Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

A SIMULATION STUDY ON SAMPLING AND SELECTING UNDER FIXED COMPUTING BUDGET

Loo Hay Lee
Ek Peng Chew

Department of Industrial & Systems Engineering
National University of Singapore
10 Kent Ridge
Singapore 119260, SINGAPORE

ABSTRACT

For many real world problems, when the design space is huge and unstructured and time consuming simulation is needed to estimate the performance measure, it is important to decide how many designs should be sampled and how long the simulation should be run for each design alternative given that we only have a fixed amount of computing time. In this paper, we present a simulation study on how the distribution of the performance measure and the distribution of the estimation error/noise will affect the decision. From the analysis, it is observed that when the noise is bounded and if there is a high chance that we can get the smallest noise, then the decision will be to sample as many as possible, but if the noise is unbounded, then it will be important to reduce the level of the noise level by assigning more simulation time to each design alternative.

1 INTRODUCTION

The essence of any decision problem is an optimization problem. Whether it is a decision on the production schedule of a manufacturing plant, a decision on the routing rules of networking, or a decision on the make-up of investment portfolios, the problem always revolves around making a choice from a population of alternatives (either finite or infinite) so as to optimize the targeted objective. Without loss of generality, the objective can be summarized by the mathematical expression that is represented by the minimization form as

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

where Θ is the space of all potential solution candidates, or the search space; θ is a design alternative; and J is the performance measure of the decision problem. The problem as defined is not new. As a matter of fact, it has been the center of many theoretical works in mathematics. However, the complexity and the scale of most optimization problems encountered in the real-world settings today often render the application of traditional mathematical approaches inadequate.

There are two major challenges that we might face in the real world setting. First, evaluating the performance measure $J(\theta)$ for the design alternative θ is not a trivial task. In some cases, $J(\theta)$ can be evaluated in a closed form, but unfortunately in many problems, such a closed form expression cannot be found and a simulation model is the only tool available to estimate the performance measure for the design alternative. Moreover, when uncertainties exist in the system, we need to repeat the simulation N times in order to estimate the average performance. However, the accuracy of the estimate cannot improve any better than $1/\sqrt{N}$. Second, the design space Θ can be very huge and unstructured. Traditionally, analysis tools play important roles in numerical optimization, and one of the examples is perturbation analysis (PA) which is used to estimate the gradient for determining the local search direction. However, these tools will fail if the design space is unstructured. (For example the decision variables are not real numbers.) In such a case, brute-force evaluation for all the design alternatives will be needed in order to find the optimal design. However, when the choice of design alternatives within the design space experiences the effect of combinatorial explosion, brute-force evaluation will be impractical. As a result, we might need to randomly sample the design alternatives from the design space or use some AI optimization tools, for example, Genetic Algorithm, to help us locate the near-optimal designs.

The effects of these two challenges will multiply, and make the optimization problem even more intractable. In view of these difficulties, ordinal optimization (Ho et al. 1992) provides a strategic redirection for the optimization problem. In ordinal optimization we settle for the good enough alternatives with high probability instead of finding

the best designs with certainty. There are two important tenets for this strategic redirection:

1. The order converges exponentially fast while the value converges at a rate of $1/\sqrt{N}$. For example, it is easier to determine whether A> B than to estimate the difference between A and B.
2. The probability of getting something "good enough" increases exponentially with the size of the "good enough set"

The theoretical proofs of these two tenets can be found in Dai (1996), Xie (1997) and Lee et al. (1999).

Ordinal optimization has provided an important insight in tackling the problem, that is, if we are only asking for good enough designs, we should not be spending too much time to run the simulation for a single design. The remaining questions are now given a fixed computing budget, how many designs should be sampled from the design space and how long should the run of the simulation be for each design so as to maximize the chance of finding some good designs.

Ranking and selection has been an important research area in the simulation field as it addresses how to allocate computing resources to design alternatives so as to guarantee a certain level for the probability of correct selection. The indifference zone (IZ) selection procedure has been a common method to tackle the problem. Rinott (1978) developed a two-stage method based on this concept. In the first step, all the design alternatives will be allocated with a small number of replications of the simulation in order to estimate the variability of the systems. Then during the second stage, based on the indifference zone concept, the number of replications of the simulation that should be allocated for each design is computed so that the probability of correct selection is within the specified confidence level. However this method is only recommended for a small number of alternatives as the formula is calculated based on the least favorable configuration assumption that the best design is hard to be separated from the others.

Goldsman and Nelson (1998) and Nelson et al. (2001) have extended this method and used the idea of sample-screen-sample-select to tackle the problem. In the first stage, subset selection scheme is used to screen out the noncompetitive designs and then the indifference zone selection procedure is used to select the best design from the survivors of the screening process.

Chen (1995) provides another direction to address the problem; he formulates the process of selecting the best design as an optimization problem. The idea is to decide how to allocate the computing resources to all the designs so as to maximize the probability of correct selection. As the closed form expression for the probability of correct selection is difficult to compute, he approximates this probability by using Chernoff bound. Then the steepest-descent algorithm is applied to solve this approximated optimization problem (Chen et al. 1997, 1998, 2000).

All the research works mentioned above only deal with a fixed number of design alternatives. To our knowledge, the problem on how many designs to sample has not been addressed adequately. In this paper, we will provide some insights for this problem. In the following section, we will define the problem of sampling, ranking and selection. Simulation study of the problem will be presented in Section 3. Then in Section 4, we will show some application examples. Finally the conclusions will be made in Section 5.

2 SAMPLING, RANKING AND SELECTION

In the optimal computing budget allocation problem proposed by Chen (1995), he targets on maximizing the probability of correct selection given that the computing time is fixed. When the problem of sampling (how many designs to sample) is considered, we should revise the objective, and one of the reasonable objectives can be to maximize the expected true performance of the observed best design.

The revised optimal computing budget allocation problem is as follows:

$$\begin{aligned} &Max \quad E[J_{[\tilde{1}]}] \\ &\text{s.t.} \quad n_1 n_2 = \text{Computing Budget} \end{aligned} \tag{2}$$

$$\text{and} \quad \tilde{J} = J + \omega \tag{3}$$

where J is the true performance value; ω is the noise; $\tilde{J}$ is the observed performance value which is equal to the sum of the true performance value and the noise; the subscript $[i]$ is the design with true rank i; $[\tilde{i}]$ is the design which is observed as rank i when the designs are ranked according to the observed performance, $\tilde{J}$; n_1 is the number of designs sampled; and n_2 is the number of replications of the simulation allocated to each design. We assume that horse race selection method is used, i.e., every design will be given equal number of replications to run. It can be observed that when n_1 is large, a lot of designs will be sampled, and each design will be given only few replications to run. Although we might be able to sample some good designs, due to the large noises, we might not be able to correctly select those good designs. On the other hand if n_1 is small, only a few designs are sampled, and each design will be allocated with a lot of replications to run. As a result, we are able to select the top design within these n_1 sampled designs, but this top design might not be good because the good designs may not have been sampled. Therefore, it is believe that there should be a tradeoff between how many designs should be sampled and how long the simulation should be run.

3 SIMULATION STUDY

The problem stated in (2) is not easy to solve as there is no closed form expression for $E[J_{[\tilde{1}]}]$. However in order to understand how the different probability distributions of the performance measure and the noise will affect the allocation decision (i.e., n_1 = how many designs to sample and n_2 = how long the replications should run), we have performed some simulation studies. The following two subsections will describe and discuss the scenarios and the results of the studies.

3.1 Uniform Distribution

In this simulation study, we assume that :

1. The computing budget is fixed at 10000 replications.
2. The true performance value, J, for the designs that we sampled from the design space follows a uniform distribution U[0, $\sqrt{12}$] (the standard deviation equals to 1).
3. The noise, ω, follows a uniform distribution U[0, $\sqrt{12}\ \sigma_N$], where σ_N = { 5, 2, 1, 0.5, 0.1}.

In the study, we first sample n_1 designs from the distribution of the true performance, i.e., U[0, $\sqrt{12}$] . Then for each design we generate n_2 noises to represent the n_2 independent replicate runs for the design, and the average of these n_2 noises will be added to the true performance measure, J, which make up the observed performance value, $\tilde{J}$, for the design. The n_1 designs are then ranked based on the observed performance values, and the design which is ranked the best, its true performance value will be recorded. We repeat this experiment 1000 times in order to estimate the expected true performance value for the observed best design. The same procedure is repeated for different values of n_1 and n_2 (the values of n_1 and n_2 are chosen in such a way that $n_1 n_2$ =10000) and the results from the simulation studies are shown in Table 1 and Figure 1. The shaded cell in each row denotes the best decision in each scenario (different σ_N).

Table 1: Expected True Performance Value for the Observed Best for Uniform Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	0.2330	0.3685	0.4346	0.4488	0.2067	0.0492
2	0.1198	0.1680	0.1951	0.2141	0.1125	0.0306
1	0.0687	0.0888	0.1109	0.1178	0.0698	0.0216
0.5	0.0517	0.0532	0.0579	0.0677	0.0428	0.0157
0.1	0.0327	0.0158	0.0162	0.0178	0.0140	0.0067

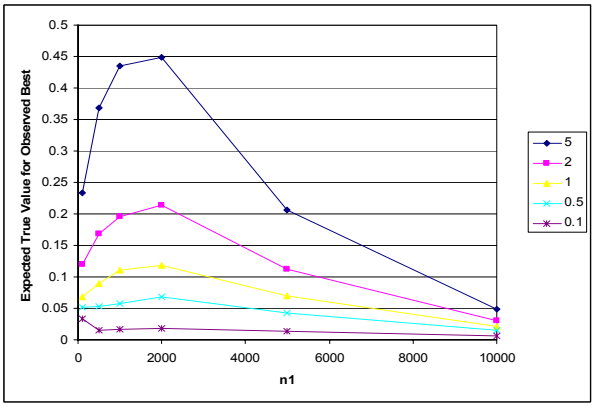

Figure 1: Expected True Performance Value for the Observed Best for Uniform Noise Case

From the table, it can be found that the best performance is always when n_1 equals to 10000 and n_2 equals to 1. In other words, "sampling more is better than doing more replications". The reason is because the distribution of the noise is uniform which has a finite bound at the left hand side region. When a good design has a noise coming from this left hand side region (i.e., very small noise), there will be no designs having observed ranks better than it unless the designs are indeed better or at least not worse than this good design. For illustration purpose, we assume that the noise follows the uniform distribution, U[0,1]. When a design with true performance value J_1 has a noise ω equal to δ, then only the design with a true performance value lower than $J_1 + \delta$ can have better observed performance than it. Since δ is small, this implies that $J_1 + \delta$ is indifferent with J_1 , and the designs with true performance values lower than $J_1 + \delta$ can also be viewed as the designs that are better or at least not worse than the design with true performance J_1. Therefore, for this situation, the best option is to sample as many as possible so as to increase the chance of getting good designs having noises coming from the left hand side region of the distribution (or having a very small noise).

In order to compare the effect of having different distributions on the noise, we repeat the same experiment, except that the distributions of the noise are varied. The following lists the different distributions of the noise used for the simulation runs:

- Normal : N(0, σ_N)
- Truncated Normal, N(0, σ_N) at $3\sigma_N$,
- Truncated Normal, N(0, σ_N) at $1\sigma_N$
- Exponential : Exp ($1/\sigma_N$)
- Negative Exponential : NegExp($1/\sigma_N$)

The results from the simulation runs are shown in Tables 2-6. Similarly the shaded cell in each row denotes the best decision in each scenario (different σ_N).

From Tables 2-6, we observe that depending on the distribution types that have been used to model the noise, some distributions will require more sampling for better

Table 2: Expected True Performance Value for the Observed Best for Normal Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	0.2424	0.3825	0.4803	0.6296	0.8482	0.9921
2	0.1176	0.1667	0.2195	0.2677	0.3984	0.5343
1	0.0714	0.0943	0.1193	0.1457	0.2055	0.2664
0.5	0.0507	0.0522	0.0635	0.0778	0.1047	0.1358
0.1	0.0354	0.0154	0.0171	0.0180	0.0262	0.0328

Table 3: Expected True Performance Value for the Observed Best for Truncated Normal at 3 σ_N Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	0.2363	0.3930	0.5159	0.6278	0.7228	0.3416
2	0.1153	0.1626	0.2106	0.2693	0.3352	0.1954
1	0.0746	0.0884	0.1096	0.1433	0.1734	0.1231
0.5	0.0492	0.0508	0.0617	0.0775	0.0971	0.0745
0.1	0.0356	0.0142	0.0162	0.0195	0.0246	0.0247

Table 4: Expected True Performance Value for the Observed Best for Truncated Normal at 1 σ_N Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	0.1436	0.2167	0.2597	0.2649	0.1651	0.0451
2	0.0761	0.0999	0.1189	0.1349	0.0847	0.0271
1	0.0533	0.0541	0.0647	0.0741	0.0537	0.0189
0.5	0.0400	0.0308	0.0361	0.0411	0.0321	0.0141
0.1	0.0356	0.0111	0.0108	0.0110	0.0105	0.0060

Table 5: Expected True Performance Value for the Observed Best for Exponential Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	0.2178	0.2542	0.2647	0.2024	0.0940	0.0254
2	0.1115	0.1254	0.1203	0.1030	0.0510	0.0163
1	0.0703	0.0738	0.0731	0.0618	0.0327	0.0114
0.5	0.0496	0.0434	0.0419	0.0377	0.0212	0.0079
0.1	0.0325	0.0147	0.0121	0.0112	0.0074	0.0037

Table 6: Expected True Performance Value for the Observed Best for Negative Exponential Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	0.2603	0.5257	0.7758	1.1230	1.4064	1.5714
2	0.1275	0.2233	0.3521	0.5574	1.0628	1.3422
1	0.0725	0.1183	0.1783	0.2958	0.6763	1.0508
0.5	0.0528	0.0661	0.0922	0.1445	0.3086	0.6726
0.1	0.0326	0.0173	0.0219	0.0293	0.0598	0.1058

performance (for example the exponential and the truncated normal at $1\sigma_N$), and as for the others, (for example, the negative exponential, the normal and the truncated normal at $3\sigma_N$), they require less sampling and more replications when the level of the noise is high. These two types of distributions give very different results; the reason is that for the exponential, the uniform and the truncated normal at $1\sigma_N$ distributions, they all have finite bounds at the left hand side of the distributions and the probability of having noises coming from this neighborhood is high. Hence it is always better to sample more so as to increase the chance of having more good designs. If we compare the results for different distributions, we will observe that the exponential distribution has the best performance, since the probability of getting the noise that comes from the neighborhood of the left hand side bound is higher when compared with the uniform and the truncated normal at $1\sigma_N$ distributions.

On the other hand, for the negative exponential, the normal and the truncated normal at $3\sigma_N$, they are either unbounded or bounded but with only a small probability of having noises from the neighborhood of the left hand side bounds. For this type of distributions, the suggestion is to have more replications so as to reduce the level of the noise. Similarly, we can also observe that the negative exponential distribution will have the worst performance, as there is a higher chance for a bad design obtaining a large negative noise which makes its observed performance better than all the good designs.

3.2 Normal Distribution

The same experiment in Section 3.1 is repeated for the case when the true performance value follows a standard normal distribution N(0,1). The case for the noise that follows normal distribution is shown in Table 7 and Figure 2.

From the Figure 2 and Table 7, we observe that there is a trade off between reducing noise (increasing replications) and increasing sample size. For the case when the level of the noise is high, it is recommended to reduce the level of the noise first by assigning more replications to run, but when the level of the noise is low, it will be better

Table 7: Expected True Performance Value for the Observed Best for Normal Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	-2.226	-2.054	-1.711	-1.415	-0.966	-0.757
2	-2.459	-2.784	-2.753	-2.566	-2.112	-1.719
1	-2.530	-2.954	-3.098	-3.166	-2.992	-2.742
0.5	-2.508	-3.032	-3.189	-3.357	-3.484	-3.416
0.1	-2.513	-3.046	-3.236	-3.448	-3.680	-3.829

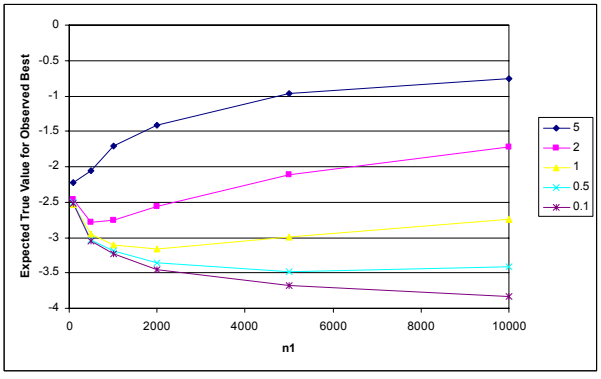

Figure 2: Expected True Performance Value for the Observed Best for Normal Noise Case

to sample more so as to obtain more good designs. Another observation is that when the level of the noise is higher than 0.5, the results suggest that it is always better to assign the number of replications of the simulation to all the designs so that the level of the noise can be reduced to the level around 0.5. In fact, the standard deviation of 0.5 for the noise is still quite high (about half of the standard deviation for the performance measure) which traditionally, we always want to assign more replications to increase the accuracy of the estimation. However, the results suggest otherwise; that is, more sampling rather than more replications. One of the reasons can be due to the first and second tenets provided by ordinal optimization, which says that the order is quite robust against the noise, and by enlarging the good enough set, the chance of getting good enough solutions will be increased. Sampling more in fact can be interpreted as enlarging the good enough set.

Similar to Section 3.1, we repeat the experiment for other distributions for the noise, and the results can be shown in Table 8-12.

From Tables 7-12, similar observations to Section 3.1 can be found. There are also two types of distributions which give different results. For the uniform, the exponential and the truncated normal at $1\sigma_N$ distributions, the results suggest to sample more. However, for the normal, the truncated normal at $3\sigma_N$, and the negative exponential distributions, we need to reduce down the level of the noise

Table 8: : Expected True Performance Value for the Observed Best for Uniform Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	-2.238	-2.016	-1.845	-1.726	-2.363	-3.028
2	-2.448	-2.780	-2.749	-2.704	-2.851	-3.265
1	-2.520	-2.972	-3.105	-3.186	-3.227	-3.458
0.5	-2.503	-2.999	-3.206	-3.348	-3.510	-3.626
0.1	-2.499	-3.044	-3.246	-3.428	-3.677	-3.843

Table 9: Expected True Performance Value for the Observed Best for Truncated Normal at 3 σ_N Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	-2.196	-2.027	-1.767	-1.430	-1.166	-1.848
2	-2.464	-2.753	-2.766	-2.583	-2.301	-2.458
1	-2.482	-2.968	-3.091	-3.116	-3.034	-2.984
0.5	-2.514	-3.030	-3.212	-3.368	-3.504	-3.476
0.1	-2.522	-3.036	-3.226	-3.436	-3.654	-3.832

Table 10: : Expected True Performance Value for the Observed Best for Truncated Normal at 1 σ_N Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	-2.393	-2.628	-2.506	-2.395	-2.596	-3.097
2	-2.504	-2.966	-3.050	-3.124	-3.175	-3.399
1	-2.513	-2.989	-3.179	-3.319	-3.487	-3.579
0.5	-2.529	-3.037	-3.234	-3.417	-3.631	-3.746
0.1	-2.480	-3.031	-3.247	-3.445	-3.697	-3.843

Table 11: Expected True Performance Value for the Observed Best for Exponential Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	-2.273	-2.311	-2.375	-2.512	-2.938	-3.294
2	-2.450	-2.843	-2.946	-3.025	-3.272	-3.516
1	-2.523	-2.974	-3.141	-3.250	-3.472	-3.667
0.5	-2.502	-3.000	-3.211	-3.373	-3.592	-3.738
0.1	-2.499	-3.045	-3.246	-3.428	-3.677	-3.851

Table 12: Expected True Performance Value for the Observed Best for Negative Exponential Noise Case

σ_N \ n_1	100	500	1000	2000	5000	10000
5	-2.223	-1.612	-1.128	-0.703	-0.307	-0.197
2	-2.440	-2.685	-2.356	-1.685	-0.743	-0.506
1	-2.520	-2.959	-3.031	-2.831	-1.656	-0.829
0.5	-2.502	-2.998	-3.194	-3.299	-3.006	-1.750
0.1	-2.499	-3.044	-3.246	-3.428	-3.674	-3.830

by assigning more replications of the simulation to run for each design. Similar to Section 3.1, the exponential distribution has the best performance while the negative exponential distribution has the worst performance.

Comparing Tables 1-6 with Tables 7-12, we have observed that the distribution of the true performance also plays an important role in deciding how many designs to sample and how many replications of the simulation to run. Generally, if the true performance follows a normal distribution, it will be better to sample more compared to the case when the true performance follows a uniform distribution. This is because when the true performance follows a normal distribution, the improvement on the sampled best designs is significant when we increase the sample size (in fact the best design for normal distribution is at -∞). However, for the uniform distribution, we will not expect any great improvement on the sampled best designs when the sample size increases.

4 APPLICATION EXAMPLES

In this section we will repeat the experiment in Section 3 except that instead of generating the performance directly from a distribution, we generate the designs uniformly (i.e., the values of decision variables will be uniformly generated). Two different functions are considered in these experiments, where one of them has a finite bound while the other does not have any bound.

4.1 Example 1

A one-dimensional Shekel function is considered:

$$f_1(x) = \sum_{i=1}^{10} \frac{1}{(k_i(x-a_i))^2 + c_i}, x \in [0,10] \qquad (4)$$

We randomly generate the values of the decision variable x from a uniform distribution U[0,10] and the estimation error is assumed to follow a normal distribution. Note that the standard deviation of the noise is adjusted so that σ_N can represent the ratio of the standard deviation of the noise to that of the performance distribution. The results from the simulation studies can be found in Table 13.

From Table 13, we found out that the results of this experiment are quite close to the results of Table 2 (the case of the true performance following a uniform distribution and the noise following a normal distribution). This is not surprising as the values for the function f_1 are always above zero, and the distribution for the performance function also follows a bounded distribution as shown in Figure 3.

Since the true performance values are bounded at the left hand side, the suggestion is to reduce the noise as much as possible so as not to miss any good designs from the horse race selection.

Table 13: Expected True Performance Value for the Observed Best for Shekel Function

σ_N \ n_1	100	500	1000	2000	5000	10000
5	0.3313	0.4055	0.4665	0.4974	0.6077	0.6646
2	0.2500	0.2956	0.3272	0.3536	0.4119	0.447
1	0.1948	0.2343	0.2576	0.274	0.3156	0.3594
0.5	0.1577	0.1761	0.1979	0.2165	0.2494	0.2786
0.1	0.1414	0.1324	0.1335	0.1366	0.1446	0.1526

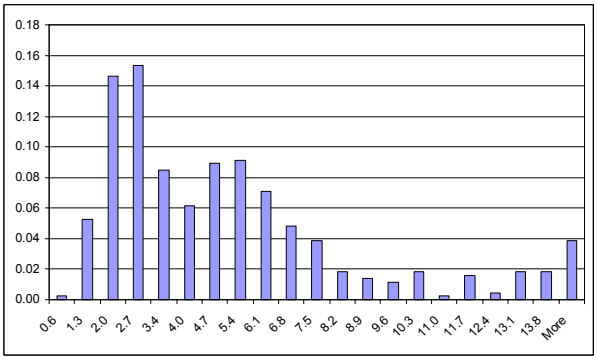

Figure 3: The Performance Distribution for the Shekel Function

4.2 Example 2

The same experiment is repeated except that the performance function is as follows:

$$f_2(x) = \begin{cases} \sqrt{\ln(x)} & if \quad 10 \geq x \geq 1 \\ -\sqrt{-\ln(x)} & if \quad 0 \leq x < 1 \end{cases} \qquad (5)$$

Similarly, we randomly generate the values of the decision variable x from a uniform distribution U[0,10] and the noise is assumed to follow a normal distribution. Note that the standard deviation of the noise is adjusted so as σ_N

will represent the ratio of the standard deviation of the noise to that of the performance distribution. The results from the simulation studies can be found in Table 14. Similar to Example 1, the results in Table 14 resemble that of the results in Table 7 (the case of both the true performance and the noise following normal distributions). This is because, similar to normal distribution, the performance distribution of the function f_2 is also unbounded at the left hand side of the distribution (see Figure 4), and the best design can have a value approaching - ∞. However, if we compare the results of this experiment with the results in Table 7 in more details, we would realize that it will suggest to sample more for this experiment. The reason is because the distribution of the performance function f_2 is skewed to the right and this means that we will have a higher chance to sample more good designs compared to the normal distribution. As a results, it will be more beneficial to sample more.

Table1 14: Expected True Performance Value for the Observed Best for the Performance Function $f_2(x)$

σ_N \ n_1	100	500	1000	2000	5000	10000
5	-2.132	-2.147	-2.008	-1.586	-0.727	-0.046
2	-2.324	-2.760	-2.779	-2.645	-2.405	-2.015
1	-2.338	-2.922	-3.074	-3.180	-3.082	-2.860
0.5	-2.369	-2.965	-3.183	-3.340	-3.508	-3.518
0.1	-2.335	-2.986	-3.192	-3.418	-3.665	-3.874

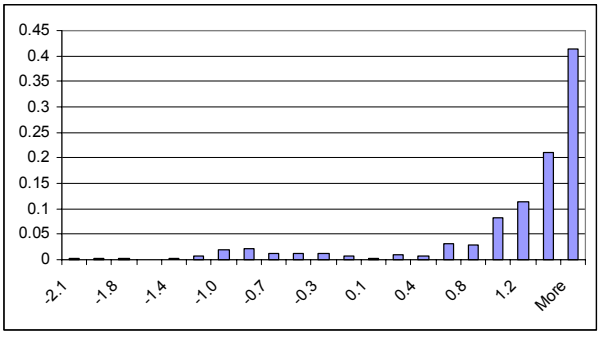

Figure 4: The Performance Distribution for the Function $f_2(x)$

5 CONCLUSIONS

In this paper, we have conducted a preliminary study on how to consider sampling when we have a fixed computing budget. Simulation experiments have been run for different scenarios, and it is found that the distribution properties of the true performance and the noise play an important role. If the probability that the noise comes from a neighborhood of the left hand side of the bound is high, then it is better to do more sampling, but if the noise is normal (or is unbounded), then it is better to reduce the level of the noise by assigning more time to run the simulation. Similarly, if the distribution for the performance is bounded at the left hand side, then it is advisable to spend time on running more replications, but if the distribution for the true performance is unbounded at the left hand side, then it will be better to sample more.

REFERENCES

Chen, C. H. 1995. An Effective Approach to Smartly Allocate Computing Budget for Discrete Event Simulation. *Proceedings of the 34th IEEE Conference on Decision and Control*, 2598-2605.

Chen, H. C., C. H. Chen , L. Dai, and E. Yücesan. 1997. New Development of Optimal Computing Budget Allocation For Discrete Event Simulation," *Proceedings of the 1997 Winter Simulation Conference*, 334-341

Chen, C. H., E. Yücesan, Y. Yuan, H. C. Chen, and L. DAI,. 1998. Computing Budget Allocation for Simulation Experiments with Different System Structures. *Proceedings of the 1998 Winter Simulation Conference*, 735-741

Chen, C. H., J. Lin, E. Yücesan and S.E. Chick. 2000. Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization. *Journal of Discrete Event Dynamic Systems: Theory and Applications*, 10, 251-270.

Dai, L. Y. 1996. Convergence Properties of Ordinal Comparison in the Simulation of Discrete Event Dynamic Systems. *Journal of Optimization Theory & Application*, 91, (2), 363-388.

Goldsman, D., B. L. Nelson. 1998. Comparing systems via Simulation. J. BANK ed. *The Handbook of Simulation*. John Wiley, New York, 273-306.

Ho, Y.C., R. Sreenivas, P. Vakili. 1992. Ordinal Optimization of Discrete Event Dynamic Systems. *Journal of Discrete Event Dynamic Systems,* 2, 61-88.

Lee, L. H., T. W. E. Lau, and Y. C. Ho. 1999. Explanation of Goal softening in Ordinal Optimization. *IEEE Transaction on Automatic Control*, 44, (1), 94-99.

Nelson, B. L., and J. Swann, D. Goldsman, and W. Song. 2001. Simple Procedures for Selecting the Best Simulated System When the Number of Alternatives is Large. *Operations Research,* 49, (6), 950-963

Rinott, Y. 1978. On Two-stage Selection Procedures and Related Probability Inequalities. *Communications in Statistics* A7, 799-811.

Xie, X.L. 1997. Dynamics and Convergence Rate of Ordinal Comparison of Stochastic Discrete Event Systems. *IEEE Transaction on Automatic Control*, 42, (4), 586-590.

AUTHOR BIOGRAPHIES

LOO HAY LEE is an Assistant Professor in the Department of Industrial and Systems Engineering, National University of Singapore, since 1997. He received his B.S (Electrical Engineering) degree from the National Taiwan University in 1992 and his S.M and PhD degrees in 1994 and 1997 from Harvard University. He is currently a member of IEEE, ORSS, and INFORMS. His research interests include simulation-based optimization, production scheduling and sequencing, logistics and supply chain planning, and vehicle routing. His email and web addresses are <iseleelh@nus.edu.sg> and www.ise.nus.edu.sg/staff/leelh/ respectively.

EK PENG CHEW is an Associate Professor and Deputy Head (Academic) in the Department of Industrial and Systems Engineering, National University of Singapore. He received his PhD degree from Georgia Institute of Technology. His research interests include logistics and inventory management, system modeling and simulation and system optimization. His email and web addresses are <isecep@nus.edu.sg> and www.ise.nus.edu.sg/staff/chewep/ respectively.