Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

A GRADIENT APPROACH FOR SMARTLY ALLOCATING COMPUTING BUDGET FOR DISCRETE EVENT SIMULATION

Chun-Hung Chen
Hsiao-Chang Chen

Department of Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104-6315, U.S.A.

Liyi Dai

Department of Systems Science and Mathematics
Washington University
St. Louis, MO 63130, U.S.A.

ABSTRACT

Simulation plays a vital role in analyzing many discrete-event systems. Usually, using simulation to solve such problems can be both expensive and time consuming. We present an effective approach to smartly allocate computing budget for discrete-event simulation. This approach can smartly determine the best simulation lengths for all simulation experiments and significantly reduce the total computation cost for obtaining the same confidence level. Numerical testing shows that our approach can obtain the same simulation quality with one-tenth the simulation effort.

1 INTRODUCTION

In order to efficiently manage and operate large man-made systems such as communication networks, traffic systems, and automated manufacturing plants, it is often necessary to apply extensive simulation to study their performance since no closed-form analytical solutions exist for such problems. Collectively, these types of systems are known as Discrete Event Systems (DES) (Ho 1991). Unfortunately, using simulation to solve such problems can be both expensive and time consuming due to their massive search space and their evolution in time according to complex man-made rules and the influence of random occurrences. In industry, with pressure to meet certain system specifications and only a limited budget to carry out necessary simulations, the limitations of traditional simulation technology can either delay a project or force it to go over budget.

Suppose we want to compare n different discrete-event systems (designs or alternatives), we do T simulation replications for all n designs (or alternatives). Totally, we need nT simulation replications. The simulation results become more accurate when T increases. If the accuracy requirement is not low (T is not small), and if the total number of designs in a decision problem is not small (n is large), then nT can be very large, which may easily make total simulation cost extremely high and preclude the feasibility of a simulation approach. To reduce total simulation time, one can either develop more efficient simulation technology or use faster computers to reduce the simulation time of each simulation experiment. In this paper, we present another approach to improve the overall simulation efficiency.

Our ideas are as follows. Intuitively, some bad designs can be discarded before completing all of the T replications. We don't have to waste efforts on simulating bad designs and so reduce overall simulation time. Then the question is how to systematically do this? When? And which designs? Ideally, we want to optimally choose the number of simulation replications for all designs to minimize the total simulation cost, while obtaining the desired confidence level. In fact, this question is equivalent to optimally decide which designs will receive computing budget for continuing simulation. Figure 1 illustrates the ideas by comparing a typical solution to this problem with the conventional approach using equal simulation lengths. Chen (1995) formulates this question and obtained promising preliminary results using very simple heuristics. In this paper, we will further discuss it and compare two approaches, one of which utilizes the gradient information.

To optimally allocate computing budget, first of all, one must have an efficient way to estimate the confidence level based on the results of the completed simulation. Further, one must have easy ways to anticipate how the confidence level will change if some computing budget is allocated and additional simulation replications are completed.

Goldsman and Nelson (1994) provide an excellent survey on current approaches (e.g., Goldsman, Nelson, and Schmeiser (1991), Gupta and Panchapakesan (1979), and Law and Kelton (1991)) to estimating simulation confidence level. In addition, Bechhofer, Santner, and Goldsman (1995) give a systematic and more detail discussion on this issue. Those approaches are mainly suitable for problems not having large number of designs

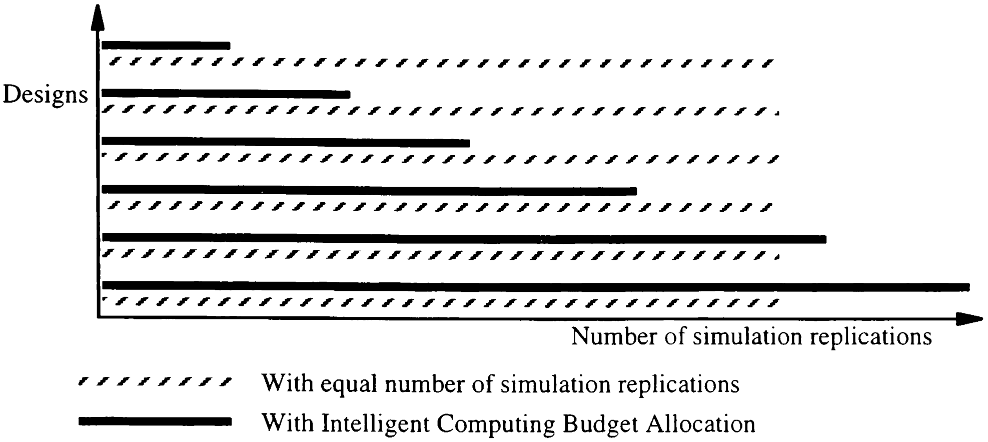

Figure 1: Comparison of Simulation Budget Allocations for Obtaining the Same Confidence Level

(e.g., Goldsman and Nelson (1994) suggest 2 to 20 designs). For real-life DES problems, the number of designs can grow up to numerous orders of magnitude easily. Chen (1996) presents a feasible way to quantify confidence level for *large* DES simulation (the "large" refers to large search space). Further, when the approach in Chen (1996) is applied the sensitivity information of the confidence level with respect to simulation replication numbers can be easily obtained, which will provide the basis to determine how to allocate computing budget among designs in this paper.

Section 2 describes the notation used in this paper and a brief overview of Chen (1996)'s approach to quantify confidence level for problems with large search space. In Section 3, we will define the "optimal computing budget allocation" problem. Two major difficulties in solving this problem will be pointed out. Since it is difficult to find an optimal solution for the computing budget allocation problem, and it is impractical to spend lot of time in finding the optimal solution, we propose a sequential approach to overcome these two difficulties in Sections 4 and 5. We call this approximation *smart computing budget allocation* scheme. Numerical testing in Section 6 shows that using this approach to smartly allocate computing budget can reduce the total computation time by about ten times for a 1000-design example. Section 7 concludes this paper.

2 A FEASIBLE APPROACH TO QUANTIFY CONFIDENCE LEVEL FOR PROBLEMS WITH LARGE SEARCH SPACE

Chen (1996) provides a simple approximation approach to quantify confidence level for problems with large search space and also provide some useful sensitivity information of the confidence level with respect to simulation replication numbers, which will provide the basis to determine how to allocate computing budget among designs in this paper. Denote

n: the total number of designs,

T: the length of simulation, the number of replication, or the total number of samples,

$\hat{J}_j(t)$: the t-th sample of the performance measure of design j,

$\bar{J}_j(T)$: the sample mean of design j, namely, $\bar{J}_j(T) = \frac{1}{T}\sum_{t=1}^{T}\hat{J}_j(t)$, and

J_j the performance measure, or more specifically, the mean performance measure of design j, i.e., the mean of $\hat{J}_j(t)$.

Assume that

i) $\hat{J}_j(t)$ is i.i.d. for all t,

ii) the simulations for designs i and j, $i \neq j$, are independent. Thus, $\hat{J}_i(t)$ and $\hat{J}_j(t)$ are independent.

For steady-state simulation, the sample $\hat{J}_j(t)$ may not be independent of $\hat{J}_j(s)$ for $s \neq t$. One possible way is to place the "raw" data in a few large batches, and work with these few batches as if they were independent (Banks, Carson, and Nelson 1995). As the strong law of large numbers, with probability 1,

$$\bar{J}_j(T) \rightarrow J_j, \text{ as } T \rightarrow \infty.$$

Without infinitely long simulations or infinite number of simulations replication, the sample mean $\bar{J}_j(T)$ is an approximation to J_j. We refer to the sample mean $\bar{J}_j(T)$ from one finite simulation experiment as an *observed* performance measure for a particular design's simulation. Let o_j be the index of the design having the j-th largest *observed* performance measure. With these notations, we have

$$\tilde{J}_{o_1} > \tilde{J}_{o_2} > \cdots > \tilde{J}_{o_n}.$$

The traditional optimization approaches are to find the design with maximum performance measure. (Without loosing generality, we only consider maximization problems in this paper.) However, even the simulation cost for a good estimation of J_j could be very high, especially for complicated systems. Instead of insisting on picking the best design, Ordinal Optimization (Ho, Sreenivas, and Vakili 1992) concentrates on finding good enough designs and reduces the required simulation time dramatically. Comparing the observed performance measures at short simulation length T, we can select the observed best design (o_1) or the observed top-r designs (o_1, o_2, , .., o_r), and then ask what is the probability that at least one of the observed top-r designs actually belongs in top-k. This is crucial to Ordinal Optimization, although estimation of such a probability is very difficult for problems having large n.

Chen (1996) adopts the Bayesian model to analyze such confidence probability. Under the Bayesian model, J_j is treated as a random variable and has a prior distribution which describes the knowledge or the subjective belief about J_j before any sampling. The posterior distribution is updated after we observe the samples $\{\hat{J}_j(t), t=1,..,T\}$. The posterior distribution p(J_j | $\{\hat{J}_j(t), t=1,..,T\}$) summarizes the statistical properties of J_j given the prior knowledge and sampling information. When simulation stops, the statistical properties is described by the posterior distributions. We can estimate the probability that J_j is in some specific region, e.g., Pr{ J_j>0 | $\{\hat{J}_j(t), t=1,..,T\}$ }, or compare two designs, e.g., Pr{J_i-J_j>0 | $\{\hat{J}_i(t), \hat{J}_j(t), t=1,..,T\}$}. For notational simplicity, we denote $\tilde{J}_j$ as the posteriori

$$J_j \mid \{\hat{J}_j(t), t=1,..,T\}.$$

Namely, Pr{ $\tilde{J}_j$>0} represents Pr{ $J_j > 0$ | $\{\hat{J}_j(t), t = 1, .., T\}$}. The posterior distribution p($\tilde{J}_j$) illustrates what value J_j may be, based on samples $\{\hat{J}_j(t), t=1,..,T\}$ and the prior knowledge. With some normal assumptions, the posterior

$$\tilde{J}_j \equiv J_j \mid \{\hat{J}_j(t), t = 1,2,\cdots,T\} \sim N\Big(\frac{1}{T}\sum_{t=1}^{T}\hat{J}_j(t), \frac{\sigma_j^2}{T}\Big)$$

Chen (1996) also shows that the Confidence Probability

CP1 ≡ Pr{At least one of designs o_1, o_2, .., o_r actually belongs in top-k}
≥ Approximated Confidence Probability

$$\text{ACP1} \equiv \prod_{j=r+k}^{n} \Pr\{\tilde{J}_{o_1} > \tilde{J}_{o_j}\},$$

and that

CP2 ≡ Pr{The true performance of the observed best design is not worse than β fraction of the performance of the true best design}
≥ Approximated Confidence Probability

$$\text{ACP2} \equiv \prod_{j=2}^{n} \Pr\{\tilde{J}_{o_1} > \beta\tilde{J}_{o_j}\}.$$

While CP1 and CP2 are very difficult to obtain, ACP1 and ACP2 can be computed very easily, and therefore will be used to approximate CP1 and CP2, respectively. Numerical testing shows that they can provide reasonably good approximation. Furthermore, since ACP1 and ACP2 are lower bounds of CP1 and CP2, we are sure that confidence level is sufficiently high when ACP1 or ACP2 is high enough. Although the definitions of CP1 and CP2 are different, the formulas for ACP1 and ACP2 are quite similar. For easy explanation, without loss of generality, we will only consider the simple case that

$$\text{ACP} \equiv \prod_{j=2}^{n} \Pr\{\tilde{J}_{o_1} > \tilde{J}_{o_j}\}$$

in the latter discussion, i.e., how to smartly allocate computing budget for obtaining satisfactory ACP.

3 PROBLEM DEFINITION

We follow the problem formulation given by Chen (1995). Let T_j be the simulation length, or the number of samples, of design j. If simulations are performed on a sequential computer and with simulation length T for all designs, the computation cost can be approximated by $T_1 + T_2 + \cdots + T_n = nT$. However, to ensure that ACP is larger than some value, we don't need to restrict ourselves to $T_1 = T_2 = \cdots = T_n$, and may choose different simulation lengths for different designs. This means T_i may not be equal to T_j for $i \neq j$. Furthermore, we can choose T_j for all j such that the total computation cost is minimized, while guaranteeing that ACP is greater than some satisfactory level. More specifically, we are considering the following problem.

$$\textbf{(P)} \quad \min_{T_1,\cdots,T_n} (T_1 + T_2 + \cdots + T_n),$$
$$\text{s.t. ACP} \geq \text{(a satisfactory level).}$$

There are two major difficulties in solving (P):

Difficulty 1. ACP($T_1, T_2, \cdots, T_n$) can be computed only after doing simulations until $T_1, T_2, \cdots, T_n$. Before performing simulations until $T_1, T_2, \cdots, T_n$, how can we predict or estimate the ACP at $T_1, T_2, \cdots, T_n$?

Difficulty 2. $T_1, T_2, \cdots, T_n$ are integers. Even if we have techniques to estimate ACP at $T_1, T_2, \cdots, T_n$, an extremely large combinatorial space must be searched to find a solution to (P), especially when n is large.

Note that the purpose of solving (P) is to further reduce computation cost for obtaining a desired confidence level. We should not exert too much effort solving (P) during simulation. Otherwise, the additional cost of solving (P) will cancel the benefits of computing budget allocation. Hence, we need to solve (P) very efficiently, even if this means obtaining a sub-optimal solution. Efficiency is more crucial than optimality in this application.

4 A SEQUENTIAL APPROACH

This section presents a sequential procedure to overcome the difficulties in solving (P). To Optimally allocate computing resource, it is equivalent to determine which designs we should do more simulation. We will sequentially decide it, although this is usually not an optimal solution any more.

Before doing simulation there is neither knowledge about ACP nor a basis for choosing T_j. First, all designs are simulated until length t_0 to obtain statistical information about sample means and sample variances. Then we try to determine how to further allocate computing budget using available statistical information. When simulation is stopped at t_0, the posterior distribution of design j is

$$\tilde{J}_j \equiv \tilde{J}_j(t_0) \equiv J_j \mid \{\hat{J}_j(t), t = 1, 2, \cdot\cdot, t_0\} \sim N\left(\frac{1}{t_0}\sum_{t=1}^{t_0} \hat{J}_j(t), \frac{\sigma_j^2}{t_0}\right)$$

At this moment, we have some ideas about each design and then can decide which designs are worthy of being allocated more computing budget. To determine how to further allocate computing budget, we have to be able to know how the ACP will change if some computing budget is allocated to some designs (Difficulty 1). More specifically, based on statistical information at t_0, we want to anticipate the posterior distribution at $t_0 + \Delta$, where Δ is a positive integer. To do this, we assume the sample mean and variance at t_0 are near those at $t_0 + \Delta$, and approximate the posterior distribution at $t_0 + \Delta$ using the *estimated* posterior distribution

$$N\left(\frac{1}{t_0}\sum_{t=1}^{t_0} \hat{J}_j(t), \frac{\sigma_j^2}{t_0 + \Delta}\right)$$

Note that the denominator of variance portion is $t_0 + \Delta$ rather than t_0. This approximation will be satisfactory assuming t_0 is large enough and if Δ is not too large. On the other hand, we don't want to choose t_0 too large, or we will defeat the purpose of this approach. Using the estimated posterior distributions, we can estimate the ACP at $t_0 + \Delta$ using the statistical information at t_0, and call it the "Estimated ACP" or EACP.

Similarly, when simulation proceeds until $T_1, T_2, \cdots, T_{i-1}, T_i, T_{i+1}, \cdots, T_n$, we can also use the available information to estimate how ACP will change if design i is given additional budget Δ, i.e., EACP($T_1, T_2, \cdots, T_{i-1}, T_i + \Delta, T_{i+1}, \cdots, T_n$). This is accomplished by using the estimated posterior distribution

$$N\left(\frac{1}{T_i}\sum_{t=1}^{T_i} \hat{J}_i(t), \frac{\sigma_i^2}{T_i + \Delta}\right)$$

for design i.

Now it is feasible to predict the ACP when the change of T_i's are not large. A possible sequential approximation approach to solving (P) is as follows. Since ACP will become larger as simulation proceeds, we sequentially add computing budget by b each time until ACP reached some satisfactory level (say P_{sat}). In order to minimize to the total computation cost, at each step, this budget b is allocated among some designs such that the EACP is maximized. Thus, at step k, k=1,2,3,..,

$$\textbf{(P-}k\textbf{)} \quad \max_{\tau_1^k, \cdots, \tau_n^k} \text{EACP}(T_1^k + \tau_1^k, T_2^k + \tau_2^k, \cdots, T_n^k + \tau_n^k),$$
$$\text{s.t.} \quad \tau_1^k + \tau_2^k + .. + \tau_n^k = b \text{ and } \tau_i^k \geq 0 \text{ for all } i.$$

More specifically, the sequential algorithm is

<u>A Sequential Algorithm for Smart Computing Budget Allocation</u> (SCBA)

```
Step 0.  PERFORM SIMULATION until length $t_0$
         for all designs,
         $k \leftarrow 0$,
         $T_1^k = T_2^k = \cdots = T_n^k = t_0$.
Step 1.  If ACP($T_1^k, T_2^k, \cdots, T_n^k$) $\geq$ $P_{sat}$, stop,
         otherwise, go to Step 2.
Step 2.  Solve (P-$k$),
         $T_i^{k+1} = T_i^k + \tau_i^k$, for $i$ = 1, .., $n$,
         $k \leftarrow k+1$,
Step 3.  PERFORM SIMULATION until
         ($T_1^k, T_2^k, \cdots, T_n^k$); go to Step 1.
```

Remarks:

1. Obviously, the computing budget allocated by this sequential approach is not the optimal way. As we discussed before, efficiency is more important than optimality. Otherwise, the additional cost of determining computing budget allocation may cancel its benefits.

2. b is the one-time increment of simulation budget. Small b means small step size and therefore will increase the total number of solving (P-k). On the other hand, large b may waste computing budget and result in a larger ACP.

5 SOLVING PROBLEM (P-k)

The next question is how to *efficiently* solve (P-k). While solving (P-k) is easier than solving (P), again, efficiency is much more important than optimality here. In this paper, we test two *quick and dirty* approaches to allocate the given computing budget b for obtaining large increment of ACP.

In the first approach, m designs are chosen and then the computing budget is equally distributed to them (each design has b/m). For each design, we calculate the anticipated increment of ACP if computing budget b/m is allocated to it. Then those designs are chosen if their anticipated ACP increments are among top-m.

Approach 1. Choose a positive integer m, and let $\Delta = b \,/\, m$ (assume Δ is an integer)

Step 1. For i =1, .., n, calculate $D_i \equiv$ EACP($T_1^k, T_2^k, \cdots, T_{i-1}^k, T_i^k + \Delta, T_{i+1}^k, \cdots, T_n^k$) - ACP($T_1^k, T_2^k, \cdots, T_{i-1}^k, T_i^k, T_{i+1}^k, \cdots, T_n^k$).

Step 2. Find the set $S(m) \equiv \{ \; i \; : \; D_i$ is within the top-highest-$m\}$

Step 3. $\tau_i^k = \Delta$, for all $i \in S(m)$.

Approach 2. In the second approach, instead of equally allocating computing budget among some m designs, we apply steepest-descent method (Luenberger 1984) to solve (P-k). We do the following approximation to estimate the gradient of ACP with respect to T_i.

Lemma 1. Suppose the random variable X ~ N(Δ, σ^2), where $\Delta > 0$. Then $\Pr\{X{<}0\} \le \exp(-\dfrac{\Delta^2}{2\sigma^2})$.

<pf> Using Chernoff bounds (Ross 1994), we have

$$\Pr\{X{<}0\} \le \inf_{t<0} \mathrm{M(t)} = \inf_{t<0} \exp(\frac{\sigma^2 t^2}{2} + \mu t).$$

Choose $t = -\dfrac{\mu}{\sigma^2}$, we have the minimum, i.e.,

$$\Pr\{X{<}0\} \le \exp(-\frac{\Delta^2}{2\sigma^2}). \qquad \text{Q.E.D.}$$

With this lemma,

$$\mathrm{ACP} = \prod_{j=2}^{n} \Pr\{\tilde{J}_{o_1} > \tilde{J}_{o_j}\} = \prod_{j=2}^{n} (1 - \Pr\{\tilde{J}_{o_1} - \tilde{J}_{o_j} < 0\})$$

$$\ge \prod_{j=2}^{n} \left(1 - \exp(-\frac{\Delta_{1j}^2}{2\sigma_{1j}^2}) \right) = \mathrm{ACP}^*,$$

where $\Delta_{1j} = \bar{J}_{o_1} - \bar{J}_{o_j}$ and $\sigma_{1j}^2 = \dfrac{\sigma_{o_1}^2}{T_{o_1}} + \dfrac{\sigma_{o_j}^2}{T_{o_j}}$.

For notation simplicity, let $s{=}o_1$. The gradient of ACP* with respect to T_i are as follows:

if $i \neq s$,

$$\frac{\partial}{\partial T_i} \mathrm{ACP}^* = \frac{\Delta_{1i}^2 T_s^2 \sigma_i^2}{2(T_s \sigma_i^2 + T_i \sigma_s^2)^2} \exp\left(-\frac{\Delta_{1i}^2}{2(\frac{\sigma_s^2}{T_s} + \frac{\sigma_i^2}{T_i})} \right)$$

$$\bullet \prod_{\substack{j=1 \\ j \neq s, j \neq i}}^{n} \left(1 - \exp(-\frac{\Delta_{1i}^2}{2(\frac{\sigma_s^2}{T_s} + \frac{\sigma_j^2}{T_j})}) \right)$$

if $i = s$, $\dfrac{\partial}{\partial T_s} \mathrm{ACP}^* =$

$$\sum_{i=2}^{n} \left\{ \frac{\Delta_{1i}^2 T_{o_i}^2 \sigma_s^2}{2(T_s \sigma_{o_i}^2 + T_{o_i} \sigma_s^2)^2} \exp\left(-\frac{\Delta_{1i}^2}{2(\frac{\sigma_s^2}{T_s} + \frac{\sigma_{o_i}^2}{T_{o_i}})} \right) \right.$$

$$\left. \bullet \prod_{j=2, j \neq o_i}^{n} \left(1 - \exp(-\frac{\Delta_{1i}^2}{2(\frac{\sigma_s^2}{T_s} + \frac{\sigma_{o_j}^2}{T_{o_j}})}) \right) \right\}$$

To avoid spending much time in iteratively finding the solution of (P-k), we only do a very limited numbers of iterations when applying steepest-descent method. Different numbers of iterations are tested in Section 6.

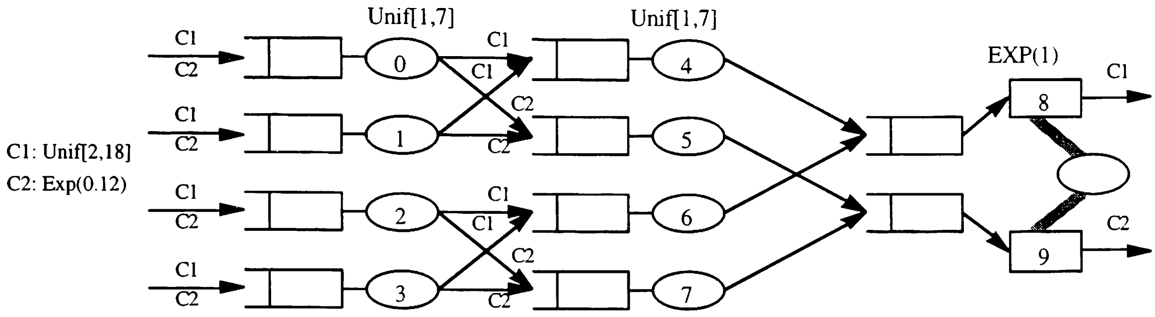

Figure 2: 10-Node Network with Priority, Interruption, and Shared Server

6 NUMERICAL TESTING

Two examples are tested. Example 1 is a steady-state simulation, and Example 2 is a terminating simulation.

Example 1. We consider a 10-node network (Please see Figure 2). Such a network could be the model for a large number of real-world systems, such as a manufacturing system, and a communication or a traffic network. For details about this example, please refer to Chen and Ho (1995). We consider the problem of optimally allocating 22 buffer units among the 10 different nodes for maximizing the throughput. Priority, interruption, blocking and multi-classes are included in this network. We denote the buffer size of node i by B_i. We set some constraints for symmetry reasons: $B_0 = B_1 = B_2 = B_3$, $B_4 = B_6$, and $B_5 = B_7$. In addition, B_8 & $B_9 \geq 1$. These constraints limit our search space to n=1000 different configurations.

The Standard Clock method (Chen and Ho 1995 and Vakili 1991), which is an efficient technique for DES simulation, is used to simulate this system. The computation cost for one design is roughly proportional to the number of clock ticks (for Standard Clock method, one event is generated at each clock tick). We define the computation cost as

$$\frac{1}{1000}\sum_{j=1}^{1000} \text{[the number of clock ticks when the simulation of design j is stopped),}$$

The $\frac{1}{1000}$ is used to rescale the cost. The computation cost for determining the smart computing budget allocation is so small as compared with the simulation cost that we ignore this portion. In this testing, we consider CP1 = Pr{At least one of the observed top-3 designs actually belongs in top-3} and CP2 = Pr{The true performance of the observed best design is not worse than 99.6% of the performance of the true best design}. We test Approach 1 and set $t_0 = 5$, $m = 25$, and $b = 125$. We repeat this testing 50 times. Each run has a different random seed. We consider the computation costs for different satisfactory confidence levels. Tables 1 and 2 contain the testing results for CP1 and CP2 respectively. The computation costs in these two tables are the average costs in the 50 testing runs.

Table 1: Speedup with the SCBA Method for ACP1

P_{sat}	without SCBA	with SCBA	Speedup
50%	24200	3952.5	6.12
60%	29100	4378.7	6.64
70%	45700	4670.0	9.78
80%	65600	5775.0	11.35

Table 2: Speedup with the SCBA Method for ACP2

P_{sat}	without SCBA	with SCBA	Speedup
50%	23000	3737.5	6.15
60%	29400	4105.0	7.16
70%	37300	4480.0	8.32
80%	54400	5446.2	9.98

Example 2. To further compare these two approaches, we test a simple single-node queue. The interarrival time is ~ Uniform[0.1, 1.9]. We consider 10 designs with different service times, which are Uniform[0.1, 1.85-i*0.05] for design i, i=1,..,10. Suppose we are interested in the average system time of the customers served between time 0 and time 10. Although the derivations in Sections 2 ~5 focus on maximization, we only need to reverse their inequality signs in order to apply to this minimization problem. We set b = 12. and t_0 = 10. 10,000 independent experiments are done to estimate the average cost for using different approaches. We consider CP = Pr{The observed best design is actually the true best design}. Table 3 shows the average total numbers

of simulation replications for obtaining the confidence level P_{sat} when setting t_0 = 10, and Table 4 is for t_0 = 5.

Table 3: Average Total Number of Simulation Replications for Different SCBA Approaches (T_0=10)

P_{sat}	60%	70%	80%	90%
App. 1 (m = 4)	172.4	230.3	350.0	608.4
App. 1 (m = 3)	170.5	225.9	322.5	542.7
App. 1 (m = 2)	170.0	226.5	322.5	511.9
App. 1 (m = 1)	176.6	228.9	324.6	509.6
App. 2 (1 itrn)	167.2	221.1	324.1	525.9
App. 2 (2 itrns)	162.2	212.0	307.7	498.6
App. 2 (4 itrns)	163.3	213.4	313.4	511.6
Without SCBA	327.1	488.2	796.4	1458.

Table 4: Average Total Number of Simulation Replications for Different SCBA Approaches (T_0=5)

P_{sat}	60%	70%	80%	90%
App. 1 (m = 4)	127.6	186.3	305.6	576.9
App. 1 (m = 3)	127.6	183.6	287.0	507.1
App. 1 (m = 2)	129.9	182.4	275.4	470.0
App. 1 (m = 1)	139.9	192.3	283.6	470.1
App. 2 (1 itrn)	130.1	185.9	290.4	496.2
App. 2 (2 itrns)	122.3	175.3	273.1	467.3
App. 2 (4 itrns)	125.9	185.3	285.5	492.1
Without SCBA	305.2	475.5	790.0	1462.

From Tables 3 and 4, we have the following observations:

- t_0 may affect the performance quite significantly, particularly when P_{sat} is small. How to choose an appropriate t_0 is problem-specific. This remains to be investigated.
- Different choices of m's obtain different performances. It is interesting to note that large m works better for low P_{sat}, while small m performs well for high P_{sat}. We conjecture that there exist some better ways which dynamically change m through simulation.
- For the steepest descent method, two iterations in each sequential optimization step works better than others. We need more testing to justify this. Ideally, we may gradually change the number of iterations through simulation to optimize the performance.
- The time savings factor of using SCBA increases as P_{sat} increases. This makes sense since we have more space to manipulate the allocation of computing budget when P_{sat}, the confidence level requirement, is higher.
- When t_0 = 5, Approach 2 with two iterations can reduce computation effort by 68% for P_{sat} = 90%. We believe that this time savings factor will be even larger if higher confidence level is required.

Since ACP is a lower bound of the confidence level CP and we use ACP to determine computing budget allocation, people may be concerned with the ending CP (actual confidence). In this testing, CP = Pr{The observed best design is actually the true best design} can be obtained numerically by calculating (total number of simulation experiments in which the observed best design is actually the true best design) / 10,000. Table 5 shows the numerical results of CP's for t_0 = 5.

Table 5: CP for Different SCBA Approaches (t_0=5)

P_{sat}	60%	70%	80%	90%
App. 1 (m = 4)	.629	.714	.815	.920
App. 1 (m = 3)	.630	.724	.817	.920
App. 1 (m = 2)	.634	.720	.813	.912
App. 1 (m = 1)	.632	.717	.804	.902
App. 2 (1 itrn)	.627	.724	.818	.918
App. 2 (2 itrns)	.627	.714	.816	.917
App. 2 (4 itrns)	.632	.713	.811	.919
Without SCBA	.696	.780	.879	.951

Our approaches stop simulation when ACP is no less than P_{sat}. We anticipate that the ending CP will be higher than P_{sat} since ACP is a lower bound of CP. Table 5 shows that CP's are not much higher than ACP's except the case in which SCBA is not used. The reason is that without using SCBA, all designs receive computing budget so that the simulation quality improvement is higher at each step. Consequently the ending CP's can be much higher than required level.

7 CONCLUDING REMARKS

In this paper we present two approaches to smartly allocate computing budget for DES simulation. Preliminary numerical testing shows that we can significantly reduce total computation cost. For real-time application problems, we have only a limited computing budget to carry out simulation. The SCBA can be applied to these problems to maximize the utilization of limited budget and obtain higher confidence level. In particular, from Table 1, the computation cost is 65,600 units for ensuring ACP1 > 80% without SCBA. On the other hand, with SCBA the computation cost is only 5,775 units. This implies that 5,775 SCBA computation units can obtain the same simulation quality as 65,600 computation units without SCBA. Application of the SCBA algorithm can obtain the same simulation quality with one-tenth the simulation effort.

In this paper, we compare two simple approaches using an example. The gradient approach performs slightly better than the other. While which approach is surprier needs more testing to justify, we fiirmly believe that there exists some more sopfisticaed way to accomplish better performance. We will test more examples in the future. The approaches using second order information is also one of the ongoing research topics.

ACKNOWLEDGMENTS

The work in this paper is partially supported by the University of Pennsylvania Research Foundation. The author would like to thank Dr. Russell R. Barton, Dr. Douglas J. Morrice, and the anonymous referees for their helpful suggestions and valuable comments.

REFERENCES

Banks, J., and J. S. Carson, B. L. Nelson. 1995. *Discrete-Event System Simulation*, Prentice-Hall.

Bechhofer, R. E., T. J. Santner, and D. M. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*, John Wiley & Sons, Inc.

Casella, G., and R. L. Berger. 1990. *Statistical Inference*, Wadsworth.

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, C. H. 1996. "A Lower Bound for the Correct Subset-Selection Probability and Its Application to Discrete Event System Simulations," To appear on *IEEE Transactions on Automatic Control.*

Chen, C. H., and Y. C. Ho. 1995. "An Approximation Approach of the Standard Clock Method for General Discrete Event Simulation," *IEEE Transactions on Control Systems Technology*, Vol. 3, #3, 309-317.

Goldsman, G., and B. L. Nelson. 1994. "Ranking, Selection, and Multiple Comparison in Computer Simulation," *Proceedings of the 1994 Winter Simulation Conference*, 192-199.

Goldsman, G., B. L. Nelson, and B. Schmeiser. 1991. "Methods for Selecting the Best System," *Proceedings of the 1991 Winter Simulation Conference*, 177-186.

Gupta, S. S. and S. Panchapakesan. 1979. *Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations*, John Wiley.

Ho, Y. C. (Editor). 1991. *Discrete Event Dynamic Systems*, IEEE Press.

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

Law, A. M. and W. D. Kelton. 1991. *Simulation Modeling & Analysis*, McGraw-Hill, Inc.

Luenberger, D. G. 1984. *Linear and Nonlinear Programming*, Addison-Wesley.

Ross, S. 1994. *A First Course in Probability*, Prentice Hall.

Vakili, P. 1991. "A Standard Clock Technique for Efficient Simulation, "*Operations Research Letters*, Vol. 10, 445-452.

AUTHOR BIOGRAPHIES

CHUN-HUNG CHEN is an Assistant Professor of Systems Engineering at the University of Pennsylvania, Philadelphia, PA. He received his Ph.D. degree in Simulation and Decision from Harvard University in 1994. His research interests cover a wide range of areas in Monte Carlo simulation, optimal control, stochastic decision processes, ordinal optimization, perturbation analysis, and their applications to manufacturing systems. Dr. Chen won the 1994 Harvard University Eliahu I. Jury Award for the best thesis in the field of control. He is also one of the recipients of the 1992 MasPar Parallel Computer Challenge Award.

HSIAO-CHANG CHEN is a Ph.D. candidate at the Systems Engineering Department, University of Pennsylvania. He received a double B.S. degree in Mathematics and Computer Science from the Eastern Michigan University in 1992, and he received an M.S. degree in Systems Science and Mathematics from Washington University, St. Louis in 1994. He is working on developing efficient approaches for discrete-event simulation.

LIYI DAI is an assistant professor in the Department of Systems Science and Mathematics at Washington University, MO. He received the M.S. degree from the Institute of Systems Science, Academia Sinica, Beijing, China, in 1986, and the Ph.D. degree from Harvard University in 1993. His research interests include discrete event dynamic systems, simulation, stochastic optimization, communication systems, and singular systems. He has coauthored over 30 papers in various journals and is the author of Singular Control Systems (Berlin: Springer-Verlag, 1989). Dr. Dai is listed in Who's Who among Asian Americans and is a recipient of the NSF CAREER award.