It allows the sophisticated process in each of the two stages. For example, the way to generate  $N_H$  could be an iterative process. In the Genetic Algorithms,  $N_H$  can be the union of populations of all generations. Similarly, the selected set  $S_H$  could also be chosen based on a set of complicated rules.

The quantification problem for the alignment probability  $P_{A,H}^*$  defined in Eq. 5 can then be formally stated as the problem to build the test  $D_0$  and its negation  $D_1$  such that for  $H_0 : |S_H \cap G_\Theta| \geq k$  and  $H_1 : |S_H \cap G_\Theta| < k$ , the type II error satisfies

$$P\{D_0|H_1\} \leq \beta_0$$

where we choose  $\beta_0 = 0.05$ . We choose to limit Type II error since it is more severe than Type I error. When Type I error happens,  $H_0$  is true but is judged to be false, that is, we underestimate the heuristic. However, when Type II error happens,  $H_1$  is true but is judged to be false, i.e., we wrongly accept  $H_0$ , which overrates the heuristic. Thus, Type II error is more severe than Type I error. We are more concerned with the Type II error.

## 4 Quantifying heuristics by Hypothesis Testing

A heuristic can output one design or multiple designs. This depends on how the heuristic is designed. When evaluating a design, there may be noise or there may not be. This depends on the problem and the crude model. Thus we have  $2 \times 2 = 4$  cases to consider: “No noise, one heuristic design”, “no noise, multiple heuristic designs”, “noise existing, one heuristic design” and “noise existing, multiple heuristic designs”. We will analyze them one by one.

### 4.1 When there is no noise

#### 4.1.1 No noise, one heuristic design

Firstly, we consider the easiest case, that is, there is no noise and the heuristic outputs only one design. We denote the heuristic design as  $\theta_H$ . And its rank in  $\Theta$  is denoted by  $R_\Theta(\theta_H)$ . The good enough set  $G_\Theta$  of  $\Theta$  is defined as the top  $n\%$  (e.g., 5%) of  $\Theta$ . We want to know whether  $\theta_H$  is a good enough design. In this case, we have  $N_H = S_H = \{\theta_H\}$ .

Since there is only one output, the alignment level  $k$  can only be 0 or 1. When  $k = 0$ , it is trivial. We are concerned with  $k = 1$ . We have

$$P_{A,H}^* \equiv P\{|G_\Theta \cap S_H| \geq 1\} = P\{R_\Theta(\theta_H) < n\% \times |\Theta|\}. \quad (6)$$

Here we use “ $<$ ” not “ $\leq$ ” which indicates that we choose top  $[0, n\%)$  of  $\Theta$  as the definition for the good enough set.

The null hypothesis and the alternative hypothesis are

$$H_0 : R_\Theta(\theta_H) < n\% \times |\Theta|; \quad H_1 : R_\Theta(\theta_H) \geq n\% \times |\Theta|. \quad (7)$$

Our idea is to compare the single design  $\theta_H$  with the set  $N$  which is obtained by uniform sampling from  $\Theta$  (as the first stage of OO). We denote the designs in  $N$  by

$\theta_{N,1}$  to  $\theta_{N,|N|}$  according to an arbitrary sequence. We use  $\theta_{N,[i]}$  to denote the  $i$ -th best design of the  $|N|$  designs, i.e., the design with the  $i$ -th smallest performance. Then  $J(\theta_{N,[i]})$  is the  $i$ -th order statistic.

Comparing  $\theta_H$  with the  $|N|$  designs, we obtain the rank of  $\theta_H$  in the  $|N|+1$  designs  $\{\theta_H\} \cup N$ . We take this rank as the observation, based on which we make judgments. The test  $D_0$  and its negation  $D_1$  are defined as

$$D_0 : J(\theta_H) < J(\theta_{N,[t]}); \quad D_1 : J(\theta_H) \geq J(\theta_{N,[t]}), \quad (8)$$

where  $t$  is a threshold to be determined based on the requirement on the probability of making the Type II error,

$$P\{D_0|H_1\} = P\{J(\theta_H) < J(\theta_{N,[t]}) | R_\Theta(\theta_H) \geq n\% \times |\Theta|\} \leq \beta_0. \quad (9)$$

We have

$$P\{J(\theta_H) < J(\theta_{N,[t]}) | R_\Theta(\theta_H) \geq n\% \times |\Theta|\} \leq P\{J(\theta_H) < J(\theta_{N,[t]}) | R_\Theta(\theta_H) = n\% \times |\Theta|\}. \quad (10)$$

This is because when  $R_\Theta(\theta_H) = n\% \times |\Theta|$ , the Type II error probability reaches the largest. Intuitively, the design with better rank in  $\Theta$  has higher probability to be observed better when comparing with designs in  $N$ . Concretely, for any instance of  $N$ , when  $R_\Theta(\theta_H) = n\% \times |\Theta|$ , compared with the case the rank of  $\theta_H$  worse than top  $n\%$ ,  $\theta_H$  has better (smaller) or at least the same performance, and then is easier to be observed better than the  $t$ -th best design in  $N$ . To control the Type II error probability no larger than  $\beta_0$ , we need and only need to control the largest Type II error probability no larger than  $\beta_0$ . To find  $t$ , we express the conditional probability in Eq. 10 as follows.

$$\begin{aligned} & P\{J(\theta_H) < J(\theta_{N,[t]}) | R_\Theta(\theta_H) = n\% \times |\Theta|\} \\ & \cong \sum_{i=0}^{t-1} \binom{|N|}{i} \left( \frac{n\%|\Theta|}{|\Theta|} \right)^i \left( \frac{|\Theta| - n\%|\Theta|}{|\Theta|} \right)^{|N|-i} \\ & = \sum_{i=0}^{t-1} \binom{|N|}{i} (n\%)^i (1 - n\%)^{|N|-i}. \end{aligned} \quad (11)$$

Here there is an approximation “ $\cong$ ”. We will give a detailed explanation about the approximation.  $\theta_H$  ranks at top  $n\% \times |\Theta|$  of the search space. But, there can be more than one designs having the same performance with  $\theta_H$ , we assume that the designs ranking from  $n\% \times |\Theta| - r_1$  to  $n\% \times |\Theta| + r_2$  ( $r_1, r_2 > 0$ ) all have the same performance as  $\theta_H$ . The necessary and sufficient condition for  $J(\theta_H) < J(\theta_{N,[t]})$  to hold is that there are at least  $|N| - t + 1$  designs with the ranks in  $\{n\%|\Theta| + r_2 + 1, n\%|\Theta| + r_2 + 2, \dots, |\Theta|\}$ , i.e., there are at most  $t - 1$  designs with the ranks in  $\{1, 2, \dots, n\%|\Theta| + r_2 - 1, n\%|\Theta| + r_2\}$ . So, we have

$$\begin{aligned} & P\{J(\theta_H) < J(\theta_{N,[t]}) | R_\Theta(\theta_H) = n\% \times |\Theta|\} \\ & = \sum_{i=0}^{t-1} \binom{|N|}{i} \left( \frac{n\%|\Theta| + r_2}{|\Theta|} \right)^i \left( \frac{|\Theta| - (n\%|\Theta| + r_2)}{|\Theta|} \right)^{|N|-i}. \end{aligned}$$

We can reasonably assume that  $r_2$  is very small compared with  $|\Theta|$ , so we can use Eq. 11 as the approximation.

**Table 1** The relationship between  $n\%$  and  $t$  when no noise, given  $\beta_0 = 0.05$ , and  $|N| = 1000$ 

$n\%$	1%	3%	5%	10%	20%	50%
$t$	5	21	39	85	179	474

The heuristic design is observed to be better than the  $t$ -th best design in  $N$  when and only when there are less than  $t$  designs sampled from the designs that are better than the heuristic design. Since the uniform sampling is used, the probability of sampling  $j$  ( $j < t$ ) designs from the designs that are better than the heuristic design obeys the binomial distribution, as is shown in Eq. 11. Based on Eq. 11, for  $\beta_0 = 0.05$ , we obtain  $t = 39$  for  $n\% = 5\%$ , and  $t = 85$  for  $n\% = 10\%$ . We list the relationship between  $t$  and  $n\%$  in Table 1.

When the heuristic design is observed to be better than the  $t$ -th design in the ordered  $|N|$  designs, we should judge the heuristic design to be within the top  $n\%$ , i.e., it is in the good enough set  $G_\Theta$ . In doing so, the probability of making Type II error is no larger than  $\beta_0$ .

#### 4.1.2 No noise, multiple heuristic designs

When the heuristic outputs multiple designs, the expression of  $P_{A,H}^*$  will not be as simple as for the single output. We assume that we have a set of heuristic designs. The set is denoted as  $N_H$ . We should find the selected set  $S_H$  and check its alignment with the good enough set  $G_\Theta$ . We order the designs in  $N_H$  by their performances from the best to the worst, and denote the designs as  $\theta_{H,1}, \theta_{H,2}, \dots, \theta_{H,|N_H|-1}, \theta_{H,|N_H|}$ . There is no noise and the evaluation is accurate. We will surely choose the top designs in  $N_H$  to constitute the selected set  $S_H$ , which are  $\theta_{H,1}, \theta_{H,2}, \dots, \theta_{H,|S_H|-1}, \theta_{H,|S_H|}$ . The  $P_{A,H}^*$  can be expressed as follows, given the alignment level  $k$ ,

$$P_{A,H}^* = P\{|S_H \cap G_\Theta| \geq k\}. \quad (12)$$

We denote two events,

$$E_1 = \{|S_H \cap G_\Theta| \geq k\}, \quad (13)$$

$$E_2 = \{R_\Theta(\theta_{H,k}) < n\% \times |\Theta|\}. \quad (14)$$

$E_1$  means the intersection of the selected set  $S_H$  and the good enough set  $G_\Theta$  is no smaller than  $k$ .  $E_2$  means the rank of the  $k$ -th design in  $S_H$  is better than the  $n\% \times |\Theta|$ -th design, i.e., it means the  $k$ -th design in  $S_H$  is a good enough design. It can be easily checked that

$$E_1 = E_2. \quad (15)$$

Then we have

$$P_{A,H}^* = P\{|S_H \cap G_\Theta| \geq k\} = P\{E_1\} = P\{E_2\} = P\{R_\Theta(\theta_{H,k}) < n\% \times |\Theta|\}. \quad (16)$$

The null and alternative hypothesizes for this case are

$$H_0 : R_\Theta(\theta_{H,k}) < n\% \times |\Theta|; \quad H_1 : R_\Theta(\theta_{H,k}) \geq n\% \times |\Theta|. \quad (17)$$

Simply speaking, we are concerned with whether the  $k$ -th design of  $N_H$  is a good enough design. Testing whether the  $k$ -th design of  $N_H$  is a good enough design has

no difference from testing whether a singleton heuristic output is a good enough design. We sample  $|N|$  designs by uniform sampling, and compare the  $k$ -th heuristic design with designs in  $N$ . We introduce the test

$$D_0 : J(\theta_{H,k}) < J(\theta_{N,[t]}); \quad D_1 : J(\theta_{H,k}) \geq J(\theta_{N,[t]}), \quad (18)$$

where  $t$  is the threshold to be determined such that the Type II error probability satisfies

$$P\{D_0|H_1\} = P\{J(\theta_{H,k}) < J(\theta_{N,[t]}) | R_\Theta(\theta_{H,k}) \geq n\% \times |\Theta|\} \leq \beta_0. \quad (19)$$

It is sufficient and necessary to have the following hold if we want Eq. 19 to hold,

$$P\{J(\theta_{H,k}) < J(\theta_{N,[t]}) | R_\Theta(\theta_{H,k}) = n\% \times |\Theta|\} \leq \beta_0. \quad (20)$$

By comparing Eq. 20 with Eq. 10 in Section 4.1.1, we know that the two expressions are the same except that we here are concerned with  $\theta_{H,k}$ . Thus the relationship between  $t$  and  $n\%$  here can also be shown by Eq. 11 and Table 1.

We summarize this section as follows. When the  $k$ -th of the ordered heuristic designs is better than the  $t$ -th of the ordered uniformly sampled designs, we should judge that there are at least  $k$  heuristic designs within the top  $n\%$  of the search space  $\Theta$ , i.e., they are in the good enough set  $G_\Theta$ . When making this judgment, the probability of making the Type II error is no larger than  $\beta_0$ .

## 4.2 When there is noise

### 4.2.1 Noise existing, one heuristic design

When there is no noise,  $OPC_\Theta$  or  $OPC_N$  do not affect the threshold  $t$  in Section 4.1.1. When calculating  $t$ , we only need to solve Eq. 11, which is the sum of binomial distribution expressions and this has nothing to do with  $OPC_\Theta$  or  $OPC_N$ . When applying this method to judge whether a design is within the top  $n\%$  of the search space  $\Theta$ , we only need to compare it with the  $t$ -th design of  $N$ , no matter what type  $OPC_\Theta$  or  $OPC_N$  is. However, when there is noise,  $OPC_\Theta$  has impact on quantifying the heuristic. This makes the problem more complex. We will explain in detail as follows.

The expression of  $P_{A,H}^*$  and the hypotheses are the same as the case that there is no noise. We still denote the heuristic design in  $\Theta$  as  $\theta_H$ .

$$P_{A,H}^* = P\{|S_H \cap G_\Theta| \geq 1\} = P\{R_\Theta(\theta_H) < n\% \times |\Theta|\}. \quad (21)$$

$$H_0 : R_\Theta(\theta_H) < n\% \times |\Theta|; \quad H_1 : R_\Theta(\theta_H) \geq n\% \times |\Theta|. \quad (22)$$

To find a test to limit the Type II error, we still compare the heuristic design  $\theta_H$  with designs in  $N$  which are denoted from  $\theta_{N,1}$  to  $\theta_{N,|N|}$ . We have

$$\begin{aligned} \hat{J}(\theta_H) &= J(\theta_H) + W_H, \\ \hat{J}(\theta_{N,i}) &= J(\theta_{N,i}) + W_{N,i}, \quad i = 1, 2, \dots, |N|, \end{aligned} \quad (23)$$

where  $W_H$  and  $W_{N,i}$  are noises. The noise is assumed to be I.I.D, and actually,  $J(\theta_{N,i})$  ( $i = 1, 2, \dots, |N|$ ) are also I.I.D. So here  $\hat{J}(\theta_{N,i})$  ( $i = 1, 2, \dots, |N|$ ) are also I.I.D. The

$i$ -th order statistic of  $\hat{J}(\theta_{N,i})$  ( $i = 1, 2, \dots, |N|$ ) is denoted as  $\hat{J}(\theta_{N,[i]})$ . We define the acceptance and rejection regions as below,

$$D_0 : \hat{J}(\theta_H) < \hat{J}(\theta_{N,[t]}); \quad D_1 : \hat{J}(\theta_H) \geq \hat{J}(\theta_{N,[t]}), \quad (24)$$

where  $t$  is the threshold chosen to limit the Type II error probability,

$$P\{D_0|H_1\} = P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[t]}) | R_\Theta(\theta_H) \geq n\% \times |\Theta|\} \leq \beta_0. \quad (25)$$

It is sufficient and necessary to require that

$$P\{D_0|H_1\} = P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[t]}) | R_\Theta(\theta_H) = n\% \times |\Theta|\} \leq \beta_0. \quad (26)$$

The relationship between Eqs. 25 and 26 is proved in Section A.1 of the Appendix.

To find  $t$ , we need to solve Eq. 26. Since we now have noise, expression of the left item of Eq. 26 is much more complex. To express it, we first give the following lemma.

**Lemma 1** *We assume that the observed performance of every design is the true performance plus a continuous I.I.D noise. The probability density function (p.d.f) of the noise is denoted as  $f_W(x)$  and the p.d.f of the observed performance of the heuristic design is denoted as  $f_H(x)$ . Under these conditions, when quantifying the heuristic with one singleton output, the expression of the Type II error probability is as follows,*

$$P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[t]}) | R_\Theta(\theta_H) = n\% \times |\Theta|\} = \int \int_{x_1 < x_2} f_H(x_1) f_{t:|N|}(x_2) dx_1 dx_2. \quad (27)$$

$f_{t:|N|}$  is the p.d.f of  $\hat{J}(\theta_{N,[t]})$ .

As obtaining the observed performance of the heuristic design is independent from obtaining the  $|N|$  designs by the uniform sampling, the joint p.d.f of  $\hat{J}(\theta_H)$  and  $\hat{J}(\theta_{N,[t]})$  is the product of their p.d.f's.

When there is no noise, we can easily calculate  $t$ . But when there is noise, it is almost impossible to calculate  $t$  directly. Solving Eq. 27 involves calculating the convolution, the distribution of the order statistics and the integration. Although it seems that we can obtain  $t$  by just solving the Eq. 27, it can be very difficult to get the closed-form expression for Eq. 27. It is not easy to solve Eq. 27 numerically, either. We try to use Monte-Carlo simulation to obtain  $t$ , as is shown below.

There is another problem when solving Eq. 27, that is, we use the  $OPC_\Theta$  which is the OPC of the whole design space  $\Theta$ .  $OPC_\Theta$  does not appear directly in Eq. 27. However, when calculating  $f_{t:|N|}(x)$  we need to use the distribution of  $\hat{J}(\theta_{N,i})$ , which is equal to  $J(\theta_{N,i}) + W_{N,i}$ .  $\theta_{N,i}$  is uniformly sampled from the search space and then the distribution of  $J(\theta_{N,i})$  is determined by  $OPC_\Theta$ .

It is almost impossible to obtain the  $OPC_\Theta$  of a real problem. To circumvent this difficulty, we can borrow the idea of “Lau and Ho (1997)”, by which they obtain Universal Alignment Probability (UAP). We may use the sampling information at the first stage to estimate the type of  $OPC_\Theta$ . In this way, we can know the type of  $OPC_\Theta$ . By taking samples in one type of  $OPC_\Theta$ , we will obtain different  $t$  for different samples. The most conservative (here the smallest)  $t$  will be taken as the threshold of this type of  $OPC_\Theta$ . The Monte-Carlo method is easy to implement. The

standardized  $OPC_{\Theta}$  can be specified by the incomplete beta function, with two shape parameters  $\alpha$  and  $\beta$  (see Lau and Ho 1997). We assume that the noise is uniformly distributed,

$$W_H, W_{N,i} \sim U[-L, L], \quad i = 1, 2, \dots, |N|, \quad (28)$$

where  $L$  is a positive number specifying how large the noise is. We set  $\alpha$  and  $\beta$  to typical numbers, e.g.,  $\alpha = 0.40$ ,  $\beta = 3.00$  specify a “Flat” type  $OPC_{\Theta}$ . And then we take the design at  $n\%$  as the heuristic design. We uniformly sample out  $|N|$  designs from the search space  $\Theta = [0, 1]$ . After adding the noises, we obtain the observed performances of the uniformly sampled designs. Then we check the rank of the heuristic design in the uniformly sampled designs by comparing their observed performances. After enough many replications, we find the  $t$  which satisfies Eq. 26 for the given level  $\beta_0$ .

In the traditional OO,  $U[-0.5, 0.5]$ ,  $U[-1.0, 1.0]$  and  $U[-2.5, 2.5]$  are used to represent small, medium and large noises (Lau and Ho 1997). Even the smallest of the three levels  $U[-0.5, 0.5]$  can result in swapping the good enough designs with the worst designs with non-zero probability. To investigate more, we do experiments for more noise levels. We define:

$$L = L_m = 0.05m. \quad (29)$$

Thus,  $U[-0.5, 0.5] = U[-L_{10}, L_{10}]$ ,  $U[-1.0, 1.0] = U[-L_{20}, L_{20}]$  and  $U[-2.5, 2.5] = U[-L_{50}, L_{50}]$ . We also do experiments for  $m = 0, 1, 2, 4, 8$  and 20,000. In our experiments,  $n\% = 10\%$ ,  $|N| = 1,000$  and the Type II error probability level  $\beta_0 = 0.05$ , and for each case, 25,000 replications are used. The results are shown in Table 2 (more numerical results can be found in Shen et al. 2009b).

Intuitively, the larger the noise is, the more difficult a good design is observed good, i.e., we should have a smaller  $t$  for larger noise. But the results of Table 2 do not coincide with this intuition. We provide the following explanation by firstly considering the two extreme cases: no noise and infinite noise. As we remember, when there is no noise, we have obtained from Table 1 that  $t = 85$ . We also verified this by experiments by setting the noise as 0. We perform a thought experiment for the case with infinite noise. When the noise is infinite, any design has the same probability to be better than another. The heuristic design has equal probability to be observed to have any rank. Since we are looking for the  $t$  so that the probability with which the heuristic design is observed to be better than the observed  $t$ -th best design of the set  $N$  generated by the uniform sampling is no larger than  $\beta_0$ , when the noise is infinite, we should have

$$t = \beta_0 \times |N| = 0.05 \times (1000 + 1) = 50.05. \quad (30)$$

**Table 2** The threshold  $t$ , obtained by Monte Carlo simulation

OPC		Noise level, $L_m$ means $U[-0.05m, 0.05m]$								
$(\alpha, \beta)$	Type	$L_0$	$L_1$	$L_2$	$L_4$	$L_8$	$L_{10}$	$L_{20}$	$L_{50}$	$L_{20000}$
(0.40, 3.00)	Flat	84	9	11	13	18	19	24	30	51
(0.40, 0.40)	U-Shaped	85	16	11	8	8	8	9	13	48
(1.00, 1.00)	Neutral	85	53	29	18	12	11	10	12	50
(3.00, 3.00)	Bell	85	59	38	22	14	13	12	16	51
(3.00, 0.40)	Steep	85	74	61	42	24	21	14	13	52

Experimentally, we set the noise as  $U[-1000, 1000] = U[-L_{20000}, L_{20000}]$  and our experiment result verifies this. The observed  $t$  is very near to 50.05. Thus, the above intuition is not correct. When the noise is very large, the heuristic designed has small opportunity to be observed with a high rank in the set  $N \cup \{\theta_H\}$ .

Now we explain why  $t$  goes smaller and then goes larger as the noise increases. For the problem here, the threshold  $t$  is affected by two factors. One is the noise. The other is the value difference between the true performances of the designs in the search space  $\Theta$ . It is specified by  $OPC_{\Theta}$ . When the noise is small, the value difference dominates. As the noise goes larger from 0, the disturbance caused by the noise becomes larger. When the disturbance is still not too large, we can use a smaller  $t$  to compensate, ensuring that the Type II error is not larger than the given level. But when the noise is too large, the noise dominates. As the noise goes larger, it is more and more difficult for the heuristic design to be observed within the top of the uniformly sampled designs, i.e., the Type II error probability goes smaller if  $t$  stays unchanged. It is obvious from Eq. 26 that as long as the noise and  $OPC_{\Theta}$  are given, as  $t$  goes larger, the probability of making Type II error goes larger. Since we only require the Type II error probability no larger than  $\beta_0$ , we can have a larger  $t$ . As the noise is infinite, the value difference does not matter at all. The  $t$  is the same with the value 50.05. In this way we explained the results of Table 2.

However, the  $t$  value is still affected by the value differences between the designs, or rather, the  $OPC_{\Theta}$ . The U-Shaped, Neutral and Bell types in Table 2 have  $t$  going smaller and then larger from  $L_1$  to  $L_{50}$ . For Flat type, the good enough designs are very “near” to each other when measuring their performances. A very small noise can flood the values. Even for the  $U[-L_1, L_1] = U[-0.05, 0.05]$ , we need  $t$  to be 9. It is the least robust to the noise among the examples of the 5 types in Table 2. For the Steep type, the good enough designs are with quite different values in performances and are not easily disturbed by the noise. It is the most robust to the noise.

In traditional OO, when we do not know how large the noise is, the largest noise can always serve as a conservative estimate. But here, since a smaller noise may correspond to a stricter (smaller)  $t$ , the large noise cannot serve as a conservative estimate. How to fix this is still an open question. This can be left for future research. Now we can ask a natural question: since we observe that  $t$  goes smaller and then larger as the noise goes larger from 0, does a lower bound of  $t$  exist?

The answer is yes. We have the following theorem. This theorem shows how good a design is in the ordinal sense when it is observed to be better than any of a given number of uniformly sampled designs obtained from the search space. It can be expressed as follows:

### Theorem 1 Assumptions:

1. The noise in evaluating a design is additive, i.e., the observed performance is the sum of the true performance and the noise.
2. The noise is I.I.D when we evaluate any design in the search space.
3. The noise has a continuous p.d.f.

**Conclusion:** We denote  $\Theta$  as the search space, and  $N$  as the set of uniform samples from  $\Theta$ . Its size is denoted by  $|N|$ . If a heuristic design, which is independent of the uniform samples, is observed to be better than any of the uniformly sampled designs in  $N$ , this heuristic design can be judged to be within the top  $n\%$  of the search space.

When doing so, the probability of making Type II error is no larger than a given level  $\beta_0$ . The value of  $n\%$  depends on  $|N|$  and  $\beta_0$  and can be expressed as follows,

$$n\% = \min_{0 < c < 1} \frac{1}{c \times \beta_0} \left\{ 1 - \exp \left\{ \frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|} \right\} \right\}. \quad (31)$$

Because

$$\min_{0 < c < 1} \frac{1}{c \times \beta_0} \left\{ 1 - \exp \left\{ \frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|} \right\} \right\} \leq \frac{1}{|N|} \min_{0 < c < 1} \frac{1}{c \times \beta_0} \ln \left( \frac{(1-c\beta_0)}{((1-c)\beta_0)} \right). \quad (32)$$

We can also judge

$$n\% = \frac{1}{|N|} \min_{0 < c < 1} \frac{1}{c \times \beta_0} \ln \left( \frac{(1-c\beta_0)}{((1-c)\beta_0)} \right). \quad (33)$$

In Eq. 33,  $n\%$  is inversely proportional to  $|N|$ . And given  $\beta_0 = 0.05$ , we can easily solve it by the numerical method that,  $\min_{0 < c < 1} \frac{1}{c \times \beta_0} \ln \left( \frac{(1-c\beta_0)}{((1-c)\beta_0)} \right)$  appears at the point very near to  $c = 0.826$ . If we take  $c = 0.826$ , from Eq. 33, we obtain

$$n\% = \frac{113.9}{|N|}, \quad \beta_0 = 0.05. \quad (34)$$

This serves as a quick formula for  $\beta_0 = 0.05$ .

We also provide a closed form for  $n\%$ . We have

$$\min_{0 < c < 1} \frac{1}{c \times \beta_0} \left\{ 1 - \exp \left\{ \frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|} \right\} \right\} \leq \frac{1}{\beta_0 |N|} \left( 1 + \sqrt{\ln \left( \frac{1}{\beta_0} \right)} \right)^2. \quad (35)$$

Thus, we can also judge  $n\%$  to be

$$n\% = \frac{1}{\beta_0 |N|} \left( 1 + \sqrt{\ln \left( \frac{1}{\beta_0} \right)} \right)^2. \quad (36)$$

The details of the proof are in the [Appendix](#). According to Theorem 1, we can show the relationship of  $|N|$  and  $n\%$  by a table, such as Table 3. The meaning of Theorem 1 lies in that no matter what noise it is, as long as the noise is additive, I.I.D and has a continuous p.d.f., if a heuristic design is observed to be better than a given number of uniform designs, we can judge this heuristic design to be within the top  $n\%$  of the search space with a small probability of making the error of overestimating the heuristic design.

**Table 3** The relation between  $n\%$  and  $|N|$ , given  $\beta_0 = 0.05$ , obtained from Eq. 31/Eq. 34

$ N $	500	600	700	800	900	1000	1500	2000	2500	3000
$n\%$	23%	19%	17%	15%	13%	12%	8%	6%	5%	4%

When using Theorem 1 to quantify heuristics, we can do in a sequential way. When we want to quantify a heuristic, we can first obtain a design by the heuristic, and then we sample out designs one by one by the uniform sampling, until one design is observed to be better than the heuristic design. The number of the uniformly sampled designs is denoted as  $|N|$  before the last uniformly sampled design is sampled out. Then, the heuristic design can be judged to be within the top  $n\%$  of the search space with the probability of making Type II error no larger than  $\beta_0$ .

#### 4.2.2 Noise existing, multiple heuristic designs

This is an extension to the case when the heuristic outputs a set of designs and there is no noise. It is also the most general case. Compared with the case of a set of designs output and no noise, we now cannot evaluate the designs accurately. Due to noise, that the observed  $k$ -th best design is good enough does not mean the designs observed better than it are also good enough. This causes difficulty in setting the hypothesis and finding the proper testing method. We still denote the selected set of the heuristic designs as  $S_H$ . These designs are sorted according to their observed performances from best (smallest) to worst (largest) and denoted as  $\theta_{H,1}, \theta_{H,2}, \dots, \theta_{H,|S_H|-1}, \theta_{H,|S_H|}$ .

We assume that the size of the selected set is given. We are concerned with the  $P_{A,H}^*$ ,

$$P_{A,H}^* = P\{|S_H \cap G_\Theta| \geq k\}. \quad (37)$$

We now have  $|S_H|$  designs. Each of them has a true performance. We assume we order them according to their true performance from best to worst, and denote them as  $\theta_{H,1}^{\text{Tr}}, \theta_{H,2}^{\text{Tr}}, \dots, \theta_{H,|S_H|-1}^{\text{Tr}}, \theta_{H,|S_H|}^{\text{Tr}}$ . We denote the following two events.

$$E_1 = \{|S_H \cap G_\Theta| \geq k\}, \quad (38)$$

$$E_2 = \{R_\Theta(\theta_{H,k}^{\text{Tr}}) < n\% \times |\Theta|\}. \quad (39)$$

$E_1$  means there are at least  $k$  designs in  $S_H$  are good enough designs. If  $E_1$  happens, the true top  $k$  designs of  $S_H$  must be good enough designs. Their ranks are better than  $n\% \times |\Theta|$ , i.e.,  $E_2$  happens. If  $E_2$  happens, the true top designs of  $S_H$  are good enough, there must be at least  $k$  designs within  $S_H$ , i.e.,  $E_1$  happens. We have

$$E_1 = E_2. \quad (40)$$

We set the hypotheses as follows,

$$H_0 : R_\Theta(\theta_{H,k}^{\text{Tr}}) < n\% \times |\Theta|; \quad H_1 : R_\Theta(\theta_{H,k}^{\text{Tr}}) \geq n\% \times |\Theta|. \quad (41)$$

The observed performance of  $\theta_{H,k}^{\text{Tr}}$  is  $\hat{J}(\theta_{H,k}^{\text{Tr}})$ . We have

$$\hat{J}(\theta_{H,k}^{\text{Tr}}) \leq \hat{J}(\theta_{H,|S_H|}^{\text{Tr}}), \quad (42)$$

since  $\hat{J}(\theta_{H,|S_H|}^{\text{Tr}})$  is the worst observed design of  $S_H$ . When testing the heuristic designs, we compare it with  $|N|$  designs by the uniform sampling. The better rank the heuristic design has in the uniformly sampled designs, the more confident we are to judge that there are at least  $k$  good enough designs. Thus we can still have

$$D_0 : \hat{J}(\theta_{H,k}^{\text{Tr}}) < \hat{J}(\theta_{N,[l]}) ; \quad D_1 : \hat{J}(\theta_{H,k}^{\text{Tr}}) \geq \hat{J}(\theta_{N,[l]}). \quad (43)$$

$\hat{J}(\theta_{N,[t]})$  is the  $t$ -th order statistic of  $\hat{J}(\theta_{N,i})$ ,  $i = 1, 2, \dots, |N|$ . We want to bound the probability of making the Type II error,

$$P\{D_0|H_1\} = P\{\hat{J}(\theta_{H,k}^{\text{Tr}}) < \hat{J}(\theta_{N,[t]}) | R_{\Theta}(\theta_{H,k}^{\text{Tr}}) \geq n\% \times |\Theta|\} \leq \beta_0. \quad (44)$$

We have solved Eq. 44 in Section 4.2.1, when we quantify the heuristic with one design output and there is noise. The result is summarized in Theorem 1.

Thus, if the  $\theta_{H,k}^{\text{Tr}}$  is observed to be better than  $\hat{J}(\theta_{N,[t]})$ , we should make the judgment that there are at least  $k$  good enough designs within  $S_H$ . If  $\theta_{H,k}^{\text{Tr}}$  is observed to be with the rank worse than  $t$ , we should make the judgment that there are less than  $k$  good designs within  $S_H$ .

But there is one problem. We do not know the observed performance of  $\theta_{H,k}^{\text{Tr}}$ . We only know  $\hat{J}(\theta_{H,k}^{\text{Tr}}) \leq \hat{J}(\theta_{H,|S_H|}^{\text{Tr}})$ , which is Eq. 42. We cannot solve this problem directly. However, we can use the following method: if  $\hat{J}(\theta_{H,|S_H|}^{\text{Tr}}) < \hat{J}(\theta_{N,[t]})$ , i.e.,  $\hat{J}(\theta_{H,|S_H|}^{\text{Tr}})$  is observed to be better than  $t$ -th of the ordered uniformly sampled designs, we can make the judgment that there are at least  $k$  good enough designs within  $S_H$ , since in this situation  $\theta_{H,k}^{\text{Tr}}$  must be observed better than the  $t$ -th of the ordered designs in  $N$ ; if  $\theta_{H,|S_H|}^{\text{Tr}}$  is observed not to be better than the  $t$ -th of the ordered designs in  $N$ , we will not make judgment.

**Corollary 1** *The selected set of the heuristic designs is denoted as  $S_H$ . The set of uniformly sampled designs from the search space  $\Theta$  is denoted as  $N$ . We evaluate the heuristic designs and the uniformly sampled designs by the crude model and obtain their observed performances.*

**Assumptions:**

1. *The noise in evaluating a design is additive, i.e., the observed performance is the sum of the true performance and the noise.*
2. *The noise is I.I.D when we evaluate any design in the search space.*
3. *The noise has a continuous p.d.f.*

**Conclusion:** *If the observed worst heuristic design in  $S_H$  still has a better observed performance than any of the uniform samples, all the designs in  $S_H$  should be judged to be within the top  $n\%$  of search space  $\Theta$ . In doing so, the probability of making Type II error is no larger than the given level  $\beta_0$ . The relationship between  $|N|$ ,  $n\%$  and  $\beta_0$  is shown as follows,*

$$n\% = \min_{0 < c < 1} \frac{1}{c \times \beta_0} \left\{ 1 - \exp \left\{ -\frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|} \right\} \right\}. \quad (45)$$

*This is the same with the formula given by Eq. 31 in Theorem 1. Thus it can also be expressed by Eq. 34, Eq. 36 or Table 3. The probability of making Type II error is expressed as*

$$\begin{aligned} P\{D_0|H_1\} &= P\{\hat{J}(\theta_{H,k}^{\text{Tr}}) < \hat{J}(\theta_{N,[t]}) | R_{\Theta}(\theta_{H,k}^{\text{Tr}}) \geq n\% \times |\Theta|\} \\ &= P\{\hat{J}(\theta_{H,k}^{\text{Tr}}) < \hat{J}(\theta_{N,[t]}) | |S_H \cap G_{\Theta}| < k\} \\ &\leq \beta_0, \end{aligned} \quad (46)$$

where  $k = |S_H|$ ,  $\theta_{H,k}^{\text{Tr}}$  is just  $\theta_{H,|S_H|}^{\text{Tr}}$ , the truly worst design in  $S_H$ .

As we have made clear how we obtain this conclusion, we will not give a formal proof here.

## 5 Examples

In this section, we show two examples, one is the Traveling Salesman Problem (TSP) and the other is the Flow Shop Problem (FSP). The purpose to show the examples are not to show how to solve the two problems in general but to verify the theoretical results of this paper.

### *Example 1* A 10-city Traveling Salesman Problem (TSP)

We take Traveling Salesman Problem (TSP) as an example. TSP can represent a large set of optimization problems and we show how to apply the results of this paper. We take the Hopfield 10-city problem (Hopfield and Tank 1985) as the example. The coordinates  $(x, y)$  of the 10 cities are:  $\{(0.4000, 0.4439); (0.2439, 0.1463); (0.1707, 0.2293); (0.2293, 0.7610); (0.5171, 0.9414); (0.8732, 0.6536); (0.6878, 0.5219); (0.8488, 0.3609); (0.6683, 0.2536); (0.6195, 0.2634)\}$ . We number the cities from 1 to 10 according to the above sequence. The best path is  $[1, 4, 5, 6, 7, 8, 9, 10, 2, 3]$  or  $[1, 3, 2, 10, 9, 8, 7, 6, 5, 4]$ . The two paths in fact specify the same round tour of visiting the 10 cities. The shortest round tour has the length 2.6907.

Now we use the nearest neighbor method, which starts from the 1<sup>st</sup> city (0.4000, 0.4439), and every time the nearest un-visited city is chosen as the next city. The obtained path is  $NN = [1, 10, 9, 8, 7, 6, 5, 4, 3, 2]$  with round tour 2.7782.

If when we choose the next city, all the unvisited cities have the same probability to be chosen, every design of search space has the same probability to be chosen. In this way, the obtained design is a uniform sample. We uniformly sample 1,000 designs from the search space, and in one experiment, the performance of the 39<sup>th</sup> best of the 1,000 uniform samples designs is 3.7990, thus the nearest neighbor design should be judged to be within the top 5% of the search space.

While keeping that the beginning city is always the 1<sup>st</sup> city, we now exchange the places of two cities in the nearest neighbor design. By doing so, we can obtain 36 designs. These 36 designs can be viewed as the designs output by a heuristic. We name this heuristic as “nearest neighbor method with exchange”, and use “NNE” for short. We still use the 1,000 uniformly sampled designs above. The 39<sup>th</sup> design has performance 3.7990. Since in the ordered 36 heuristic designs, the 21<sup>st</sup> design has performance 3.7365 and the 22<sup>nd</sup> design has performance 3.8711, we should judge 21 of the 36 designs to be within the top 5% of the search space.

In fact, there are  $(10 - 1)! = 362,880$  designs in the search space. We can enumerate all of them. The design exactly at top 5% is  $362,880 \times 5\% = 18,144$ . 18,144 and 18,143 have the same performance due to the symmetry of this TSP. Their paths are  $[1, 8, 7, 9, 10, 5, 4, 6, 2, 3]$  and  $[1, 3, 2, 6, 4, 5, 10, 9, 7, 8]$ . The performance of these two designs is 3.9141. Thus, this is the accurate border for top 5%. In the 36 heuristic designs, the 23<sup>rd</sup> best design has the performance 3.8858 and the 24<sup>th</sup> best has 3.9341. So in the 36 heuristic designs there are in fact 23 good enough designs.

And, the nearest neighbor design in fact ranks at 10 of the search space, thus it is top  $10/362,880 = 0.0028\%$ .

Now we assume there is  $U[0, 0.1]$  noise added to the distance between any two cities. This is not unreasonable since in the real life, due to the traffic and weather conditions, the time spent in traveling between two cities can be variable. We still try to quantify the nearest neighbor design, and the 36 designs obtained by exchanging two cities of the nearest neighbor designs. We generate 1,000 designs by the uniform sampling. The best observation of the 1,000 designs has performance 3.4380. And the nearest neighbor design is observed to have the performance 3.2513, thus from Table 3, given  $\beta_0 = 0.05$  this nearest neighbor design is judged to be within the top

$$n\% = 12\%. \quad (47)$$

We take the top 4 designs from the 36 designs to constitute the selected set  $S_H$ . In one experiment, even the worst observed design of the 4 designs still has an evaluated performance better than 3.4380. Thus, these 4 designs should also be judged to be within the top 12% of the search space. In fact, as we have discussed above, these 4 are within top 5%, and then are surely within top 12%.

#### Example 2 A 10 part Flow Shop Problem (FSP)

This example is taken from the OR library with the name “instance car5”.

Table 4 gives the processing times for the flow shop system. There are 10 parts and 6 machines. The machines are numbered from  $M_0$  to  $M_5$ . Each row in the table stands for a job. We index the jobs from 1 to 10.

We assume that all jobs go through the six machines in the same order. This kind of FSP is called permutation flow shops. And we assume the buffers are all large enough so that we do not need to consider the blocking. The decision is to decide the service sequence of the jobs. There are 10 jobs, and then the size of the search space is  $10! = 3,628,800$ . We use the total completion time as the measurement. We consider the random searching algorithm. For this algorithm, every design has the equal probability to be considered. We start from randomly taking a job as the first one, and then we select other jobs with equal probability, and so on. In one experiment, we considered 2000 designs, and denoted the best of them, which is with the service sequence [3 2 5 6 4 1 8 7 10 9], as  $\theta_H$  (with total completion time 51960). We can enumerate the search space and find out that, the best sequence is

**Table 4** Carrier 10x6 instance

	$M_0$	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
Job <sub>1</sub>	333	991	996	123	145	234
Job <sub>2</sub>	333	111	663	456	785	532
Job <sub>3</sub>	252	222	222	789	214	586
Job <sub>4</sub>	222	204	114	876	752	532
Job <sub>5</sub>	255	477	123	543	143	142
Job <sub>6</sub>	555	566	456	210	698	573
Job <sub>7</sub>	558	899	789	124	532	12
Job <sub>8</sub>	888	965	876	537	145	14
Job <sub>9</sub>	889	588	543	854	247	527
Job <sub>10</sub>	999	889	210	632	451	856

$\theta_0 = [4\ 5\ 6\ 3\ 2\ 1\ 10\ 9\ 7\ 8]$  with total completion time 50545. And  $\theta_H$  ranks at 299, by percentage, it is,  $299/10! = 0.0082\%$ . Now we quantify the design  $\theta_H$ . We take out 20,000 uniform samples. In one experiment, we find out that the design  $\theta_H$  ranks at the 4-th in  $N \cup \{\theta_H\}$ . So by the method in Section 4.1, we can judge the design  $\theta_H$  to be within top 0.0184% of the search space.

We assume there is a noise added to the evaluation of each design. The noise is assumed to obey  $N(0, 100^2)$ . The best performance is 50545, and the worst performance is 80973 with the corresponding sequence  $[10\ 8\ 7\ 1\ 9\ 4\ 2\ 6\ 3\ 5]$ . The difference between the worst and the best is 30428. Considering this, we know how large the noise is. In one experiment, we still consider the design  $\theta_H = [3\ 2\ 5\ 6\ 4\ 1\ 8\ 7\ 10\ 9]$ . In one observation its performance is 52134. We take the sequential sampling method, and in one experiment, we did not obtain an observed performance better than or equal to 52134 until the 7481<sup>st</sup> design. So we know the size of the  $N$  is 7480. And by Theorem 1, we have,  $n\% = 113.9/7480 \times 100\% = 1.52\%$ . The judgment is obviously correct as the true rank of  $\theta_H$  is 0.0082%. Considering the noise issue, the quantification result can be seen as good.

It should be made clear that this example is only for proof of concept purpose. In practice, one usually does not want to obtain good designs by random samples which is very inefficient. Problem related knowledge should be used.

## 6 Conclusions

By the OO idea, we define the top  $n\%$  (say, 5%) of search space as the good enough set. In this paper we provide a method to judge whether the heuristic designs are good enough or not in terms of ranking in the search space. We compare the heuristic design(s) with uniformly sampled designs and by Hypothesis Testing technique, the probability of making Type II error when making the judgment can be bounded under a given level. In principle our method can apply no matter how many designs the heuristic outputs and no matter whether there is noise or not. For the case there is noise, if we know the type of the Ordered Performance Curve of the search space and the noise level, we can improve our method. But how to achieve this is still an open question and should be investigated. We show by a TSP and an FSP how to use our method. We aim to provide more practical applications of our method in the future.

About the computation efficiency of the quantifying method in this paper, generally, we know that we pay more so that we can obtain more. We do not deny that we need to pay computation cost in order to obtain a good prediction on how good heuristic design(s) are. One virtue of our method is that, no matter for the no noise case or for the noisy case, we can make a budget on the computation, and we will stop until our computation resource is exhausted or we have obtained a satisfactory result. Moreover, about the noisy case, we want to emphasize that as we use the crude model for the evaluation, generally speaking, there will be no heavy burden for evaluation process.

In this paper, what we quantify is actually heuristic design(s). For a given problem and given problem instance, every time a heuristic may output different designs, i.e., the heuristic contains randomness itself. This is common, as GA, SA, ACO etc. are all of this kind. We argue that for a heuristic containing randomness itself, we must

run it many times, in principle we can quantify its design(s) obtained in any run, and then we know how good the heuristic is. Without running it many times, we have no idea about its internal randomness.

The quantified heuristic can improve OO since the heuristic can obtain more good enough designs from the search space than the uniform sampling does. If we have some knowledge about  $\Theta$ , as we usually do in practice, heuristics may be used to replace the uniform sampling to generate samples from  $\Theta$ . And usually a heuristic can outperform the uniform sampling, that is, the set of samples obtained by the heuristic contains more good enough designs. In other words, we should evaluate the Alignment Probability (AP) under the new setting where the heuristic is used to generate samples at the first stage of OO. We denote the sample space as  $N_H$  and the selected set as  $S_H$  in the new OO.  $S_H$  can be obtained from  $N_H$  based on some selection rule such as Blind Picking (BP) or Horse Racing (HR). The AP for the new OO is defined as

$$P_{A,H}^* = P\{|G_\Theta \cap S_H| \geq k\}.$$

Usually we expect

$$|S_H| \leq |S|,$$

which means, by choosing a smaller selected set we still have the same effect, if the heuristic is used. This is one topic for the future efforts.

The quantification method in this paper could be improved from the following two aspects. For OO and our method here, there are two kinds of randomness. One is the sampling randomness from the search space  $\Theta$  to the sample space  $N$ . The other is the randomness when evaluating designs with a crude model. We here use the uniform sampling when sampling from  $\Theta$  to  $N$ . In OO, the uniform samples can represent the search space. In quantifying heuristics, the uniform samples provide a basis to measure heuristics. One concern is whether we can use other better sampling technique instead of the uniform sampling. For OO, if we can use a better sampling technique, the intensity of good enough designs in  $N$  can be increased, as discussed above. For quantifying heuristics, a better  $N$  can be a “higher” basis to quantify heuristics. Actually, a well-performing heuristic with randomness can be viewed as an “enhanced sampling” technique, as it can provide designs from the search space, and the designs are usually better than the uniform samples. It is interesting and useful to replace the uniform sampling with the non-uniform sampling. This will be investigated in our following research. Another possible improvement is in evaluating the designs. In OO and our method of quantifying heuristics, the computation budget is equally allocated to the designs. Optimal Computation Budget Allocation (OCBA) makes improvement to OO by allocating more computation budget to the promising designs. It may be possible to improve our quantification method by introducing OCBA. This can also be a topic for future research.

**Acknowledgement** The authors would like to express their sincere thanks to Prof. Yu-Chi Ho of Harvard University & Tsinghua University. He initializes and guides the research in this paper. But of course, all remaining possible errors are solely the responsibility of the authors.

# Appendix A Proof of Theorem 1

Please refer to Section 4.2.1 for the presentation of Theorem 1. We will first prove the fundamental Eq. 31 of Theorem 1 in Section A.5 based on the results established in Sections A.1–A.4. Other results in Theorem 1 are all proved in Section A.6.

## A.1 Maximum of the probability of making Type II error

In this section we will prove

$$\begin{aligned} P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]}) | R_\Theta(\theta_H) \geq n\% \times |\Theta|\} \\ \leq P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]}) | R_\Theta(\theta_H) = n\% \times |\Theta|\}. \end{aligned} \quad (48)$$

We introduce the indicator function:

$$I_E = \begin{cases} 1, & \text{when } E \text{ happens,} \\ 0, & \text{otherwise.} \end{cases} \quad (49)$$

We have the following,

$$P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]})\} = EI_{\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]})\}}. \quad (50)$$

This holds no matter which design  $\theta_H$  is. By the total expectation formula, we have

$$EI_{\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]})\}} = E\left(E\left(I_{\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x\right)\right). \quad (51)$$

We have

$$E\left(I_{\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x\right) = E\left(I_{\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x\right). \quad (52)$$

Since  $\hat{J}(\theta_H) = J(\theta_H) + W_H$ , from Eq. 52,

$$\begin{aligned} E(I_{\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x) &= E(I_{\{J(\theta_H) + W_H < x\}} | \hat{J}(\theta_{N,[l]}) = x) \\ &= E(I_{\{W_H < x - J(\theta_H)\}} | \hat{J}(\theta_{N,[l]}) = x). \end{aligned} \quad (53)$$

Please pay attention to Eq. 53, it is a conditional probability, for a given  $x$ , when  $J(\theta_H)$  becomes smaller,  $x - J(\theta_H)$  becomes larger, the event  $W_H < x - J(\theta_H)$  is more likely to happen.

If we have two designs,  $\theta_{H,1}$  and  $\theta_{H,2}$ ,  $J(\theta_{H,1}) \leq J(\theta_{H,2})$ , from Eq. 53 and the above analysis, we have

$$\begin{aligned} E(I_{\{\hat{J}(\theta_{H,1}) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x) &= E(I_{\{W_{H,1} < x - J(\theta_{H,1})\}} | \hat{J}(\theta_{N,[l]}) = x) \\ &\geq E(I_{\{W_{H,2} < x - J(\theta_{H,2})\}} | \hat{J}(\theta_{N,[l]}) = x) \\ &= E(I_{\{\hat{J}(\theta_{H,2}) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x). \end{aligned} \quad (54)$$

where  $W_{H,1}$  and  $W_{H,2}$  are the noises when evaluating  $\theta_{H,1}$  and  $\theta_{H,2}$ . They are I.I.D. Eq. 54 tells us, when  $J(\theta_{H,1}) \leq J(\theta_{H,2})$ ,

$$E(I_{\{\hat{J}(\theta_{H,1}) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x) \geq E(I_{\{\hat{J}(\theta_{H,2}) < \hat{J}(\theta_{N,[l]})\}} | \hat{J}(\theta_{N,[l]}) = x). \quad (55)$$

From Eqs. 50, 51 and 55, we have, when  $J(\theta_{H,1}) \leq J(\theta_{H,2})$ ,

$$P\{\hat{J}(\theta_{H,1}) < \hat{J}(\theta_{N,[t]})\} \geq P\{\hat{J}(\theta_{H,2}) < \hat{J}(\theta_{N,[t]})\}. \quad (56)$$

Equation 56 tells us that, the smaller the true performance of the heuristic design  $\theta_H$  is, the higher is the probability that it is observed to be better than the  $t$ -th best design in  $N$ . Let us denote  $\theta_{n\%}$  as the design whose true rank in  $\Theta$  is exactly top  $n\%$  of  $\Theta$ . Thus, we have

$$\begin{aligned} P\{D_0|H_1\} &= P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[t]}) | R_\Theta(\theta_H) \geq n\% \times |\Theta|\} \\ &\leq P\{\hat{J}(\theta_H) < \hat{J}(\theta_{N,[t]}) | R_\Theta(\theta_H) = n\% \times |\Theta|\} \\ &= P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[t]})\}. \end{aligned} \quad (57)$$

This is because,  $\theta_{n\%}$  is the best design of the designs with ranks not smaller than  $n\%$ .

Thus, in order to limit the probability of making Type II error, we need and only need to consider the case when the probability of making Type II error is the largest. When  $t = 1$ , from Eq. 57,

$$P\{D_0|H_1\} \leq P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \leq \beta_0. \quad (58)$$

## A.2 An upper bound for $P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\}$

We will prove

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \leq \int_{-\infty}^{\infty} (1 - n\% F_W(x))^{|N|} f_W(x) dx, \quad (59)$$

where  $f_W(x)$  and  $F_W(x)$  are the p.d.f and the c.d.f of the noise  $W$  respectively. We have  $\hat{J}(\theta_{n\%}) = J(\theta_{n\%}) + W$ .  $W$  is the noise when evaluating this design. We have

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} = P\{J(\theta_{n\%}) + W < \hat{J}(\theta_{N,[1]})\} \quad (60)$$

Then,

$$P\{J(\theta_{n\%}) + W < \hat{J}(\theta_{N,[1]})\} = \int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,[1]})\} f_W(x) dx. \quad (61)$$

If  $\theta_{n\%}$  is observed better than  $\hat{J}(\theta_{N,[1]})$ , it means that this design is observed better than any of the designs in  $N$ , i.e., it is observed better than  $\hat{J}(\theta_{N,i}) = J(\theta_{N,i}) + W_{N,i}$ ,  $i = 1, 2, \dots, |N|$ . Since any  $\theta_{N,i}$  is uniformly sampled from the search space and  $W_{N,i}$  is the I.I.D noise,  $\hat{J}(\theta_{N,i})$  is I.I.D. Thus Eq. 61 is transformed into

$$\begin{aligned} &\int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,[1]})\} f_W(x) dx \\ &= \int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,1}), \dots, J(\theta_{n\%}) + x < \hat{J}(\theta_{N,|N|})\} f_W(x) dx \\ &= \int_{-\infty}^{\infty} \prod_{i=1}^{|N|} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,i})\} f_W(x) dx \\ &= \int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,i})\}^{|N|} f_W(x) dx, \quad \forall i. \end{aligned} \quad (62)$$

The last “=” in Eq. 62 holds because  $\hat{J}(\theta_{N,i})$  is I.I.D. Now let us focus on Eq. 62, for any  $i$ , we have

$$P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,i})\} = P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i}\}. \quad (63)$$

By the total probability formula, we have

$$\begin{aligned} & P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i}\} \\ &= P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i} | W_{N,i} < x\} P\{W_{N,i} < x\} \\ & \quad + P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i} | W_{N,i} \geq x\} P\{W_{N,i} \geq x\}. \end{aligned} \quad (64)$$

In Eq. 64, we have,

$$\begin{aligned} & P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i} | W_{N,i} < x\} \\ &= P\{J(\theta_{n\%}) - J(\theta_{N,i}) < W_{N,i} - x | W_{N,i} < x\} \\ &\leq P\{J(\theta_{n\%}) - J(\theta_{N,i}) < 0 | W_{N,i} < x\} \\ &= P\{J(\theta_{n\%}) - J(\theta_{N,i}) < 0\}. \end{aligned} \quad (65)$$

The last “=” in Eq. 65 holds because the noise is independent from the true performance of a uniformly sampled design. For Eq. 65, we have

$$P\{J(\theta_{n\%}) - J(\theta_{N,i}) < 0\} \leq 1 - n\%. \quad (66)$$

It is “ $\leq 1 - n\%$ ” not “ $= 1 - n\%$ ” in Eq. 66 because  $OPC_\Theta$  may not be strictly increasing. For a strictly increasing  $OPC_\Theta$ , in (B30) “ $\leq 1 - n\%$ ” should be replaced by “ $= 1 - n\%$ ”. This is because for strictly increasing  $OPC_\Theta$ , every design has a different performance. The heuristic design with rank at  $n\%$  is better than a uniformly sampled design when and only when the uniformly sampled design is sampled from designs with worse ranks than the heuristic design. For the not strictly increasing  $OPC_\Theta$ , the ranks are still from 1 to  $|\Theta|$ . Ties are broken arbitrarily and the designs with ties are given different but sequential ranks. Thus, there can be some designs having the same performance with  $\theta_{n\%}$  but with worse ranks than  $\theta_{n\%}$ . This is the reason why it is “ $\leq 1 - n\%$ ” in Eq. 66.

From Eqs. 65 and 66, we know Eq. 64 can be changed to

$$\begin{aligned} & P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i}\} \\ &= P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i} | W_{N,i} < x\} P\{W_{N,i} < x\} \\ & \quad + P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i} | W_{N,i} \geq x\} P\{W_{N,i} \geq x\} \\ &\leq (1 - n\%) P\{W_{N,i} < x\} + P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i} | W_{N,i} \geq x\} P\{W_{N,i} \geq x\}. \end{aligned} \quad (67)$$

Since any conditional probability cannot be larger than 1, we have

$$P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i} | W_{N,i} \geq x\} \leq 1. \quad (68)$$

$F_W(x)$  is used to stand for the the c.d.f of the I.I.D noise. By Eqs. 67 and 68, we have

$$\begin{aligned} & P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i}\} \\ &\leq (1 - n\%) P\{W_{N,i} < x\} + P\{W_{N,i} \geq x\} \\ &= (1 - n\%) F_W(x) + 1 - F_W(x) \\ &= 1 - n\% F_W(x). \end{aligned} \quad (69)$$

From Eqs. 62, 63 and 69, we have

$$\begin{aligned} & \int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,[1]})\} f_W(x) dx \\ &= \int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,i})\}^{|\mathcal{N}|} f_W(x) dx \text{ (due to (62))} \\ &= \int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < J(\theta_{N,i}) + W_{N,i}\}^{|\mathcal{N}|} f_W(x) dx \text{ (due to (63))} \\ &\leq \int_{-\infty}^{\infty} (1 - n\% F_W(x))^{|\mathcal{N}|} f_W(x) dx \text{ (due to (69))}. \end{aligned} \quad (70)$$

As we can see, we have proved

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \leq \int_{-\infty}^{\infty} (1 - n\% F_W(x))^{|N|} f_W(x) dx. \quad (71)$$

### A.3 An upper bound of $\int_{-\infty}^{\infty} (1 - n\% F_W(x))^{|N|} f_W(x) dx$

In this section, we shall prove that the right hand side (RHS) of Eq. 71 is no larger than  $\min_{p \in [0,1]} \{p + (1 - n\% p)^{|N|} (1 - p)\}$ , that is,

$$\begin{aligned} P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} &\leq \int_{-\infty}^{\infty} (1 - n\% F_W(x))^{|N|} f_W(x) dx \\ &\leq \min_{p \in [0,1]} \{p + (1 - n\% p)^{|N|} (1 - p)\}. \end{aligned}$$

We rewrite Eq. 70 as follows, i.e., the integration is divided into two parts, separating by  $x_0$ .  $x_0$  can be any real number,

$$\begin{aligned} &\int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,[1]})\} f_W(x) dx \\ &\leq \int_{-\infty}^{\infty} (1 - n\% F_W(x))^{|N|} f_W(x) dx \\ &= \int_{-\infty}^{x_0} (1 - n\% F_W(x))^{|N|} f_W(x) dx + \int_{x_0}^{\infty} (1 - n\% F_W(x))^{|N|} f_W(x) dx. \end{aligned} \quad (72)$$

For the first item in the RHS of Eq. 72, we have

$$F_W(x) \geq 0, \quad x \in (-\infty, x_0). \quad (73)$$

For the second item in the RHS of Eq. 72, we have

$$F_W(x) \geq F_W(x_0), \quad x \in [x_0, +\infty). \quad (74)$$

From Eqs. 73 and 74, Eq. 72 can be changed to

$$\begin{aligned} &\int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,[1]})\} f_W(x) dx \\ &\leq \int_{-\infty}^{\infty} (1 - n\% F_W(x))^{|N|} f_W(x) dx \\ &\leq \int_{-\infty}^{x_0} f_W(x) dx + \int_{x_0}^{\infty} (1 - n\% F_W(x_0))^{|N|} f_W(x) dx \\ &= \int_{-\infty}^{x_0} f_W(x) dx + (1 - n\% F_W(x_0))^{|N|} \int_{x_0}^{\infty} f_W(x) dx \\ &= F_W(x_0) + (1 - n\% F_W(x_0))^{|N|} \times (1 - F_W(x_0)). \end{aligned} \quad (75)$$

Please recall that we are concerned with

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} = \int_{-\infty}^{\infty} P\{J(\theta_{n\%}) + x < \hat{J}(\theta_{N,[1]})\} f_W(x) dx. \quad (76)$$

Thus, from Eqs. 71 and 75, finally Eq. 76 can be bounded by

$$\begin{aligned} &P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \\ &\leq F_W(x_0) + (1 - n\% F_W(x_0))^{|N|} \times (1 - F_W(x_0)), \quad \forall x_0 \in (-\infty, \infty). \end{aligned} \quad (77)$$

Please notice that, for any  $x_0$ , Eq. 77 holds. Thus, for any  $p \in [0, 1]$ , we can derive from (B40) that

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \leq p + (1 - n\% p)^{|N|} \times (1 - p). \quad (78)$$

Thus, we have

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \leq \min_{p \in [0,1]} \{p + (1 - n\% p)^{|N|} (1 - p)\}. \quad (79)$$

Thus, we now change our problem to the problem of optimizing Eq. 79. If we can find the minimum of the expression shown in Eq. 79, we find the upper bound of the probability of making Type II error.

#### A.4 The convexity of the bound in the RHS of Eq. 79

In this section, we shall show  $p + (1 - n\%p)^{|N|}(1 - p)$  is a convex function of  $p$  in  $[0, 1]$ . We denote,

$$h(p) = p + (1 - n\%p)^{|N|}(1 - p), \quad p \in [0, 1]. \quad (80)$$

We notice,

$$\begin{aligned} h(0) &= 1, \\ h(1) &= 1, \\ h(p) &= p + (1 - n\%p)^{|N|}(1 - p) < p + 1 \times (1 - p), \quad p \in (0, 1). \end{aligned} \quad (81)$$

Thus, there must be a minimum and it appears within  $(0, 1)$ . Next, we prove the convexity of  $h(p)$  over  $(0, 1)$ .

For  $|N| \geq 2$ , we have

$$\frac{dh(p)}{dp} = 1 + (1 - n\%p)^{|N|-1}(-|N|n\% + |N|n\% \times p - 1 + n\% \times p), \quad (82)$$

$$\begin{aligned} \frac{d^2h(p)}{dp^2} &= (|N| - 1)(1 - n\%p)^{|N|-2}(-n\%)\{-|N|n\% + |N|n\% \times p - 1 + n\% \times p\} \\ &\quad + (1 - n\%p)^{|N|-1} \times (|N|n\% + n\%). \end{aligned} \quad (83)$$

For  $N = 1$ , we have

$$\frac{dh(p)}{dp} = 1 + (-|N|n\% + |N|n\% \times p - 1 + n\% \times p) = n\% \times (2p - 1), \quad (84)$$

$$\frac{d^2h(p)}{dp^2} = 2n\% > 0. \quad (85)$$

Please see Eq. 83. The second item in the RHS is obviously larger than 0. And, in the “{ }”,

$$-|N|n\% + |N|n\% \times p - 1 + n\% \times p = -|N|n\% \times (1 - p) - (1 - n\% \times p) < 0. \quad (86)$$

Thus, the first item in the RHS of Eq. 83 is also larger than 0. Then we have for any integer  $|N|$ ,

$$\frac{d^2h(p)}{dp^2} > 0, \quad p \in [0, 1]. \quad (87)$$

We finish proving the convexity of  $p + (1 - n\%p)^{|N|}(1 - p)$  for  $p \in [0, 1]$ .

A.5 Solution to  $\min_{p \in [0,1]} \{p + (1 - n\%p)^{|N|}(1 - p)\}$

In Section A.3, we have proved Eq. 79, which is

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \leq \min_{p \in [0,1]} \{p + (1 - n\%p)^{|N|}(1 - p)\}.$$

Thus, if

$$\min_{p \in [0,1]} \{p + (1 - n\%p)^{|N|}(1 - p)\} \leq \beta_0$$

happens, the following is guaranteed,

$$P\{\hat{J}(\theta_{n\%}) < \hat{J}(\theta_{N,[1]})\} \leq \beta_0.$$

We know from Section A.4, there is one and only one minimum of  $h(p)$ ,  $p \in [0, 1]$ , and this minimum appears within  $(0,1)$ .

We should solve Eq. 82, which is  $\frac{dh(p)}{dp} = 0$  to obtain the  $p$  corresponding to the minimum. But it is difficult to obtain a closed form of  $p$  by Eq. 82.

There are at least two methods to address this difficulty. The first method is numerical calculation. Since we have Eq. 87, which guarantees the convexity, the gradient method can work very well. The second method is as follows.

What we want is  $\min_{p \in [0,1]} \{p + (1 - n\%p)^{|N|}(1 - p)\} \leq \beta_0$ . To make it hold, we only need to find a  $p \in [0, 1]$  such that the following hold,

$$p + (1 - n\%p)^{|N|}(1 - p) \leq \beta_0. \quad (88)$$

For a coefficient  $c$ ,  $0 < c < 1$ ,

$$p = c \times \beta_0. \quad (89)$$

Since  $p = c \times \beta_0 \in (0, 1)$ , we can substitute Eq. 89 into Eq. 88, we have

$$c \times \beta_0 + (1 - n\%c \times \beta_0)^{|N|}(1 - c \times \beta_0) \leq \beta_0. \quad (90)$$

From Eq. 90, we obtain

$$n\% \geq \frac{1}{c \times \beta_0} \left\{ 1 - \exp \left\{ \frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|} \right\} \right\}, \quad \forall c, 0 < c < 1. \quad (91)$$

Equation 91 means, if a heuristic design is observed to be better than any design in  $N$ , the heuristic design can be judged be within the top  $n\% = \min_{0 < c < 1} \frac{1}{c \times \beta_0} \left\{ 1 - \exp \left\{ \frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|} \right\} \right\}$  of the search space  $\Theta$ , no matter what kind and how large the noise is. In doing so, the probability of the Type II error is not larger than the given level  $\beta_0$ . We finish the proof of Eq. 31. Solving Eq. 91 by numerical method, we obtain Table 3.

## A.6 The quick formula and the closed form for $n\%$

In this section, we shall prove that,

$$n\% = \frac{113.9}{|N|}, \beta_0 = 0.05.$$

This is Eq. 34 in Theorem 1. And we will also prove,

$$\min_{0 < c < 1} \frac{1}{c \times \beta_0} \left\{ 1 - \exp \left\{ \frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|} \right\} \right\} \leq \frac{1}{\beta_0 |N|} \left( 1 + \sqrt{\ln \left( \frac{1}{\beta_0} \right)} \right)^2.$$

If the above is true, the heuristic design can be judged to be within top  $n\%$  with the following expression,

$$n\% = \frac{1}{\beta_0 |N|} \left( 1 + \sqrt{\ln \left( \frac{1}{\beta_0} \right)} \right)^2.$$

This is Eq. 36 in Theorem 1. Before we give the proof, we give a lemma.

**Lemma 2**  $1 - \exp(x) \leq -x$  for any real  $x$ . And by simple transformations, we have,  $\ln(1+x) \leq x$ ,  $x > -1$ .

*Proof* Let  $f(x) = \exp(x) - 1 - x$ . We have

$$\frac{df(x)}{dx} = \exp(x) - 1, \quad \frac{d^2 f(x)}{dx^2} = \exp(x) > 0.$$

Thus,  $f(x)$  is convex. The only minimum appears at  $\frac{df(x)}{dx} = \exp(x) - 1 = 0$ , solve this, we obtain  $x = 0$ . Thus, the minimum of  $f(x)$  is,  $f(0) = \exp(0) - 1 - 0 = 0$ . We then have

$$f(x) = \exp(x) - 1 - x \geq 0.$$

Then,  $\exp(x) \geq 1 + x$ . Taking logarithm at both sides, we have,  $\ln(1+x) \leq x$ ,  $x > -1$ .  $\square$

Apply Lemma 2 to Eq. 91. For  $x = \frac{\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{|N|}$  we can easily establish

$$\min_{0 < c < 1} \frac{1}{c \times \beta_0} \{1 - \exp(x)\} \leq \min_{0 < c < 1} \frac{1}{c \times \beta_0} \{-x\} = \min_{0 < c < 1} \frac{-\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{c \times \beta_0 |N|}. \quad (92)$$

So, we can judge  $n\%$  to be

$$n\% = \frac{1}{|N|} \min_{0 < c < 1} \frac{-\ln \left( \frac{(1-c)\beta_0}{1-c\beta_0} \right)}{c \times \beta_0}. \quad (93)$$

Given  $\beta_0$ , by the numerical method, we can find the  $c$  that makes  $\min_{0 < c < 1} \frac{-\ln\left(\frac{(1-c)\beta_0}{1-c\beta_0}\right)}{c \times \beta_0}$  achieve its minimum. We find out that when  $\beta_0 = 0.05$ , the  $c$  that makes  $\min_{0 < c < 1} \frac{-\ln\left(\frac{(1-c)\beta_0}{1-c\beta_0}\right)}{c \times \beta_0}$  achieve minimum is very near to  $c = 0.826$ . When  $c = 0.826$ , from Eq. 93, we have

$$n\% = \frac{113.9}{|N|}, \quad \beta_0 = 0.05. \quad (94)$$

To go further, we pay attention to the numerator of the RHS of Eq. 92,

$$-\ln\left(\frac{(1-c)\beta_0}{1-c\beta_0}\right) = \ln\left(\frac{1-c\beta_0}{(1-c)\beta_0}\right). \quad (95)$$

We have

$$0 < \frac{1-c\beta_0}{(1-c)\beta_0} \leq \frac{1}{(1-c)\beta_0}. \quad (96)$$

From Eq. 96, Eq. 95 can be changed to

$$\begin{aligned} \ln\left(\frac{1-c\beta_0}{(1-c)\beta_0}\right) &\leq \ln\left(\frac{1}{(1-c)\beta_0}\right) \\ &= \ln\left(\frac{1}{\beta_0}\right) + \ln\left(\frac{1}{1-c}\right) = \ln\left(\frac{1}{\beta_0}\right) + \ln\left(1 + \frac{c}{1-c}\right). \end{aligned} \quad (97)$$

By Lemma 2,  $\ln(1+x) \leq x$ ,  $x > -1$ , we have for the second item of the RHS of Eq. 97

$$\ln\left(1 + \frac{c}{1-c}\right) \leq \frac{c}{1-c}. \quad (98)$$

From Eqs. 92, 95, 97 and 98, we have

$$\begin{aligned} \min_{0 < c < 1} \frac{1}{c \times \beta_0} \left\{ 1 - \exp\left\{ -\frac{\ln\left(\frac{(1-c)\beta_0}{1-c\beta_0}\right)}{|N|} \right\} \right\} &\leq \min_{0 < c < 1} \frac{1}{c \times \beta_0 |N|} \left( \ln\left(\frac{1}{\beta_0}\right) + \frac{c}{1-c} \right) \\ &= \min_{0 < c < 1} \frac{1}{\beta_0 |N|} \left( \frac{\ln\left(\frac{1}{\beta_0}\right)}{c} + \frac{1}{1-c} \right). \end{aligned} \quad (99)$$

We denote

$$f(c) = \frac{\ln\left(\frac{1}{\beta_0}\right)}{c} + \frac{1}{1-c}. \quad (100)$$

We want to find the minimum of this function. We have

$$\frac{df(c)}{dc} = \frac{-\ln\left(\frac{1}{\beta_0}\right)}{c^2} + \frac{1}{(1-c)^2}, \quad (101)$$

$$\frac{d^2 f(c)}{dc^2} = \frac{2\ln\left(\frac{1}{\beta_0}\right)}{c^3} + \frac{2}{(1-c)^3} > 0. \quad (102)$$

Thus,  $f(c)$  is convex for  $0 < c < 1$ . This is one and only one minimum for  $0 < c < 1$ . The minimum appears when

$$\frac{df(c)}{dc} = 0. \quad (103)$$

Solve it, we obtain

$$c = \frac{\sqrt{\ln\left(\frac{1}{\beta_0}\right)}}{1 + \sqrt{\ln\left(\frac{1}{\beta_0}\right)}}. \quad (104)$$

Substitute this  $c$  into  $f(c)$ , we obtain

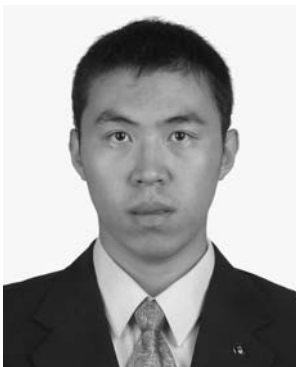
$$\min_{0 < c < 1} f(c) = \left(1 + \sqrt{\ln\left(\frac{1}{\beta_0}\right)}\right)^2, \quad (105)$$

based on which Eq. 36 can be easily derived.

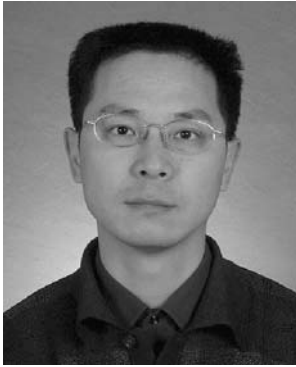
## References

- Chen CH, Wu SD, Dai L (1999) Ordinal comparison of heuristic algorithms using stochastic optimization. *IEEE Trans Robot Autom* 15(1):44–56
- Chen CH, Lin J, Yücesan E, Chick SE (2000a) Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discret Event Dyn Syst Theor Appl* 10:251–270
- Chen HC, Chen CH, Yücesan E (2000b) Computing efforts allocation for ordinal optimization and discrete event simulation. *IEEE Trans Automat Contr* 45(5):960–964
- Dorigo M, Gambardella LM (1999) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 1(1):53–66
- Dorigo M, Di Caro G, Gambardella LM (1999) Ant algorithms for discrete optimization. *Artif Life* 5(2):137–172
- Ho YC (1989) Introduction to special issue on dynamics of dynamics of discrete event systems. *Proc IEEE* 77(1):3–6
- Ho YC (1999) An explanation of ordinal optimization: soft computing for hard problems. *Inf Sci* 113:169–192
- Ho YC, Sreenivas R, Vakili P (1992) Ordinal optimization of discrete event dynamic systems. *J DEDS* 2(2):61–88
- Ho YC, Zhao QC, Jia QS (2007) Ordinal optimization: soft optimization for hard problems. Springer, New York
- Holland JH (1975) Adaptation in natural and artificial systems. University Michigan Press, Ann Arbor

- Hopfield JJ, Tank DW (1985) “Neural” computation of decisions in optimization problems. *Biol Cybern* 52:141–152
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE int. conf. on neural networks, 1942–1948, Perth, Australia
- Lau TWE, Ho YC (1997) Universal alignment probabilities and subset selection for ordinal optimization. *J Optim Theory Appl* 93(3):455–489
- Lin SY, Ho YC (2002) Universal alignment probability revisited. *J Optim Theory Appl* 113(2):399–407
- Mori H, Tani H (2003) A hybrid method of PTS and ordinal optimization for distribution system service restoration. In: IEEE international conference on systems, man and cybernetics, vol 4, pp 3476–3483
- Pinedo M (2002) Scheduling theory, algorithms, and systems, 2nd edn. Prentice-Hall, Englewood Cliffs
- Shen Z, Bai H-X, Zhao YJ (2005) Ordinal optimization references list. <http://cfins.au.tsinghua.edu.cn/en/resource/index.php>
- Shen Z, Zhao QC, Jia QS, Sun J (2009a) Universal alignment probability revisited. *J Optim Theory Appl* 141:371–376
- Shen Z, Zhao QC, Jia QS (2009b) Quantifying heuristics in the ordinal optimization framework. Technique report, Center for Intelligent and Networked Systems. [http://www.cfins.au.tsinghua.edu.cn/files/paper/Quantify\\_JDEDS09\\_full.pdf](http://www.cfins.au.tsinghua.edu.cn/files/paper/Quantify_JDEDS09_full.pdf)
- Shi LY, Chen CH (2000) A new algorithm for stochastic discrete resource allocation optimization. *Discret Event Dyn Syst Theor Appl* 10:271–294
- Shi LY, Ólafsson S (2000) Nested partitions method for global optimization. *Oper Res* 48(3):390–407
- Yen CH, Shan D, Wong H, Jang SS (2004) Solution of trim-loss problem by an integrated simulated annealing and ordinal optimization approach. *J Intell Manuf* 15:701–709



**Zhen Shen** received the B.E. degree and the Ph.D. degree in 2004 and 2009 respectively from Department of Automation, Tsinghua University, Beijing, China. Currently, he is an assistant professor at Institute of Automation, Chinese Academy of Sciences. He was a visiting scholar at Department of Manufacturing Engineering and Center for Information and Systems Engineering, Boston University, MA, USA, from Oct. 2007 to Apr. 2008. He specializes in the area of the discrete event dynamic systems (DEDS) theory and applications, and the optimization of complex systems.



**Qian-Chuan Zhao** received the B.E. degree in automatic control in July 1992, the B.S. degree in applied mathematics in July 1992, and the Ph.D. degree in control theory and its applications in July 1996, all from Tsinghua University, Beijing, China. He is currently a Professor and Associate Director of the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua University. He was a Visiting Scholar at Carnegie Mellon University, Pittsburgh, PA, and Harvard University, Cambridge, MA, in 2000 and 2002, respectively. He was a Visiting Professor at Cornell University, Ithaca, NY, in 2006. His research interests include modeling, control and optimization for networked systems and applications. Dr. Zhao is an associate editor for the Journal of Optimization Theory and Applications and an associate editor for IEEE Transactions on Automation Science and Engineering. He is a Senior Member of IEEE.



**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 has been a lecturer in CFINS, Department of Automation, Tsinghua University, since August 2006. He was a visiting scholar in the Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA, from August 2006 to February 2007. His research interests include performance evaluation and optimization of complex systems and stochastic optimization. He is a member of IEEE.