JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 93, No. 3, pp. 455-489, JUNE 1997

Universal Alignment Probabilities and Subset Selection for Ordinal Optimization[1]

T. W. EDWARD LAU[2] AND Y. C. HO[3]

Abstract. We examine in this paper the subset selection procedure in the context of ordinal optimization introduced in Ref. 1. Major concepts including goal softening, selection subset, alignment probability, and ordered performance curve are formally introduced. A two-parameter model is devised to calculate alignment probabilities for a wide range of cases using two different selection rules: blind pick and horse race. Our major result includes the suggestion of quantifiable subset selection sizes which are universally applicable to many simulation and modeling problems, as demonstrated by the examples in this paper.

Key Words. Subset selection, stochastic optimization, alignment probability, ordered performance curve, simulation, modeling.

1. Introduction

Consider a standard optimization problem,

$$\min_{\theta \in \Theta} J(\theta), \tag{1}$$

where Θ is the design space and $J(\cdot)$ is a performance measure defined on the design space. By design space we mean a collection of alternatives or designs available to a designer. For example, a design space can be a subset of the Euclidean space, and designs are simply points in the subset (there are infinitely many of them). As another example, in the well-known travelling salesman problem (TSP), the design space is the collection of all possible routes formed by all cities [total number of routes is $(n-1)!$ for an n-city

[1]The work in this paper was partially supported by NSF Grants EID-9212122, EEC-9402384, Army Contracts DAAL03-92-G0115, DAAH04-95-0148, and Air Force Contract F49620-95-1-0131.

[2]Postdoctoral fellow, Division of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts.

[3]Professor in Engineering and Applied Mathematics, Division of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts.

0022-3239/97/0600-0455$12.50/0

problem]. As one more example, in a decision problem, the design space can be the strategy space formed by all possible strategies.[4] A performance measure, on the other hand, is usually a scalar function of the designs in the design space.[5] It can be deterministic in nature, such as the tour length of a route in the TSP, or it can be the expected value of a random quantity, such as the average sojourn time of a customer in a queueing system. In the latter case, $J(\theta)$ in (1) can be written as

$$J(\theta)=E[L(X(\theta, \omega))], \tag{2}$$

where ω represents all randomness or noise and $X(\theta, \omega)$ is a trajectory for design θ generated according to ω. We refer to problems in the form of Eq. (2) as stochastic optimization,[6] which is our main focus in this paper. Although many interesting methods have successfully been developed, and references are widely available, most solution techniques, as we attempt to contrast in this paper, serve the objective of finding the optimizer (if possible) or at least some near-optimal solutions. Yet, the problem remains very challenging, analytically as well as computationally, owing to at least three kinds of difficulties. First, the design space could be horrendously immense in many complicated problems. For instance, combinatorial explosion of the design space due to increasing problem size is a classical topic in complexity theory (e.g., NP-completeness). Second, the design space could have very little structure or even no structure for the efficient use of existing techniques. This can be related to a proper representation of the search space for efficient search. Third, performance evaluations may be corrupted by very large noise. Noise could be part of the inherent random nature of the system, which makes the performance measure a random quantity. Usually, a large number of replications is required in order to obtain sound statistical results. For example, many lengthy simulations may be needed so as to bring down the variance of any performance estimates, and consequently they inflate the computing budget incredibly. These three kinds of difficulties are commonly found in designing systems such as production facilities, communication networks, and other human-made systems collectively known as discrete event dynamic systems (DEDS).

In this paper, we will explain a broader viewpoint for the search problem using a framework called ordinal optimization (Ref. 2). We submit that two

[4]The maximum number of strategies is the cardinality of the decision space (e.g., controls) raised to the power of the cardinality of the information space (e.g., states).

[5]We do not consider multicriteria optimization in this paper.

[6]Some authors use the term "stochastic optimization" for iterative procedures, such as simulated annealing, which involve generation of successive solutions by means of chance mechanism. This is different from what we consider here, in the sense that randomness is an inherent feature of the design problem.

concepts are fundamental, not only in this paper, but indeed to all optimization problems: the good enough subset and the selected subset. Informally, a good enough subset is a subset of the search space in which the members satisfy some design criteria set forth by the designer. Oftentimes, a good enough subset is easy to specify but difficult to obtain.[7] On the other hand, the selected subset is a subset of Θ in which the members are picked out by the designer using certain evaluation scheme (algorithms, heuristics, crystal ball gazing, and so on) as the outcome for the design process. Every optimization problem can, in principle, be conceived as the goal of matching a selected subset with the good enough subset. As a matter of fact, most traditional ways of describing an optimization problem ask for the good enough subset and the selected subset both to be singleton, i.e., to pick one design (selected singleton) with target at the true optimum (good enough singleton) in the design space.

To illustrate this further, consider again the minimization problem (1). Suppose that Θ is the line segment $[-10, 10]$ on the real line and that

$$J(\theta) = \theta^2, \qquad \theta \in [-10, 10].$$

We would normally ask for the point(s) in Θ corresponding to the true optimum of $J(\cdot)$ as member(s) of the good enough subset. In this case, we know easily that the good enough subset (in fact, the global minimum) is located exactly at $\theta = 0$. Using the gradient descent method or other means, the algorithm may also output $\theta = 0$ as the selected subset, a singleton $\{0\}$ in this case. For this particular example, the good enough subset happens to be identical to the selected subset. Nevertheless, most design problems are so complicated that expecting such perfect match between the good enough subset and the selected subset may be too optimistic. For example, if $J(\cdot)$ has many local minima, then the gradient descent method could easily trap the selected subset in a local minimum other than the true optimum; therefore, it will never be matched with the good enough subset. The situation is made even worse if performance evaluations and other auxiliary entities such as gradients, Hessians, etc. (if they exist) are subject to large noise. Meanwhile, although many interesting techniques were developed in order to circumvent this problem [see, for example, Ahlswede (Ref. 3), Zhigljavsky (Ref. 4)], should the designer insist on getting singletons for both the good enough subset and the selected subset (of the true optimum), mismatches between the two sets will likely be inevitable. An appropriate analogy drawn here would be that of "hitting one speeding bullet with another."

[7]We use the term "good enough," because any suboptimal solution indeed carries the flavor of "good enough". Momentarily, a more precise interpretation of "good enough" will be given.

In ordinal optimization, proposed by Ho et al. (Ref. 1), a more liberal point of view is taken for both the good enough subset and the selected subset. Briefly speaking, the basic idea is the softening of the membership criteria for the good enough subset and the selected subset, while maintaining reasonable matching outcomes with efficiency and confidence. For example, the membership criterion for the good enough subset can be chosen as the top n-percentile of the design space. We refer to this as goal softening. The merit of taking such a viewpoint is that of the economy of computation. To see this, suppose we can obtain a set of representative samples from the design space, in the sense that the samples are drawn from a big urn of all designs, each with equal probability. Then, softening n from $n=0.005$ to $n=5$ can provide a reduction factor of 1,000 for the sample size.[8]

Meanwhile, we consider picking not only one but a set of designs to form the selected subset from the set of representative samples. Notice that, in the absence of noise, we essentially need to pick only the smallest evaluated design for the selected subset. When the performance estimate is noisy, as it is in this paper, it becomes necessary to include more than one design in order to secure with higher confidence a certain degree of matching, or alignment, between the selected subset and the good enough subset. The degree of matching is called the alignment level, and the confidence of achieving a certain alignment level is referred to as the alignment probability. Both concepts will be made precise in the subsequent section. The goal of this paper is to determine the appropriate size of a selected subset under appropriate conditions where the desired level of alignment and alignment probability are given.

At this point, alert readers may raise a legitimate concern that the top $n\%$ of a very large search space is still a large portion. We hasten to supply an explanation. It should be emphasized that it is the top $n\%$ of the designs that ordinal optimization is concerned about, but not the top $n\%$ of the performance values. There can be a big difference in some problems. However, value is never the intent of our approach, and the very connotation "ordinal" precludes anything that one can say about cardinal notions. We should also emphasize that the goal of ordinal optimization is not to replace but to complement many existing techniques for optimization and search problems. Our approach helps in speeding the process of narrowing down potentially promising and manageable subsets of designs on which one can lavish further attention using other means, including traditional ones. This is a crucial step during the initial phases in many search problems that involve: (i) search space which is immense and even structureless; and (ii)

[8]This can be seen by considering the probability of obtaining at least one good enough design of $n\%$ in a total of N samples, which is given by $1-(1-n\%)^N$.

performance evaluation which is corrupted by large noise. A difficult problem, such as that of a needle in a haystack, will always be hard without sufficient knowledge or clue to the solution. Getting within the top $n\%$ of such problem may not be very satisfactory (1% of a 10^{10} search space is 10^8th from the optimum). However, because of the efficient narrowing down process promised by ordinal optimization, a designer can learn from a one-step application of ordinal optimization the properties of good solutions, such as which portion of the design space is to be explored next, or what a better representation of the problem should be. Both the paper by Ruml et al. (Ref. 5) and the paper by Deng and Ho (Ref. 6) illustrate this point. In the latter, they have concluded from successful application of ordinal optimization that

$$\text{new search representation (or region)}$$
$$= f(\text{old search representations (or regions)}),$$

in the sense of traditional hill climbing. Determining the function $f(\cdot)$ in the above step is a rich and barely explored subject which is outside the scope of the paper. Adaptive search schemes such as genetic algorithms or other man-machine interactions in many artificial intelligence based methods are strongly implied. On top of these, as demonstrated by Vakili et al. (Ref. 7) and Patsis et al. (Ref. 8), the generation and simulation of the representative samples is also an area where modern massively parallel computing technologies can be exploited.

In this paper, we investigate an important step involved in the implementation of ordinal optimization procedures: the subset selection problem. Although similar problems are discussed in the statistics literature, usually called ranking and selection procedures [see the extensive references by Gupta and Panchapakesan (Ref. 9), Santner and Tamhane (Ref. 10), and also a recent survey by Goldman and Nelson (Ref. 11)], the problem that we are dealing with here has two major differences. First, the ranking and selection procedures deal with selection from a set of samples which has size of usually less than a hundred, while we consider subset selection from a set of representative samples of size in the thousands or more. Second, the notions of softened criterion and ordinal comparison are not represented in the ranking and selection procedures. For example, one important quantity used in these procedures, called the indifference zone, which is the distance between the best and the rest in the design space, is nevertheless a cardinal concept. Furthermore, the probability of the observed best being the actual best is often too small to be useful, while the probability that one of the observed top 50 is actually among the true top 50 can be close to one for various problems, even with large estimation noise. As our results will show, it is the softening of goals that makes ordinal optimization ideas efficient in narrowing down choices. Therefore, our objective in this paper is to provide quantitative results for search reduction in ordinal optimization via alignment probability calculation.

This paper is organized as follows. Section 2 introduces formally the notions of good enough subset and selected subset, and discusses alignment probabilities with special attention to the case where we pick blindly from the search space. Section 3 examines the horse race selection and proposes a model to describe different shapes of performance profile. The results of subset selection based on Monte Carlo studies are summarized and shown in Section 4. The use of these results in search reduction is illustrated in Section 5. We conclude in Section 6.

2. Ordered Performance Curves and Alignment Probability

2.1. OPC, Good Enough Subset, and Representative Samples. Let us begin with a thought experiment. Suppose that we evaluate for all designs in Θ their performance values exactly, and that we plot these values from the smallest to the largest with the corresponding design labels (names) laid down and spaced equally on the abscissa axis. In such a way, we have created (by definition) a nondecreasing curve, which we term the ordered performance curve (OPC, Fig. 1). Notice that we have assumed the search

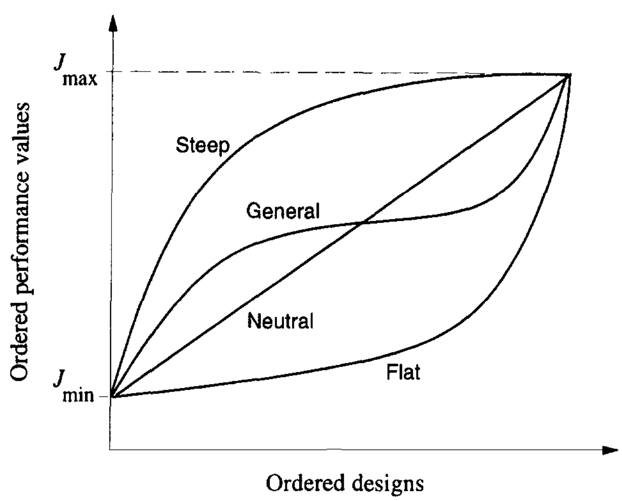

Fig. 1. Examples of ordered performance curves: flat, neutral, steep, and general.

space Θ to be finitely denumerable with bounded performance values.[9] Conceptually, it is possible to envision an ordered performance curve for every optimization problem, and if the OPC were available, then any optimization problem, whether to minimize or to maximize, could be solved directly by reading off from the OPC. Unfortunately, most problems in real life do not lend us such luxury, because exact performance evaluation can be very costly and time consuming due to the challenges mentioned in the previous section. Nevertheless, the concept of OPC does motivate our definition of good enough subset.

Definition 2.1. A good enough subset $G(\Theta)$ of a design space is the subset consisting of the top $n\%$ elements in the design space.

How we choose the value n depends on the softness of our goal in the optimization problem as well as certain background knowledge. We will discuss some related issues at the end of the paper. Using the above definition, the good enough subset $G(\Theta)$ is the set containing the smallest (i.e., top) $n\%$ elements in Θ. Graphically, the good enough subset is $n\%$ of the leftmost domain in the OPC abscissa axis. An important distinction here is that $G(\Theta)$ is defined as a portion of the search space, and it does not depend on how the performance values are distributed. In other words, construction of $G(\Theta)$ bears no dependence on the shape of OPC or the range of performance values, but only on the fact that designs are ranked according to the nondecreasing property of OPC. As a result, the good enough subset is a concept based on ordinal comparison, rather than cardinal differences. It also makes apparent the use of the term "ordinal optimization."

Next, we consider a set of representative samples drawn from the design space which forms the basis of our selection process. We rely on the following important assumption.

Assumption 2.1. Uniform Sampling. We can sample the design space Θ uniformly; i.e., each $\theta \in \Theta$ can be obtained with equal chance by some sampling scheme.

It is a statistical fact that, under independent uniform sampling, we can obtain a set of representative samples from the design space [David (Ref. 12)]. The validity of Assumption 2.1 could be problem dependent, however. At the end of this paper, we will discuss some issues pertinent to uniform sampling. Holding Assumption 2.1 valid, we then need to decide on the

[9]There is no restriction, however, to the size of the search space. In fact, we assume that $|\Theta| \approx \infty$ and $|\Theta| < \infty$. If the design space is a continuum of the Euclidean space, then proper discretization would be needed. We omit further details here.

number of representative samples, denoted by N, used in the selection process. This is our first occasion to apply the theme of goal softening in ordinal optimization. With respect to the good enough subset of the design space Θ, we are concerned about the event that at least one of the N samples falls into the top n-percentile of Θ. The probability of such event is given by

$$\begin{aligned} P &= 1 - \text{Prob}\{\text{all } N \text{ samples not in the top } n\text{-percentile of } \Theta\} \\ &= 1-(1-n\%)^N. \end{aligned} \tag{3}$$

For $N=1{,}000$ and $n=5$, we have $(1-5\%)^{1000} \approx 5.29\times 10^{-23}$, which makes $P\approx 1$. In other words, we have practically a very slim chance (of order 10^{-23}) that none of the 1,000 uniformly generated samples belongs to the top 5% of the design space. Thus, throughout this paper, we consider a representative set of 1,000 design samples. Note that $N=1{,}000$ and $n=5$ are by no means the only choices, nor the most general, but are used for illustrative purposes in this paper. Perhaps, the reader is still bothered by questions such as "Why not $N=2{,}000$ or $N=1\times 10^6$?" However, we contend, that while for fixed confidence (i.e., P) n is approximately inversely proportional to N,[10] increasing the number of samples unjustifiably in order to squeeze out a smaller n does not agree with the premise of goal softening in ordinal optimization, which aims at efficiently locating some good enough solutions with high confidence. Lastly, if the reader still wants to obtain alignment results for other values of N, some analytical results reported in Ho and Deng (Ref. 13) will be helpful. These results provide building blocks for calculating alignment probability (to be defined in the next subsection) when N is a multiple of 1,000.

2.2. Alignment Probability and Blind Picking Lower Bounds. Our subset selection process begins now with a representative set of 1,000 design samples, obtained under the assumption in the previous section. We shall denote this set by $\hat{\Theta}$ with the understanding that $|\hat{\Theta}|=N=1{,}000$. The designs are $\theta_i\in\hat{\Theta}$, $i=1,\ldots,N$, and their corresponding performance values are $J_i\equiv J(\theta_i)$. Notice that we have replaced the optimization problem (1) on Θ with a selection problem on $\hat{\Theta}$,

$$\min_{\theta\in\hat{\Theta}} J(\theta). \tag{4}$$

With respect to $\hat{\Theta}$, a good enough subset $G(\hat{\Theta})$ can be defined as previously. In particular, we are interested in the top g designs of $\hat{\Theta}$ where[11]

$$g=|G(\hat{\Theta})|=\lceil N\times n\%\rceil.$$

[10] A quick examination of (3) gives that $n\% = 1-e^{(1/N)\log(1-P)} \Rightarrow Nn\approx 100\log(1-P)$.

[11] Should there be a tie situation of more than g possible choices, we arbitrarily select those which tie with each other; and once selected, they will be fixed as members of $G(\hat{\Theta})$ from then on.

The ordered performances $J_{[i]}$, $i=1, \ldots, N$, are obtained by reordering the J_i such that

$$J_{[1]} \leq J_{[2]} \leq \cdots \leq J_{[N]},$$

and the corresponding ordered design samples are

$$\theta_{[1]}, \theta_{[2]}, \ldots, \theta_{[N]},$$

where[12]

$$J_{[i]} = J(\theta_{[i]}).$$

In this light, the OPC for $\hat{\Theta}$ is the graph of the ordered pairs $(\theta_{[i]}, J_{[i]})$, $i=1, \ldots, N$. Notice that the group $\{\theta_{[1]}, \theta_{[2]}, \ldots, \theta_{[N]}\}$ is a permutation of the designs $\theta_1, \theta_2, \ldots, \theta_N$. Consequently, the good enough subset can be directly written as

$$G(\hat{\Theta}) \equiv \{\theta_{[1]}, \theta_{[2]}, \ldots, \theta_{[g]}\}.$$

Notice that, with respect to $\hat{\Theta}$, we have summoned goal softening once again. Our objective now is to pick out a selected subset $S(\hat{\Theta})$ from $\hat{\Theta}$, based on certain selection rule, so as to achieve a reasonable degree of matching with $G(\hat{\Theta})$. Since our focus is now switched to $\hat{\Theta}$, then for typographical simplicity, we shall use G and S in lieu of $G(\hat{\Theta})$ and $S(\hat{\Theta})$, should there be no confusion.

We shall make precise the meaning of "reasonable degree of matching" as follows. By matching or alignment, we mean the intersection of the good enough subset and the selected subset. We are then interested in the alignment probability of $\{G \cap S\}$, defined by

$$\mathscr{P}_{\mathscr{A}} = \text{Prob}\{|G \cap S| \geq k\}, \tag{5}$$

where k is called the alignment level. Notice that

$$1 \leq k \leq \min(|G|, |S|),$$

and we let

$$\mathscr{P}_{\mathscr{A}} = 1, \qquad \text{when } k=0.$$

Alignment probability depends on the alignment level k, as well as:

(i) size of the good enough subset (i.e., g or $n\%$);
(ii) size of the selected subset size ($|S| \equiv s$);
(iii) subset selection rule;
(iv) noise characteristics during performance evaluations;
(v) OPC of $\hat{\Theta}$.

[12] In fact, $J_{[1]} \leq J_{[2]} \leq \cdots \leq J_{[N]}$ form the order statistics from the performance density of the design space. See the Appendix for detailed explanation.

The subset sizes [items (i) and (ii)] are parameters chosen by the designer according to certain softened criteria. These subset sizes are suggestions that the authors aim at providing in this paper. Item (iii), i.e., the selection rule, determines how S is picked, and we will examine two selection rules in this paper: blind picking (BP) and horse race rule (HR). Other possible selection rules such as round-robin tournament, pairwise elimination, and seeded competition are commonly found in sport games. Determining the appropriate choice of selection rule could sometimes be assisted by certain specific knowledge of the problem such as the form of solution, structure of search space (if applicable), some asymptotic or limiting behavior, and above all common sense.

Meanwhile, noise characteristics and the underlying OPC [items (iv) and (v)] are not choices for the designer, but rather relate to general knowledge about the selection problem, such as whether there would be a large proportion of good designs, or bad designs, or intermediate designs, etc. This general knowledge could be guessed by the designer based on very crude performance evaluations, or even a priori, and this can often be easily done. More about this will be examined in the next section. As explained at the beginning of this section, the actual OPC is hard to obtain due to noisy performance evaluation. The determination of the OPC and noise characteristics is a separate estimation problem, and this will be investigated in a future paper; see the Appendix for related ideas.[13] On the other hand, should there be absolutely no knowledge regarding how to select from the search space, then the designer is in no better position than to blindly pick out a selected subset. We refer to blind picking as the procedure which involves selecting a subset S from the representative set $\hat{\Theta}$: (a) randomly, (b) without replacement, and (c) without comparison. This is an interesting case because:

(A) it is by itself a selection rule [item (iii)];
(B) it is equivalent to the case where OPC is absolutely flat [item (v)];
(C) it is equivalent to the case where noise is infinite [item (iv)].

Either (B) or (C), or both, would warrant the blind picking situation, because every design has the same tendency to be evaluated to any rank[14] in $\hat{\Theta}$. In addition, the alignment probability for this special case admits a

[13]Nevertheless, to pinpoint the exact noise characteristics is often of secondary concern so long as we can be conservative with the alignment results.

[14]The reason why BP is equivalent to the case of an absolutely flat OPC is that the designer is blindfolded and therefore is not able to perform any comparisons of the picked designs during the picking process. It is as if the OPC is absolutely flat, exhibiting no detectable difference among all designs.

closed form expression, i.e.,

$$\mathscr{P}_{\mathscr{A}}(k, s, g \mid \mathrm{BP}) = \mathrm{Prob}\{|G \cap S| \geq k\}$$
$$= \sum_{i=k}^{\min(g,s)} \frac{\begin{bmatrix} g \\ i \end{bmatrix}\begin{bmatrix} N-g \\ s-i \end{bmatrix}}{\begin{bmatrix} N \\ s \end{bmatrix}}, \tag{6}$$

which is the hypergeometric distribution (the reader is reminded that $N = 1{,}000$). Equation (6) was reported in Ho et al. (Ref. 1) and Ho and Deng (Ref. 13), and we shall call it here the blind picking lower bound (BPLB) alignment probability. It is a lower bound because, when the OPC is not absolutely flat and noise is not infinite (i.e., when there is relevant knowledge, however approximate), the alignment situation will certainly be improved.[15]

Equation (6) can also give us suggestions on the subset size s when the desired confidence $\mathscr{P}_{\mathscr{A}}$ and alignment level k are given. Notice that no performance evaluation is required when we select by using the BPLB probabilities, except for spinning a lottery device to get s designs from $\hat{\Theta}$. Figure 2 shows the required subset sizes versus the desired alignment levels for the special case when $g = s$ and three BPLB probability values: 0.99, 0.95, 0.90. To understand these curves, consider alignment level $k = 1$. According to the curves, if we blindly pick out 67 designs from $\hat{\Theta}$ (again, with no performance evaluation), then we are guaranteed with probability 0.99 that at least one of the 67 designs is indeed among the top 67 designs out of $N = 1{,}000$ (a 15-fold reduction from $\hat{\Theta}$). Similarly, if we spin out 48 designs from the lottery device, then with 90% confidence we can be sure that at least one of the 48 that we have picked belongs actually to the true top 48 designs (a 20-fold reduction). The results are surprising, because we have absolutely no knowledge about $\hat{\Theta}$ and we have done no performance evaluation. These results are analogous to the famous example of the birthday paradox.[16]

3. Observed Performance under Horse Race Rule

3.1. Horse Race Rule and Ordered Noisy Performance. Departing from the blind picking situation, our subset selection will need to be based

[15] We assume here that the inherent randomness of all designs is independent and identically distributed; see Assumption 3.1 in the next section.

[16] The birthday paradox states that, with more than 50% chance, there would be a birthday matched among a handful of 23 people, and for merely 50 people, the chance of matching is increased to 97%.

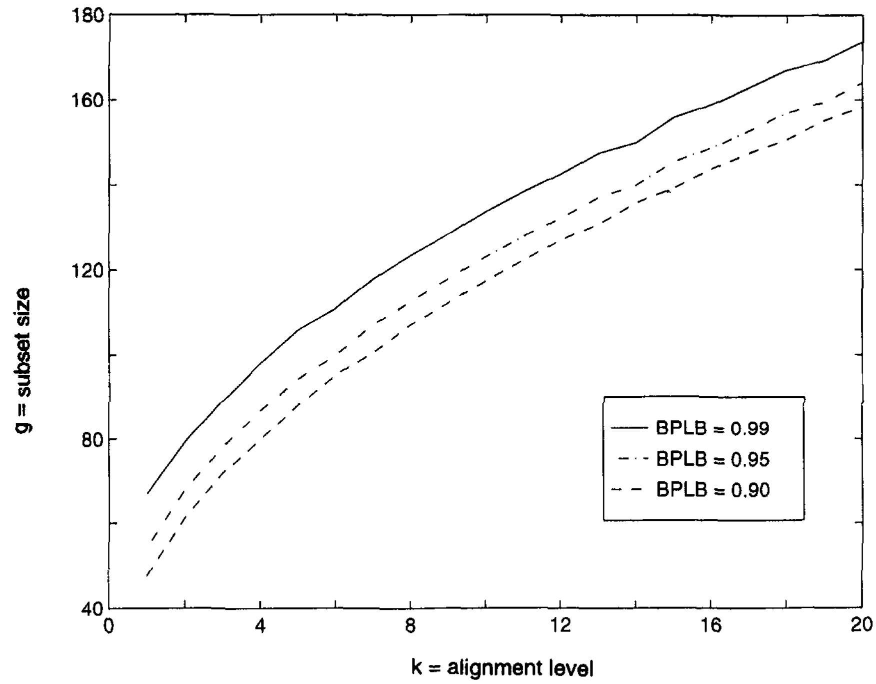

Fig. 2. Subset size versus alignment level for BPLB = 0.99, 0.95, 0.90.

on some estimates of design performance, however crude or inaccurate. In particular, we consider the horse race rule (HR) in comparing the performance estimates. The HR rule can be pictured as having all designs in $\hat{\Theta}$ competing at the same time, very much similar to N horses running a race in which some designs could be leading at a certain moment in the evaluation process, but could also be falling behind at another instant. The positions of the designs are determined by their estimated performance values. The running is stopped simultaneously, and the performance estimates at the stopping time determine a rank for each design, on which our subset selection is based. Such picture of simultaneous racing can be naturally generated when the evaluation process is carried out in the modern massively parallel computing environment, although any sequential method of performance evaluation can also give us the necessary snapshots.

The performance estimate $\tilde{J}(\theta_i)$ of each design $\theta_i \in \hat{\Theta}$ is the true performance $J(\theta_i)$ corrupted with additive noise. Mathematically, we have, for $i = 1, \ldots, N$,

$$\tilde{J}_i \equiv \tilde{J}(\theta_i) = J_i + \omega_i, \tag{7}$$

where $J_i = J(\theta_i)$ and ω_i summarizes all randomness involved in evaluating design θ_i. The following assumption will be imposed:

Assumption 3.1. Noise Independence. The ω_i are independent and identically distributed with probability density $\xi(\cdot)$.

Despite the required Assumption 3.1 for our results, Deng, Ho, and Hu (Ref. 14) showed that, under reasonable assumption on noise,[17] correlations can in fact enhance the identification of the good enough designs in the simulation process. Furthermore, notice that any ordinal comparison of the θ_i based on the $\tilde{J}_i$ will be unaffected even if $\xi(\cdot)$ has nonzero mean, because all performance estimates are equally shifted. Therefore, without loss of generality, we further impose, for all $i = 1, \ldots, N$,

$$E(\omega_i) = 0, \qquad \mathrm{Var}(\omega_i) = \sigma_\xi^2, \tag{8}$$

where $\sigma_\xi^2 < \infty$. However, the reader may notice that, because of the noise effect, it becomes impossible for us to identify the design label θ_i with the true performance value J_i or the observed performance value $\tilde{J}_i$. What we actually collect at the end of the horse race are the reordered version of the observed performances, or the ordered noisy performances, given by

$$\tilde{J}_{[1]} \le \tilde{J}_{[2]} \le \cdots \le \tilde{J}_{[N]}.$$

Here, $\tilde{J}_{[i]}$ stands for the ith smallest observed performance value. It should be emphasized that the bracketed subscript $[i]$ bears no direct relationship with the design label θ_i or $\tilde{J}_i$. This is because, for each design θ_j, the additive noise can displace $J(\theta_j)$ to a different position, say i, in the list of ordered noisy performances. Let the random variable for the ith observed design be $\tilde{\theta}_{[i]}$, $i = 1, \ldots, N$. Then, we have

$$\tilde{J}_{[i]} = \tilde{J}(\tilde{\theta}_{[i]}) = \tilde{J}(\theta_j),$$

for some j. In the worst case, when observed performances are very perverse, the θ_j corresponding to the smallest J_j can be displaced to the last positions, i.e., reordered as realization of $\tilde{\theta}_{[N]}$ or $\tilde{\theta}_{[N-1]}$, etc.

Subset selection for the HR rule is based on the ordered noisy performances, and the selected subset is defined as

$$\begin{aligned} S &\equiv \{\theta_j \mid i \le s, \tilde{J}_{[i]} = J_j + \omega_j, \text{ for some } j\} \\ &= \{\tilde{\theta}_{[1]}, \tilde{\theta}_{[2]}, \ldots, \tilde{\theta}_{[s]}\}, \end{aligned}$$

[17] Essentially, noise is normally distributed.

where $s = |S|$ is the size of the selected subset. The following summarizes the selection process using the HR selection rule:

Design samples of $\hat{\Theta}$	$\theta_1, \ldots, \theta_N$		$\omega_1, \ldots, \omega_N$	Noise samples $\sim \xi(\cdot)$
	$\Downarrow$		$\Downarrow$	
Performances $J(\theta_i)$	$J_1, \ldots, J_N$	$\overset{+}{\Rightarrow}$	$\tilde{J}_1, \ldots, \tilde{J}_N$	Noisy performances
	$\Downarrow$		$\Downarrow$	
	Reordering		Reordering	
	$\Downarrow$		$\Downarrow$	
Ordered performances	$J_{[1]}, \ldots, J_{[N]}$		$\tilde{J}_{[1]}, \ldots, \tilde{J}_{[N]}$	Ordered noisy performances
	$\Downarrow$		$\Downarrow$	
Ordered design samples	$\theta_{[1]}, \ldots, \theta_{[N]}$		$\tilde{\theta}_{[1]}, \ldots, \tilde{\theta}_{[N]}$	Observed design labels
	$\Downarrow$		$\Downarrow$	
Good enough designs G	$\{\theta_{[1]}, \ldots, \theta_{[g]}\}$		$\{\tilde{\theta}_{[1]}, \ldots, \tilde{\theta}_{[s]}\}$	Selected designs S

The alignment probability for the HR rule is defined in the same way as in (5), i.e.,

$$\mathscr{P}_{\mathscr{A}} \equiv \text{Prob}\{|G \cap S| \geq k\} = \mathscr{P}_{\mathscr{A}}(k, s, g|\text{HR}).$$

However, unlike the blind pick selection rule, it does not admit a closed-form solution but can be calculated via simple Monte Carlo methods. We have carried out a number of Monte Carlo studies, and the results are shown in the next section. Meanwhile, notice from the above diagram that the HR alignment probabilities depend upon the underlying OPC of $\hat{\Theta}$ and the noise density $\xi(\cdot)$. In view of the conditions imposed in (8), we can consider parametrizing $\xi(\cdot)$ by σ_ξ^2 under some functional form. Thus in characterizing alignment probabilities for the HR rule, what remains is to devise appropriate models for various OPCs.

3.2. Shapes of OPCs. The OPC of $\hat{\Theta}$ is determined by the spread of the ordered performances $J_{[1]}, J_{[2]}, \ldots, J_{[N]}$. Without loss of generality, let us normalize the $J_{[i]}$ into the range $[0, 1]$; i.e., for $i = 1, \ldots, N$,

$$y_i = (J_{[i]} - J_{[1]})/(J_{[N]} - J_{[1]}). \tag{9}$$

Meanwhile, we also consider mapping the ordered design samples, spaced equally, into the range $[0, 1]$; i.e., we have a mapping $x: \hat{\Theta} \to [0, 1]$ such that, for all $i = 1, \ldots, N$,

$$x(\theta_{[i]}) \equiv x_i = (i-1)/(N-1). \tag{10}$$

Such normalization facilitates comparisons between different types of OPCs and between different noise characteristics. For the former, we consider five general categories of OPC models:

(i) lots of good designs;
(ii) lots of good and lots of bad designs, but few intermediate ones;
(iii) equally distributed good, bad, and intermediate designs;
(iv) lots of intermediate designs, but few good and few bad designs;
(v) lots of bad designs.

The readers may recall that the exact shape of an OPC is usually unavailable because of noisy performance evaluation. However, it is reasonable to suppose that, if a designer is able to guess from one of the above five categories what the search space is likely to be, then he/she would be able to further improve alignment results in the subset selection process. This guessing could be achieved by comparing very crude performance estimates, or by some general knowledge that is related to the designer expertise in solving the problem. Here, man-machine interaction could play a role in some artificial intelligence based search schemes. Meanwhile, a natural question is: Would the five categories be sufficient in describing most scenarios in optimization? To this end, it is often reasonable to ask the designer to classify problems into five types; then, we need to show that we can use one typical shape in these classes to compute the alignment probability without causing too much error. We shall demonstrate this via numerical results in the next section that these five categories are indeed sufficient.

Notice that the graph of (x_i, y_i) given by (9) and (10) is nondecreasingly piecewise linear. In order to accommodate the five OPC types using the smallest number of parameters, we consider modeling the OPC by a smooth curve $y = \Lambda(x)$ for $x \in [0, 1]$, with the properties that $\Lambda(0) = 0$, $\Lambda(1) = 1$, and $\Lambda(\cdot)$ is nondecreasing. We shall call $\Lambda(\cdot)$ the standardized OPC. The function that we are going to employ for $\Lambda(\cdot)$ is the inverse mapping of the incomplete beta function, parametrized by a pair of numbers α and β. More precisely, for beta density $f(y \mid \alpha, \beta)$, $\alpha > 0$, and $\beta > 0$,

$$f(y \mid \alpha, \beta) = Cy^{\alpha-1}(1-y)^{\beta-1}, \tag{11}$$

where

$$C = \Gamma(\alpha+\beta)/\Gamma(\alpha)\Gamma(\beta)$$

and $\Gamma(\cdot)$ is the gamma function, we have the cumulative distribution function, or the incomplete beta function, given by

$$F(y \mid \alpha, \beta) = \int_0^y f(z \mid \alpha, \beta)\, dz.$$

Then, the standardized OPC is determined by

$$\Lambda(x \mid \alpha, \beta) = F^{-1}(x \mid \alpha, \beta) = F(x \mid 1/\alpha, 1/\beta), \tag{12}$$

where $F(\cdot \mid \cdot, \cdot)$ can be evaluated[18] via numerical approximation formulas (Ref. 15). The density function $f(y \mid \cdot, \cdot)$ plays the role of the so-called standardized performance density of $\hat{\Theta}$. In the Appendix, we will show that performance density and OPC are mutually equivalent concepts. Meanwhile, this two-parameter model does provide us the flexibility in describing the five OPC categories by varying α and β. In particular, α and β correspond to the flatness of $\Lambda(\cdot)$ near the regions of low and high performance values. For example, when $\alpha \ll 1$, the standardized OPC near the origin is very flat, which implies that there is a larger proportion of small-valued designs, whereas for $\beta \gg 1$ the standardized OPC near [1, 1] is very steep, and consequently there are only few large-valued designs. When $\alpha \approx 1$ and $\beta \approx 1$, the standardized OPC is close to the 45° straight line, which means that the performance values for all designs in $\hat{\Theta}$ are relatively equally separated. In the limiting case where either $\alpha = 0$ or $\beta = 0$, we have an absolutely flat OPC, which corresponds to the blind picking phenomenon. The relationship between α, β, and $\Lambda(\cdot)$ is summarized in Fig. 3.

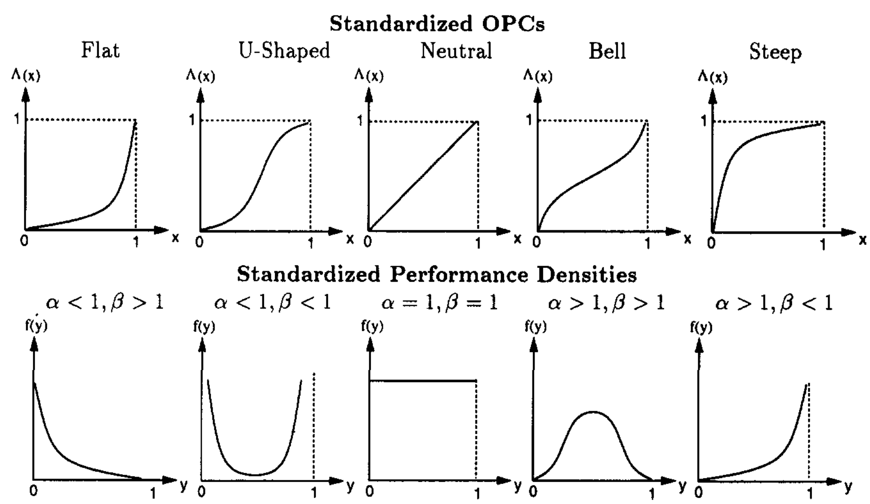

Fig. 3. Examples of beta densities and corresponding standardized OPCs.

[18] The last equality in (12) can be easily seen when either $\alpha = 1$ or $\beta = 1$; otherwise, the equality is observed graphically.

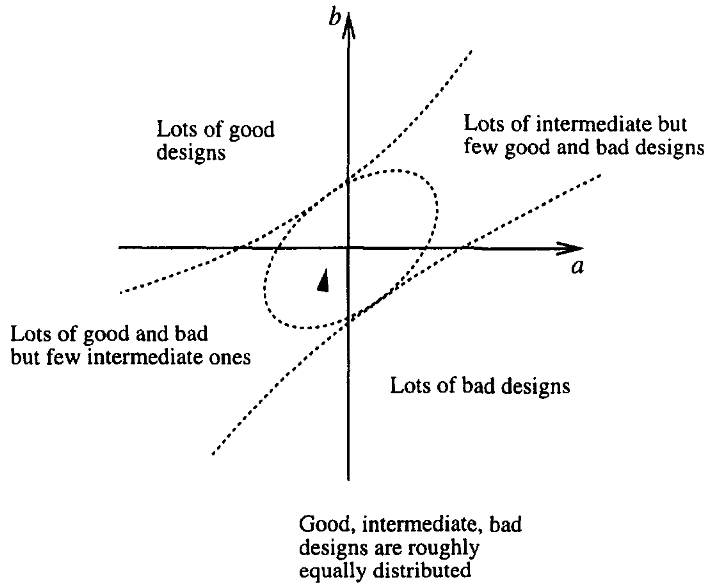

Fig. 4. Partitions of the ab-plane for five OPC categories.

Let us now relate the parameters α and β with the five categories of OPC models. It is actually easier to see the relationship between the $\alpha\beta$-pair and the five OPC categories in the ab-plane, where $a = \log \alpha$ and $b = \log \beta$. In particular, the origin of the ab-plane corresponds to the linear OPC (i.e., when α and β both equal to 1). The limiting case of absolutely flat OPC is represented by either a or b approaching ∞ or $-\infty$. The relative locations of the five OPC categories in the ab-plane are displayed in Fig. 4. The partitioning boundaries in the ab-plane are nevertheless for approximation only, since the distinction of lots of good designs, etc., could be subjectively varied. A fuzzifying approach may also be applied, but will not be considered in this paper.

4. Numerical Results for Selected Subsets

A natural question from the user point of view is: If the user would give an estimate of the underlying OPC class $\mathscr{C}$, as well as the noise situation $\xi(\cdot)$, then how many from the N designs should the user pick so that, with probability $\mathscr{P}_{\mathscr{A}}$, at least k designs of the true top g ones are included in the selected subset? Our objective in this section is to provide an answer to the above question by giving suggestions on the selected subset size. Formally, the required subset sizes are given by the function

$$Z = Z(g, k; N, \mathscr{C}, \xi(\cdot), \mathscr{P}_{\mathscr{A}}), \tag{13}$$

where $\mathscr{C}$ is one of the five generic OPC classes and $\xi(\cdot)$ is the noise characteristics, both supplied by the user. In this paper, we consider only $N = 1{,}000$ and $\mathscr{P}_{\mathscr{A}} = 0{\cdot}95$; therefore, the dependence on N and $\mathscr{P}_{\mathscr{A}}$ will be dropped for notational convenience. Notice that $Z(g, k)$ is decreasing in g and increasing in k. Closed-form expressions for (13) cannot be easily obtained, except for the blind pick selection rule mentioned earlier and a few other special cases.[19] However, we can estimate $Z(\cdot, \cdot)$ by running Monte Carlo simulations using a certain choice of $\xi(\cdot)$. Let us discuss our ideas as follows.

Using the two-parameter OPC model introduced in the previous section, we studied a total of 88 OPCs indexed by α and β as follows:

$$\alpha = \{0.15, 0.25, 0.40, 0.65, 1.00, 1.50, 2.00, 3.00, 4.50, 8.00\},$$

$$\beta = \{0.15, 0.25, 0.40, 0.65, 1.00, 1.50, 2.00, 3.00, 4.50, 8.00\}.$$

These parameters are marked on the *ab*-plane as shown in Fig. 5. The actual OPCs are grouped into each class, and they sufficiently represent the five types of OPC considered, which can be confirmed by the grid points in Fig. 5. Comparing Fig. 5 with Fig. 4, we ascribe the 10 OPCs in the third quadrant of the *ab*-plane as the U-shaped class, the 19 OPCs near the origin as the neutral class, and the 15 OPCs in the first quadrant as the bell-shaped class. The rest of the 88 cases in the second and fourth quadrants of the *ab*-plane belong respectively to the flat and steep classes.

As for noise characteristics, we employed in this study the uniform noise density $\xi(\cdot) \sim U[-W, W]$, which has a variance $\sigma_\xi^2 = W^2/3$. In particular, we considered three values: $W = 0.5$, 1.0, 2.5. Notice that, for $W = 0.5$, the range of noise is in fact equal to the range of the OPC, which with nonzero probability can result in swapping the ranks of some good enough designs with the worst designs. The noise range for $W = 1.0$ is twice as large as the range of OPC values. We therefore ascribe these respectively as medium and large noise ranges. The case $W = 2.5$ produces intermediate alignment results between large noise and infinite noise, and we call this the very large noise range.

[19]Those special cases are referred to as the least favorable configurations in the ranking and selection literatures, and they are similar to the blind pick scenario, except that the good enough subset is separated from the rest of the designs by a distance called the indifference zone. See Gupta et al. (Ref. 9) and Santner et al. (Ref. 10) for further details. Nonetheless, their treatments are not befitting of the premises of ordinal optimization, as explained in Section 1.

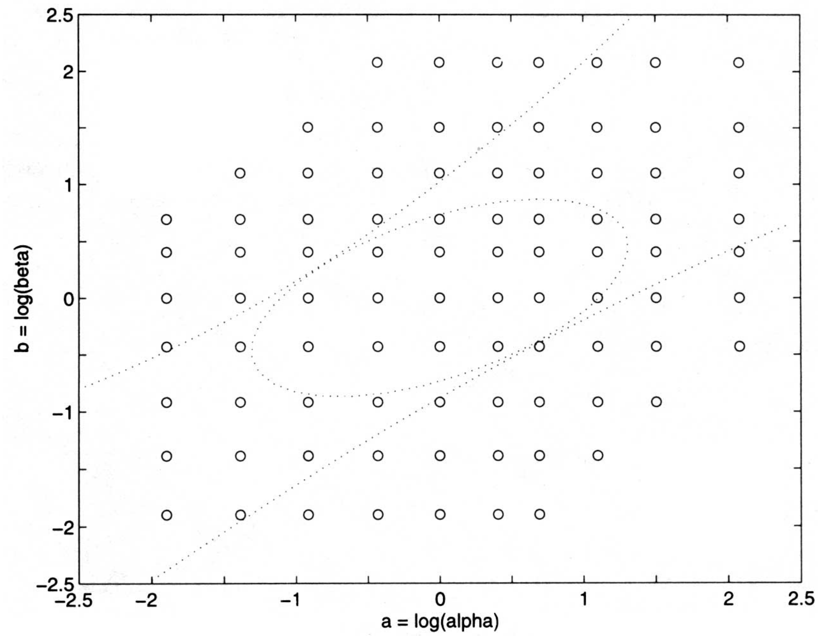

Fig. 5. Positions of the studied OPCs in the *ab*-plane.

In all of our Monte Carlo calculations, we simulated 10,000 realizations of noisy OPC's, and the alignment probabilities for $g = |G|$ and $s = |S|$ ranging from 20 to 200 are recorded. Then, we interpolated the selected subset sizes at alignment probability $\mathscr{P}_{\mathscr{A}} = 0.95$ for alignment levels k from 1 to 10. Typical results for the case $(\alpha, \beta) = (1.00, 1.00)$ are shown in Fig. 6. These figures suggest that a low-order polynomial in k and g may be used to approximate the subset sizes sufficiently well. Notice that the subset sizes decrease in g and increase in k (with $s = 0$ when $k = 0$). We tried fitting polynomials of various orders and found that the following functional form suits well in all cases:

$$Z(k, g) = e^{Z_o} k^{\rho} g^{\gamma} + \eta, \tag{14}$$

where Z_o, ρ, γ, η are constants depending on the OPC types and noise characteristics. For each of the five OPC classes, we take the maximum of the required subset sizes based on the simulated results from all OPCs belonging to the same class. Then, we perform a regression on the MAXed data, which in turn produces the coefficients appearing in (14). Together with the blind pick selection rule [i.e., the alignment probabilities given by (6)], we tabulate all coefficients in Table 1. It should be noted that the

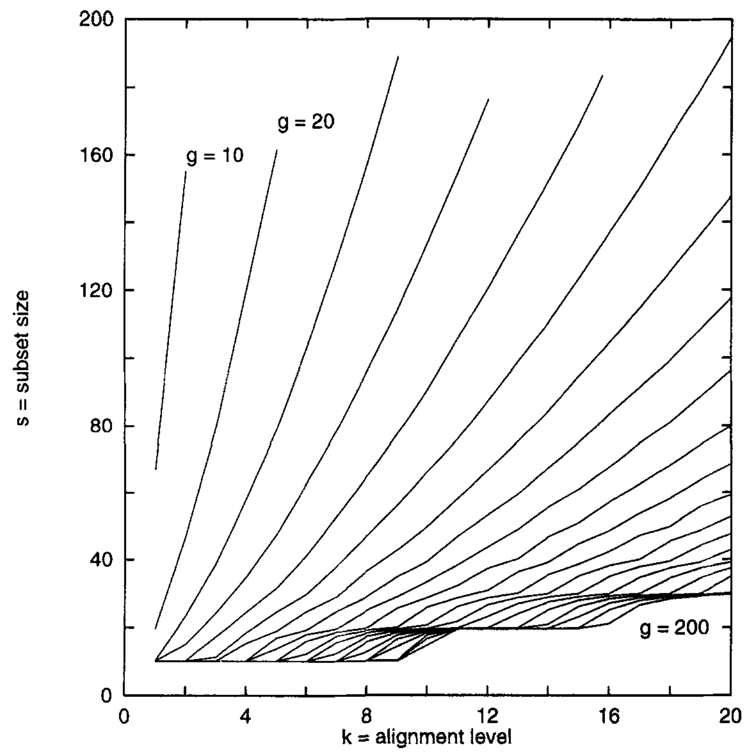

Fig. 6. Subset sizes interpolated from simulated data for OPC case $\alpha = \beta = 1.00$ and $W = 1.0$.

subset sizes calculated by the coefficients above have a working range of $20 \leq g \leq 200$, $Z(\cdot, \cdot) < 180$, and when the fraction k/g is small. We have calculated the subset sizes for the five OPC classes using all noise cases, as well as those of BP selection (i.e., absolutely flat OPC or infinite noise). Figure 7 compares the subset selection sizes at different noise levels when $g = 50$. In the occasions when the noise factor is characterized to be within these predetermined levels, proper interpolation of the subset sizes will suffice.

5. Examples of Search Reduction

We demonstrate in this section the utility of the subset selection results obtained previously. Let us also emphasize that the results are universally applicable to problems of a wide range, such as in modeling and simulation,

Table 1. Regressed values of Z_o, ρ, γ, η in $Z(k, g)$.

Noise	∞	$U[-0.5, 0.5]$				
OPC class	B-Pick	Flat	U-Shape	Neutral	Bell	Steep
Z_o	7.8189	8.1378	8.1200	7.9000	8.1998	7.7998
ρ	0.6877	0.8974	1.0044	1.0144	1.9164	1.5099
γ	−0.9550	−1.2058	−1.3695	−1.3995	−2.0250	−2.0719
η	0.00	6.00	9.00	7.00	10.00	10.00
Noise	∞	$U[-1.0, 1.0]$				
OPC class	B-Pick	Flat	U-Shape	Neutral	Bell	Steep
Z_o	7.8189	8.4299	7.9399	8.0200	8.5988	7.5966
ρ	0.6877	0.7844	0.8989	0.9554	1.4089	1.9801
γ	−0.9550	−1.1795	−1.2358	−1.3167	−1.6789	−1.8884
η	0.00	2.00	7.00	10.00	9.00	10.00
Noise	∞	$U[-2.5, 2.5]$				
OPC class	B-Pick	Flat	U-Shape	Neutral	Bell	Steep
Z_o	7.8189	8.5200	8.2232	8.4832	8.8697	8.2995
ρ	0.6877	0.8944	0.9426	1.0207	1.1489	1.3777
γ	−0.9550	−1.2286	−1.2677	−1.3761	−1.4734	−1.4986
η	0.00	5.00	6.00	6.00	7.00	8.00

Numeric ranges: $g \in [20, 200]$, $k \in [1, 10]$, $Z(\cdot, \cdot) < 180$.

and the designer is required to supply only a minimal amount of information. In particular, if the designer is unable to decide which particular OPC class is to be considered, a fair guess using the neutral class would be sufficient to begin the search. Moreover, there is always the blind pick selection size to serve as fallback solution for the designer (upper bound of a selected subset).

Example 5.1. Picking with an Approximate Model. The first example concerns using a simple model to approximate a complex model. Consider the following function defined on the range $\Theta = [0, 1]$:

$$J(\theta) = a_1 \sin(2\pi\rho\theta) + a_2\theta + a_3, \tag{15}$$

where

$$a_1 = 3, \qquad a_2 = 5, \qquad a_3 = 2.$$

For $\rho = 500$, there are five hundred cycles in the range [0, 1]. To estimate the exact functional form of (15) may require extensive evaluation of the entire domain [0, 1] and the proper choice of a fitting function. Similar identification problems appear in many application areas. However, here we

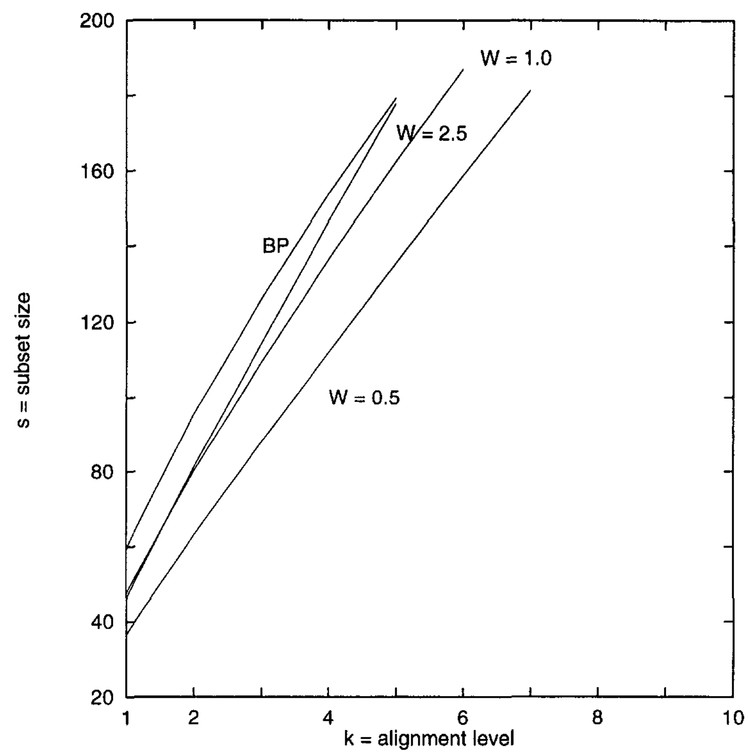

Fig. 7a. Subset selection sizes for the flat OPC class at different noise levels with $g = 50$.

consider using a crude model to approximate (15). In particular, based on the observation that there is a general rising trend in [0, 1], we use a linear function,

$$\tilde{J}(\theta) = 5\theta. \tag{16}$$

Notice that only the linear part of (15) is contained in the crude model (16), which can be considered as a noisy version of the true model. In other words,

$$\tilde{J}(\theta) = J(\theta) + \text{error}.$$

By generating 1,000 uniform samples from [0, 1], we have

$$\hat{\Theta} = \{\theta_1, \ldots, \theta_{1{,}000}\};$$

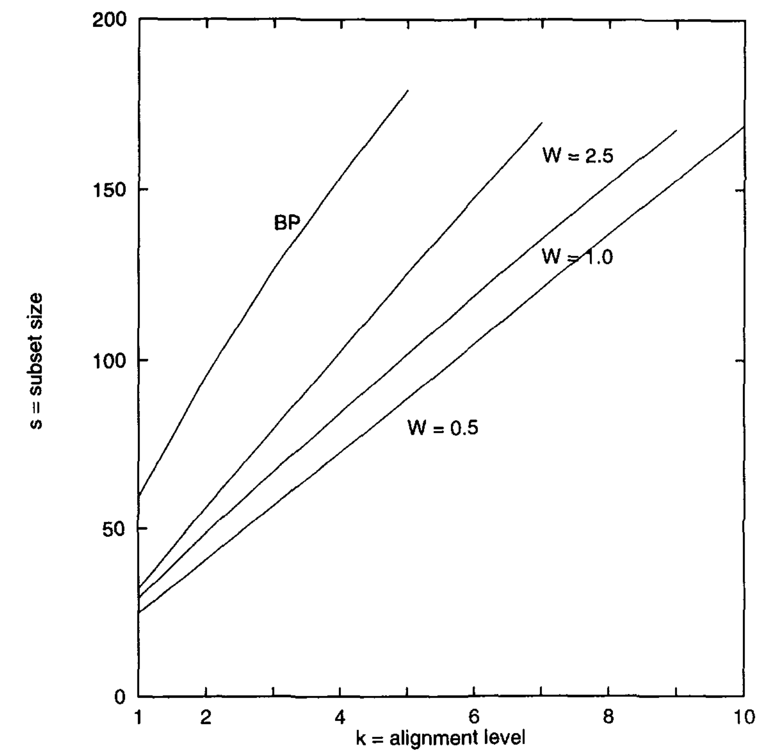

Fig. 7b. Subset selection sizes for the U-shaped OPC class at different noise levels with $g=50$.

the noise range can be estimated by

$$W=\max_{\theta_i\in\hat{\Theta}}(|J(\theta_i)-\tilde{J}(\theta_i)|) \tag{17}$$

after adjusting for the mean values.

We selected the neutral OPC class for this example. Once the good enough criterion g and the alignment level k are specified, the required selected subset size s from the crude model (16) is given by $Z(g, k \mid \text{neutral}, W)$. Notice that these selected elements correspond to the first s members of $\hat{\Theta}$, because of the monotone property of the crude model. Then, we compare the selected subset with the true model to determine the number of elements which indeed match with the good enough designs.

Accordingly, we have carried out 1,000 experiments, each with a different $\hat{\Theta}$ generated, so as to validate the alignment probability against $\mathscr{P}_{\mathscr{A}}=$

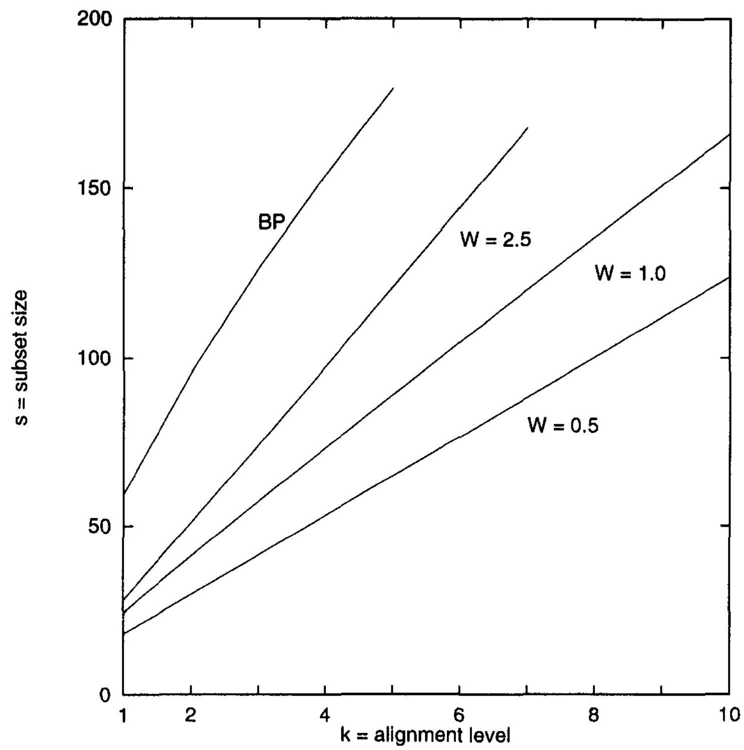

Fig. 7c. Subset selection sizes for the neutral OPC class at different noise levels with $g=50$.

0.95. We determined the alignments of each subset of size $Z(g, k \mid \text{neutral}, W)$, where $g=20, \ldots, 200$ and $1 \leq k \leq 10$. Some of the alignment probabilities are plotted in Fig. 8. Each line in Fig. 8 represents the fraction of the 1,000 experiments in which there are at least k of the g good enough designs matched in the selected subset. Note that:

(i) the alignment probabilities are in general greater than 0.95, and this can be attributed to the conservative estimates of $Z(\cdot, \cdot)$;
(ii) some fluctuations of the alignment probabilities are observed, and this is due to the residues of the regression functions of $Z(\cdot, \cdot)$.

As seen from this example, by adopting a softened criterion, one can indeed achieve good alignment results by employing a very crude model in lieu of a complex model. This shows the importance of capturing the trend or general behavior of a system prior to the study of essential details. Perhaps, this also explains why a designer's intuition is often more valuable in

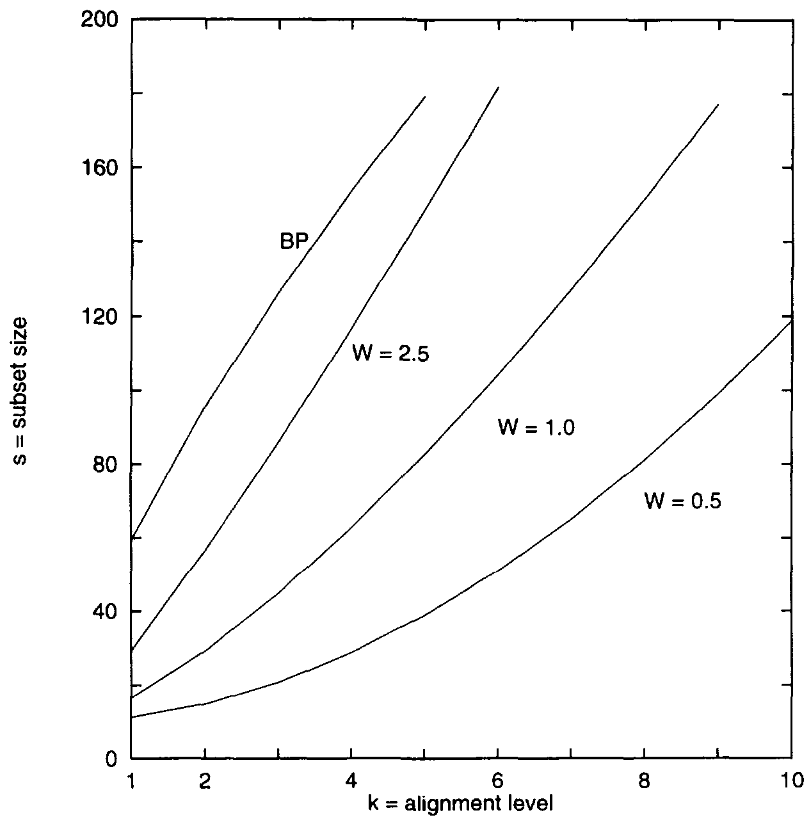

Fig. 7d. Subset selection sizes for the bell-shaped OPC class at different noise levels with $g=50$.

the initial phase of a design process. Once a number of good enough designs are singled out, detailed studies of these designs can be done in the subsequent stages of the design process.

Example 5.2. Picking in Running Short Simulations. In our second example, we consider the cyclic server problem discussed in Ho et al. (Ref. 1). The system has 10 buffers (of unlimited capacity) for 10 arrival streams modeled by Poisson processes with rates $\lambda_1, \ldots, \lambda_{10}$ respectively. There is a single cyclic server serving the 10 buffers in a round-robin fashion: at buffer i, m_i jobs are served until the buffer becomes empty, whichever comes first; then, the server moves from buffer i to buffer $i+1$ with a changeover time of length δ_i (Fig. 9). A holding cost of C_i units at buffer i is incurred.

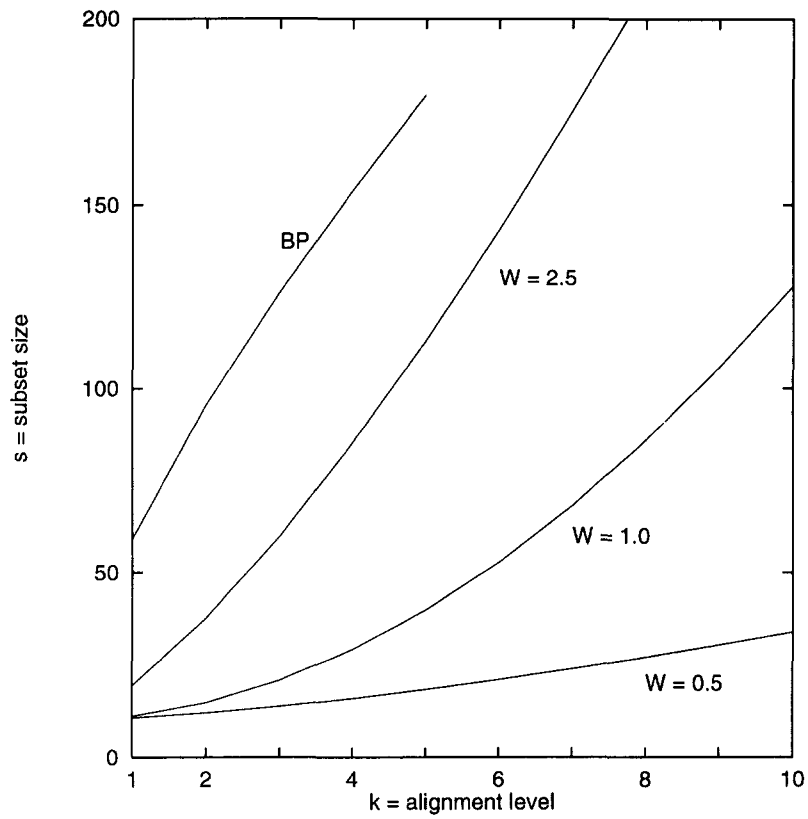

Fig. 7e. Subset selection sizes for the steep OPC class at different noise levels with $g = 50$.

The objective is to find a service policy $(m_1, m_2, \ldots, m_{10})$ such that it minimizes the average holding cost per job per unit time in the system. We assume that $0 \leq m_i \leq 10$ for all i; in other words, no more than 10 jobs may be served at each buffer for any policy. The design space Θ is therefore the lattice

$$\Theta = \{m = (m_1, m_2, \ldots, m_{10}) | 0 \leq m_i \leq 10, \forall i\}.$$

The cost coefficients and arrival rates are respectively

$$(C_1, \ldots, C_{10}) = (1, 1, 1, 10, 1, 50, 1, 1, 1, 1),$$

$$(\lambda_1, \ldots, \lambda_{10}) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1),$$

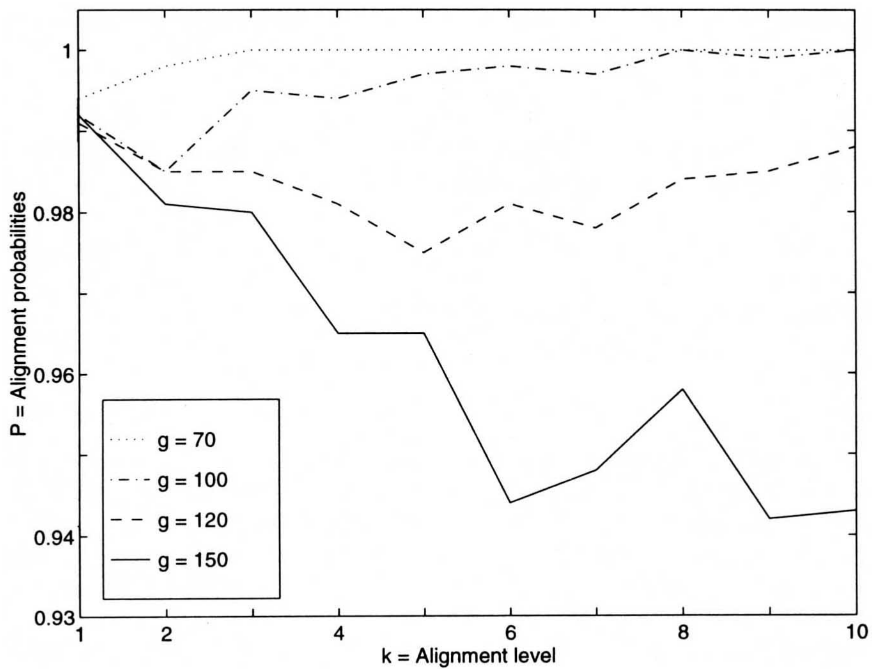

Fig. 8. Alignment probability validation of the first example.

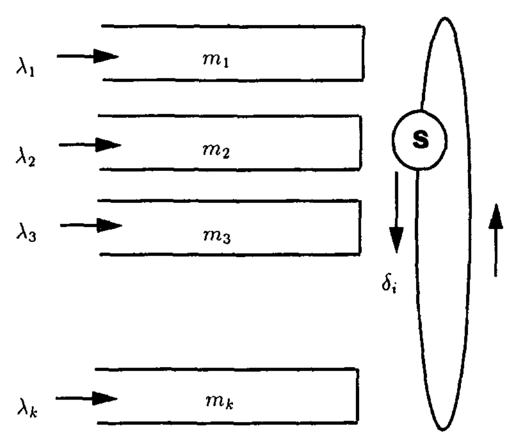

Fig. 9. Cyclic server serving K stream of arrivals.

with a service rate of the server $\mu = 20$, and the mean changeover time of δ_i is

$$E(\delta_i) = 1/30, \qquad \text{for all } i.$$

All random quantities are exponentially distributed. Notice that buffer 4 and buffer 6 have much higher cost coefficients.

We have generated 1,000 policies (designs) from Θ and run long simulations for each policy to obtain their true ordering.[20] After 16,753 jobs have arrived the system, the best 20 ordered designs are

$$\{\theta_{[1]}, \theta_{[2]}, \ldots, \theta_{[20]}\} = \{761, 166, 843, 785, 417, 456, 205, 925, 234, 70, 586, 91, 93, 493, 818, 565, 928, 250, 716, 840\},$$

which will be taken as the true ordering of the top 20 designs. Assume that we are interested in obtaining any of these top 20 designs; i.e., they form the good enough subset from the 1,000 design samples; then, we could have stopped the simulation at much earlier instants. Suppose that we had terminated the simulation at the times when 161 and 330 jobs had arrived in the system.[21] Let us call these two time instants T_1 and T_2, respectively, and we have taken the corresponding noise levels to be large and medium. Without any prior knowledge, we conjectured a neutral OPC for the 1,000 designs. Then, the required subset selection sizes at these two instants are given as

$$s_{T_1} = Z(20, 1 \mid \text{neutral, large}) = 65,$$

$$s_{T_2} = Z(20, 1 \mid \text{neutral, medium}) = 47.$$

Let us first examine the 65 designs at T_1,

$$S_{T_1} = \{201, \mathit{166}, \mathit{565}, \mathit{818}, 702, 335, 487, 471, 73, 331, \mathit{843}, 172, 139, 595, 945, 905, 156, 658, 649, 431, 969, 233, 130, 204, 307, 105, \mathit{840}, 29, 179, 189, 58, 305, 40, 38, 9, 525, 31, 286, 17, 366, 982, 914, 529, 655, 567, 828, 640, 621, 53, 301, 527, 924, 165, 459, 126, 597, 285, 643, \mathit{761}, 958, 681, 242, 379, 83, 927\};$$

[20]Each policy is generated as follows: a buffer size between 0 and 10 inclusive is generated for each m_i, $i = 1, \ldots, 10$. Thus, each design is a point sampled from the lattice Θ.

[21]The number of jobs 161,330 and 16,753 correspond respectively to 500, 1,000, and 50,000 standard clock ticks. Simulation up to 50,000 clock ticks is needed for the confidence intervals of the performance values of all designs to separate from each other. A standard clock tick is equivalent to an event happening to all 1,000 systems operating under all policies. See Vakili (Ref. 16) for further details about the standard clock.

we see that six designs (in italics) are included in S_{T_1}. At T_2, the 47 selected designs are

$$S_{T_2} = \{\mathit{761}, 595, \mathit{565}, 873, \mathit{843}, 139, 525, 105, \mathit{166}, \mathit{818}, 477, 643, 567, 447, 417, 980, 969, \mathit{234}, \mathit{928}, 366, 686, 201, 702, 738, 704, 111, 255, 314, 982, 361, \mathit{785}, 640, 773, 910, 901, 235, 455, \mathit{70}, 914, 172, 925, 335, 897, 31, \mathit{456}, 217, 176\};$$

we see that ten designs (in italics) from the good enough subset have been captured. It is also interesting to point out that, from our experiments, we have observed a very fast convergence of design orders. Dai has shown in his recent papers (Refs. 17–18) that ordinal comparisons indeed converge at a much faster rate for several major types of discrete event simulation. Similar results are also reported in Xie (Ref. 19).

Example 5.3. Picking in a Set of Random Processes. In this last example, we consider $N = 1{,}000$ random processes driven by the dynamical equation

$$X_i(t+1) = \max\{(1+r_i)X_i(t) + \sigma\omega, 0\}, \qquad i = 1, \ldots, N, \tag{18}$$

where $X_i(t)$ is the position of an object progressing in one direction (say, to the right) with unknown rate r_i. The object is constantly subject to bombardments which cause the object to displace either to the left or to the right. The bombardment is a Gaussian random variable with zero mean and variance σ^2, hence the last term of (18) in which ω is the standard Gaussian random variable. The initial positions are $X_i(0) = x_0 > 0$ for all objects. Moreover, the objects will never pass to the left of the origin; i.e.,

$$X_i(t) \geq 0, \qquad \text{for any } i = 1, \ldots, N \text{ and } t > 0.$$

Our good enough criterion is the set of g objects with the largest r_i. Notice that this is a maximization problem, but our selection results still apply. Since we can only observe the positions $X_i(t)$ and we assume that the measurements are exact, one possibility is to wait long enough until a time T^* such that, for all $t \geq T^*$, the growth of $X_i(t)$ dominates over the magnitude of the bombardment; consequently, by comparing the positions of $X_i(t)$, we can identify the good enough objects.[22] However, using the selection results that we have calculated in the previous section, we can also select a subset which contains some good enough objects at a much earlier time.

[22]This differs from the second example above in the sense that the signal is becoming stronger rather than the noise dissipating over time.

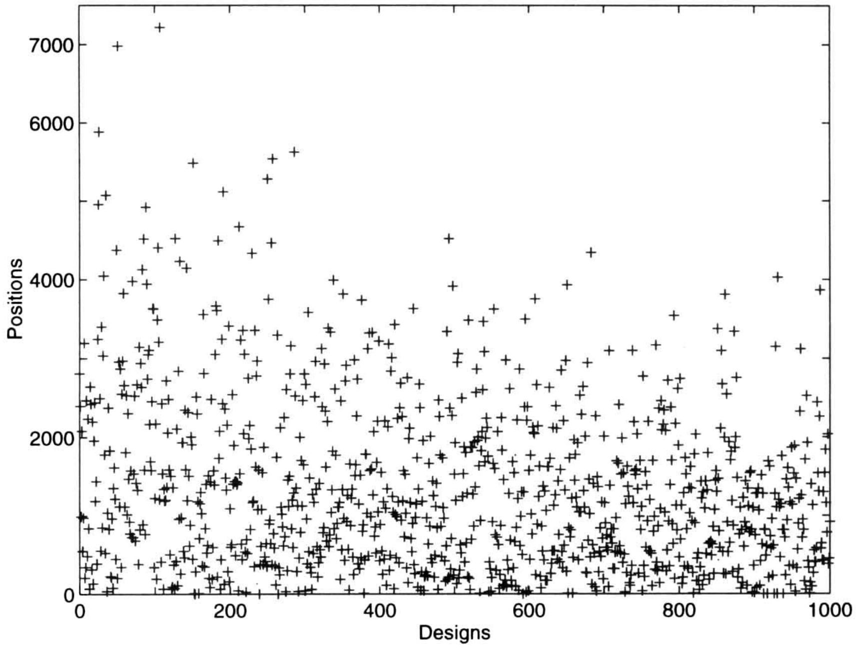

Fig. 10. Positions of the 1,000 objects at $t = t_{\text{short}}$.

In our experiments, all the r_i are chosen to be within 0.001 to 0.002 spaced equally; without loss of generality, we set the r_i in decreasing order such that $r_1 = 0.002$ and $r_N = 0.001$. We also set $\sigma^2 = 400$ and $x_0 = 100$. Although we have determined that $T^* > 5{,}000$ is needed, we considered picking a subset using the neutral OPC and the large noise range at $t = t_{\text{short}} = 1{,}000$. A realization snapshot showing the displacements of the 1,000 objects at t_{short} is shown in Fig. 10. Notice that the displacements have not separated enough to give a good resolution of the good enough objects. The realized alignment probabilities for $g = 30$ and $g = 50$ are plotted in Fig. 11, which are all within the range of 0.95, and again validates the correctness of the subset selection sizes.

6. Conclusions

In this paper, we have suggested a quantitative approach for search reduction using the framework of ordinal optimization. Subset sizes are calculated based on a two-parameter model via Monte Carlo studies. We have also demonstrated the uses of these quantitative results. The subset

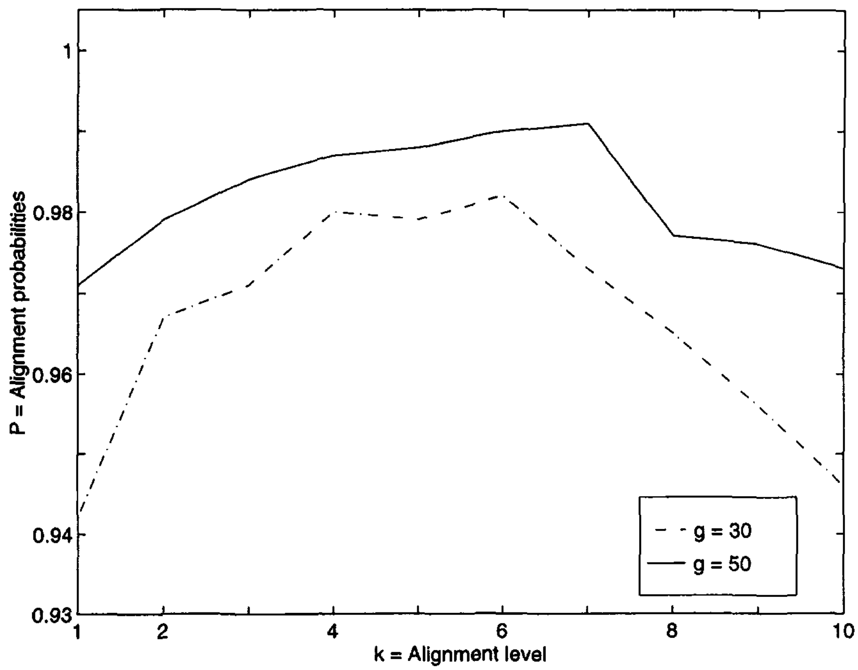

Fig. 11. Realized alignment probabilities for Example 5.3.

selection procedure is an important step in the initial search phase for many optimization algorithms. While more studies on the modeling and uses of subset sizes will require further research effort, other issues pertinent to the successful implementation of ordinal optimization warrant some mention:

(i) Uniform Sampling. Assumption 2.1 is a first starting point of studying various statistical issues in ordinal optimization. To truly perform uniform sampling can itself be a difficult task. It amounts to a sound understanding of the structure and/or the representation of the system under investigation. The simplest case, if possible, would be to reduce the search space to a hypercube or hypersphere in the Euclidean space, and there exists methods for uniform sampling in these cases [Zhigljavsky (Ref. 4)]. A more difficult problem is to uniformly sample in a region with equality and/or inequality constraints.

(ii) Good Enough Criterion. For minimization problems, if the underlying OPC is flat, then it may be reasonable to adopt a greater top $n\%$ as the good enough criterion, since there exist in fact more designs of relatively low performance values. This will enhance the alignment probabilities. On the other hand, if the performance curve is very steep, and thus it is very

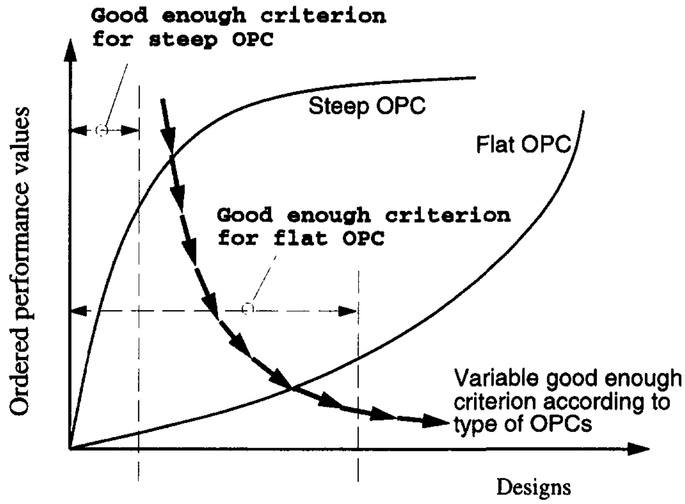

Fig. 12. Variable good enough criterion for different underlying OPCs.

difficult to obtain design samples of small performance values, then asking for the top 0.1% (say) designs would incur heavy sampling costs. Therefore, the good enough criterion should be flexibly chosen depending on the designer's knowledge of the underlying OPC, as suggested in Fig. 12. In addition, the membership criterion for the good enough subset bears a natural extension to the membership function employed in the theory of fuzzy sets. This may require future research attention in the artificial intelligence area.

(iii) Different Subset Selection Rules. We have considered two selection rules in this paper, namely, blind pick and horse race. As mentioned earlier, other possibilities such as round-robin tournament or pairwise comparison could induce different alignment results as well as costs of evaluation, and these are yet to be studied. Moreover, combination of selection rules (e.g., nested applications of selection rules during the course of simulation) is an interesting area to be explored.

(iv) Adaptive Search Schemes. As we have emphasized in Section 1, ordinal optimization is a useful complementary technique during the initial phase of many search problems, and it aims at narrowing down potentially promising and manageable subsets of solutions. Based on a selected subset, a designer can examine the properties of the solutions from a single run of ordinal optimization. Oftentimes, good insights can be obtained, which helps improving the quality of solution further.[23] Therefore, adaptive search

[23]A good example is discussed in Deng's recent work (Ref. 20).

schemes can be devised using the gained knowledge. Existing techniques such as genetic algorithms are natural candidates for this purpose. Strong human–machine interaction is also a worthwhile dimension for development. In this light, the alignment situations, i.e., the values g and k, could serve as important parameters in the adaptive schemes.

7. Appendix: OPC and Performance Density

Assume that the performance values of all designs θ in Θ (or $\hat{\Theta}$) are bounded above and below. As a result, there exists $J_{\min}$ and $J_{\max}$ which are the minimum and maximum values of $J(\theta)$ attained respectively by some $\underline{\theta}$ and $\bar{\theta}$ in Θ. Again, in our thought experiment, suppose that we had evaluated for all θ the performances $J(\theta)$ and we had constructed a histogram by appropriately tallying all $J(\cdot)$s between $J_{\min}$ and $J_{\max}$. We call this histogram the performance density, denoted by $\psi(y)$, $J_{\min} \leq y \leq J_{\max}$. Ho and Deng (Ref. 13) reported some performance densities for different instances of the travelling salesman problem. When the number of designs is very large, the performance density function can be approximated by a continuous curve. Notice that $\psi(y) = 0$ for $y < J_{\min}$ and $y > J_{\max}$. Next, consider shifting the curve from $J_{\min}$ to 0 on the abscissa axis, and scaling the abscissa axis by $1/(J_{\max} - J_{\min})$. By multiplying $\psi(\cdot)$ by the appropriate normalization constant, we can reduce the area under this transformed density function to unity, obtaining the standardized performance density function $f(y)$. The integral of $f(\cdot)$ is the standardized performance distribution, denoted by $F(\cdot)$, i.e.,

$$F(y) = \int_0^y f(z)\, dz, \tag{19}$$

where $F(y)$ is defined only for $y \in [0, 1]$. Notice that

$$f(y) = dF(y)/dy, \qquad \text{for } y \in (0, 1).$$

We can now draw the relationship between a standardized OPC $\Lambda(x)$ and a standardized performance distribution $F(y)$; notice that both are approximate models. For the latter, $F(y)$ is the proportion of design space, say x, which has scaled performance value less than or equal to y; i.e.,

$$x = F(y), \qquad \text{for } 0 \leq x \leq 1.$$

Since the standardized OPC $\Lambda(\cdot)$ is defined on [0, 1], we quickly see that

$$y = \Lambda(x) = F^{-1}(x), \tag{20}$$

for $x \in [0, 1]$. In other words, the ordered performance curve and the performance distribution are inverse mappings of each other.

An interesting property related to the standardized performance density is noted as follows. Under uniform sampling (cf. Assumption 2.1), the performance values of the N designs $J_1, J_2, \ldots, J_N$ can be thought of as variates generated from the performance density $\psi(\cdot)$. As discussed in Ho et al. (Ref. 1), the ordered performance values are the order statistics $J_{[1]}, J_{[2]}, \ldots, J_{[N]}$. With respect to the transformation stated above, i.e., $J_{[i]} \mapsto y_{[i]}$, David (Ref. 12) showed that

$$E[F(y_{[i]})] = i/(N+1).$$

In other words, the N values of $F(y_{[i]})$ divide the standardized performance density function into $N+1$ parts, each of which has an expected area of $1/(N+1)$. This result is useful in terms of determining the required sample size N. More information about order statistics can be found in Balakrishnan and Cohen (Ref. 21) and David (Ref. 12).

To date, extensive study is not found on the noise version of order statistics. The analysis in this paper can be considered as a first attempt. With respect to the noise model $\tilde{J}_i = J_i + \omega_i$, where $J_i \sim \psi(\cdot)$ and $\omega_i \sim \xi(\cdot)$, the ordered noisy performances,

$$\tilde{J}_{[1]} \leq \tilde{J}_{[2]} \leq \cdots \leq \tilde{J}_{[N]},$$

can be compared to the order statistics generated by the density $\tilde{\psi}(\cdot)$,

$$\tilde{\psi}(y) = \int \psi(y-s)\xi(s)\, ds.$$

That is, $\tilde{\psi}(\cdot)$ is the convolution between the performance density and the noise density. Further examination of this area is an important research direction.

References

1. Ho, Y. C., Sreenivas, R. S., and Vakili, P., *Ordinal Optimization in DEDS*, Journal of Discrete Event Dynamic Systems, Vol. 2, pp. 61–68, 1992.
2. Ho, Y. C., *Overview of Ordinal Optimization*, Proceedings of the 33rd IEEE Conference on Decision and Control, Lake Buena Vista, Florida, pp. 1975–1977, 1994.
3. Ahlswede, R., *Search Problems*, Wiley, New York, New York, 1987.
4. Zhagljavsky, A. A., *Theory of Global Random Search*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1991.

5. RUML, W., NGO, J. T., MARKS, J., and SHEIBER, S., *Easily Searched Encodings for Number Partitioning*, Journal of Optimization Theory and Applications, Vol. 89, pp. 251-291, 1996.
6. DENG, M., and HO, Y. C., *Sampling-Selection Method for Stochastic Optimization Problem*, Preprint, 1996.
7. VAKILI, P., MOLLAMUSTAFLAGLU, L., and HO, Y. C., *Massively Parallel Simulation of a Class of Discrete Event Systems*, Proceedings of the IEEE Frontiers MPC 1992 Symposium, Washington, DC, pp. 412-419, 1992.
8. PATSIS, N. T., CHEN, C. H., and LARSON, M. E., *SIMD Parallel Discrete Event Dynamic System Simulation*, IEEE Transactions on Control Systems Technology, Vol. 5, pp. 30-41, 1997.
9. GUPTA, S. S., and PANCHAPAKESAN, S., *Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations*, Wiley, New York, New York, 1979.
10. SANTNER, T. J., and TAMHANE, A. C., *Design of Experiments: Ranking and Selection*, Marcel Dekker, New York, New York, 1984.
11. GOLDSMAN, D., and NELSON, B., *Ranking, Selection, and Multiple Comparison in Computer Simulation*, Proceedings of the 1994 Winter Simulation Conference, 1994.
12. DAVID, H. A., *Order Statistics*, 2nd Edition, Wiley, New York, New York, 1982.
13. HO, Y. C., and DENG, M., *The Problem of Large Search Space in Stochastic Optimization*, Proceedings of the 1994 Conference in Decision and Control, pp. 1470-1475, 1994.
14. DENG, M., HO, Y. C., and HU, J. Q., *Effect of Correlated Estimation Errors in Ordinal Optimization*, Proceedings of the 1992 Winter Simulation Conference, pp. 466-475, 1992.
15. PEARSON, K., *Tables of the Incomplete Beta Function*, 2nd Edition, Cambridge University Press, London, England, 1968.
16. VAKILI, P., *A Standard Clock Technique for Efficient Simulation*, Operations Research Letters, Vol. 10, pp. 445-452, 1991.
17. DAI, L. Y., *Convergence Properties of Ordinal Comparison in the Simulation of Discrete Event Dynamic Systems*, Proceedings of the 34th IEEE Conference on Decision and Control, pp. 2604-2609, 1995.
18. DAI, L. Y., *The Effects of Correlation on Ordinal Comparison of Discrete Event Dynamic Systems*, Unpublished Manuscript, 1995.
19. XIE, X. L., *Dynamics and Convergence Rate of Ordinal Comparison of Stochastic Discrete Event Systems*, Unpublished Manuscript, 1995.
20. DENG, M., *Sampling-Selection and Space Narrowing Methods for Stochastic Optimization*, PhD Thesis, Division of Engineering and Applied Sciences, Harvard University, 1995.
21. BALAKRISHNAN, N., and COHEN, A. C., *Order Statistics and Inference*, Academic Press, New York, New York, 1991.