Proceedings of the 40th IEEE
Conference on Decision and Control
Orlando, Florida USA, December 2001

Search Space Reduction in Ordinal Optimization for Performance Evaluation of DEDS

Kyung-Joon Park and Chong-Ho Choi [1]

Abstract

Simulation plays a vital role in analyzing DEDS. However, using simulation to analyze complex systems can be time-consuming and expensive. Particularly, in the case of precise performance evaluation, computing budget, time constraint, and pseudo-random number generator limitations can become prohibitive. Ordinal optimization is an effective approach for improving the efficiency of simulation and optimization of DEDS. However, the ordinal optimization approach does not pay much attention to the problem of large search space. In this paper, for the reduction of search space, we propose a combined approach, ordinal optimization with orthogonal arrays. With this approach, the problem of large search space in stochastic optimization can become more manageable. Consequently, the proposed method can be more efficient for the reduction of simulation burden compared to the conventional ordinal optimization method.

1 Introduction

Most systems have been analyzed successfully based on calculus and differential equations, but man-made systems of recent invention are not easily tractable by calculus-based methods. Large artificial systems such as communications networks, automated manufacturing plants, and computer networks belong to this class of systems called Discrete Event Dynamic Systems (DEDS) [1]. For performance evaluation of DEDS, computer simulation is the only general purpose tool because no analytical methods exist for those problems. However, in the case of precise performance evaluation using computer simulation, computing budget, time constraint, and pseudo-random number generation often become prohibitive.

Many approaches have been proposed to resolve this problem. One of the most successful methods is importance sampling (IS), which can speed up simulation in network systems significantly [13]. IS is based on the notion of "biasing" the underlying probability mass in such a way that the events occur much more frequently. However, if this "biased" probability mass excludes the occurrence of the event of interest, the estimated value will be wrong, and this situation usually occurs when we have insufficient prior knowledge of system behavior [15]. Most of the existing simulation methods, including the IS-based method, concentrate on how to obtain estimations of the system performance more accurately with less effort. But it is very difficult, if not impossible, to pursue less effort and more accuracy simultaneuosly, and there is a trade-off between the two goals.

Ordinal optimization [10] is a recent approach for improving the efficiency of simulation and optimization of DEDS, which can resolve the problems of conventional computer simulation. Ordinal optimization can achieve simulation reduction by changing the notion of optimization. The idea of ordinal optimization is mainly based on two tenets. First, ordinal optimization focuses on the "order" instead of the "value" of the system performance. With this approach, we can reduce the simulation burden dramatically [10] [9]. And secondly, ordinal optimization seeks for "good-enough" solutions, not optimal solutions [12]. This is usually referred to as "goal softening", which means changing the goal from optimal solutions to "good-enough" solutions. Here, the notion of "good-enough" should be well defined [12]. However, ordinal optimization does not pay much attention to the problem of large search space. Many search spaces in stochastic optimization are combinatorially large [8]. For example, if we have r continuous parameters each discretized to k values, then the total design possibilities are k^r. In such a large search space, to get within the top-1% of a search space is no comfort at all. The solution within the top-1% of a search space of 10^{10} still can be 10^8 away from the optimal solution, which can be very bad.

In this paper, we suggest a novel approach for overcoming the problem of large search space in stochastic optimization. The ordinal optimization approach reduces simulation burden dramatically by shortening the simulation time. But, ordinal optimization does not consider the reduction of the given search space. We introduce here a combined aprroach, ordinal optimization with orthogonal arrays for the reduction of search space. Orthogonal arrays are a very effective method usually used for robust product design [4] [14] [5]. Instead of searching for the full search space, the orthogonal array is a method that only requires a frac-

[1]School of Electrical Engineering & Computer Science and ASRI, Seoul National University, Seoul, KOREA. Emails: {kjpark, chchoi@csl.snu.ac.kr}. Research supported in part by the Brain Korea 21 Program of the Korea Ministry of Education.

0-7803-7061-9/01/$10.00 © 2001 IEEE

tion of the search space. Orthogonal refers to the balance of the various combinations of values of variables so that no one variable is given more or less weight in the experiment than the other variables. Orthogonal also means that the effect of each variable can be mathematically assessed independently of the effects of other variables. Together with orthogonal arrays, the analysis of means (ANOM) [5] is used to find the optimal solution. ANOM employs a simple averaging of the output of each row in the orthogonal array to find the effects of each variable. With the result of ANOM, we can estimate the optimal solution by using only a fraction of the search space.

In the following section, we briefly explain the basics of orthogonal arrays and the analysis of means. In Section 3, we present the proposed method explicitly,and in Section 4, we derive some theoretical result on the proposed method. In Section 5, we give a simulation result of the proposed approach. Finally in Section 6, we discuss the usefulness of the proposed method.

2 Preliminaries

In this section, we introduce orthogonal arrays and the analysis of means, which are very effective for reducing the number of experiments in product design [5]. The orthogonal array (OA) is a method of setting up experiments that only requires a fraction of the full factorial combinations. The treatment combinations are chosen to provide sufficient information to determine the factor effects using the analysis of means (ANOM). Here, "orthogonal" refers to the balance of the various combinations of variables so that no one variable is given more or less weight in the experiment than the other variables. "Orthogonal" also refers to the fact that the effect of each variable can be mathematically assessed independently of the effects of the other variables.

Table 1 is an example of an orthogonal array that would accommodate a seven-variable experiment. In Table 1, there are seven varaibles (A to G) and each variable has two levels {1,2}. If we use the full factorial method to find the optimal combination of variables, we need to experiment with 128 ($= 2^7$) runs, whereas the orthogonal array allows us to experiment with only eight runs. The procedure of using the orthogonal array is quite simple. As shown in Table 1, in the first run, all the variables are set to *level 1*, and in the second run, variables A to C are set to *level 1* and variables D to G are set to *level 2*, and so on. This array also can be used to illustrate the concept of orthogonality. *level 1* and *level 2* occur the same number of times in each column in the array. Furthermore, for the four rows with *level 1* in column A, two rows have *level 1* in column B and two rows have *level 2* in column B. The same can be said for the four rows with *level 2* in column A. in fact, the balance of variable levels can be found for every pair of columns in the array. This balance is also illustrated for the first two columns in Table 1. When we compare *level 1* of A to *level 2* of A, the effect of B in *level 1* of A is the same as the effect of B in *level 2* of A.

Table 1: Example of an orthogonal array.

Run	A	B	C	D	E	F	G
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

We can compose numerous OAs, and in general, the OA is represented in $La(b^c)$ form. Here a represents the number of runs to be performed, b is the number of levels of each variable, and c is the number of variables. The OA in Table 1 is represented as $L8(2^7)$ or simply as $L8$. To determine the importance of variables using the experimental results of OA, we use the analysis of means (ANOM) method. In the OA, the orthogonality guarantees that all the levels are tested equally and the influences from other variables are assumed to be equal. Therefore, the means of different levels of a variable are compared to determine the appropriate level of that variable. In the following section, we introduce the proposed method, ordinal optimization with orthogonal arrays.

3 Ordinal Optimization with Orthogonal Arrays

For the explanation of the proposed method, let's consider a three-node tandem queueing network with Poisson arrival and exponential service time at each node, which is reported in [9]. Each node has a finite system capacity of $K = 10$ customers (buffer plus cutomer in service). Clearly, Node 1 shows M/M/1/10 queue behavior, but the arrivals at Node 2 and Node 3 are not Poisson due to buffer overflows.

Each Node i is assigned an exponential service rate $\mu_i = 1/\rho$, where $\rho \in \{0.50, 0.51, ..., 0.58\}$. Here we adopt a search space of size $9 \cdot 9 \cdot 9 = 729$ instead of $10 \cdot 10 \cdot 10 = 1000$ to apply the proposed method more efficiently. Compared with the original problem in [9], the worst case, or $\rho = 0.59$, is dicarded. Since we can easily know this is the worst case, this reduction of search space is reasonable. It is inefficient, though possible, to apply the orthogonal array directly to this problem. Instead, we suggest an iterative orthogonal array method, in which the orthgonal arrays are applied iteratively in two "stages." In the first stage, we group each μ_i, $i = 1, 2, 3$ into three subsets, which are

Table 2: $L9(3^4)$ orthogonal array.

Run	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

$\{1/(0.51+0.01k),\ k=-1,0,1\}$, $\{1/(0.54+0.01k),\ k=-1,0,1\}$, and $\{1/(0.57+0.01k),\ k=-1,0,1\}$, respectively. We take the representative value of each subset with the value for $k=0$. In the first stage of simulation, we make the orthogonal array with these representative values. And, if we let $\overline{\mu_i} = 1/\overline{\rho_i}$ be the selected value in the first stage, then we make the orthgonal array with the values of $\{1/(\overline{\rho_i}+0.01k),\ k=-1,0,1\}$, $i=1,2,3$ for the second stage of simulation. Simulation is stopped when a pre-specified number of customers has completed service at Node 3. These customers do not include those lost due to buffer overflow. We adopt the definition of $P_L(B_k)$ in [9]:

$$P_L(B_k) = Prob(customers\ loss\ with\ capacity = k) = \frac{\#\ lost}{\#\ lost + \#\ completed\ service}$$

The steps we used to this problem are as follows.

1. *Obtain an appropriate orthogonal array.*
   First, determine the control variables. In this problem, the control variables are the service rates, μ_i, $i=1,2,3$. Therefore, we have three control variables.
   Next, determine the number of levels for each control variable. Here, we decide $3^2=9$ as the number of levels for each control variable. We perform Step 2 and Step 3 up to as many times as the power of 3, which is two in this case. We call each execution of these two steps a "stage." The levels of μ_i, $i=1,2,3$ are as follows:

$$\mu_i \in \{(\frac{1}{0.50},\frac{1}{0.51},\frac{1}{0.52})(\frac{1}{0.53},\frac{1}{0.54},\frac{1}{0.55})(\frac{1}{0.56},\frac{1}{0.57},\frac{1}{0.58})\}$$

   Now, determine an orthogonal array according to the number of variables and the number of levels, which are decided above. We choose $L9(3^4)$ as the appropriate array for this problem. $L9(3^4)$ is shown in Table 2. In this case, the last column of the array is not used.

2. *For each row of the selected orthogonal array, perform simulation until a pre-specified number of customers have completed service at Node 3.*
   This step, along with Step 3 below is the first stage of the proposed method, and we take the median value of each subset in the first stage as follows:

$$\mu_i \in \{\frac{1}{0.51},\frac{1}{0.54},\frac{1}{0.57}\},\ i=1,2,3$$

   Each value represents the subset it belongs to. And, in the second stage, we choose the levels among the elements of the selected subset in the first stage.

3. *Select the levels by ANOM.*
   Calculate the average performance of each level of every control variable. Then, select the level for each control variable by comparing the orders of this average performance.

4. *Repeat Step 2 and Step 3.*
   We perform Step 2 and Step 3 once more for the second stage.

In general, the proposed method can be summarized as follows:

Step 1. Obtain an appropriate orthogonal array.

- **(a)** Determine the control variables.
- **(b)** Determine the number of levels for each control variable (Usually, we decide the numbers in the form of 3^n. Here, n will be the number of stages to be performed.)
- **(c)** Obtain an orthogonal array correspoding to the number of variables in (a) and appropriate levels in (b).

Step 2. Perform simulation.
For each row of the selected orthogonal array, perform simulation until a pre-specified simulation time.

Step 3. Select the best level for each control variable by ANOM.

Step 4. Repeat Step 2 and Step 3 $(n-1)$ times.

By applying the proposed method, we get only one candidate for the optimal solution. This is different from subset selection in ordinal optimization. In ordinal optimization, we can choose very good estimates by selecting not one sample, but a subset of the search space. This subset selection method increases the probability of choosing a good design [6] [8] [7] [11]. However, if we apply the proposed method several times rather than just once, we can also form a a set of candidates for the optimal solution. We will dicuss about this later.

4 Theoretical Result

In this section, we derive some theoretical result on the proposed method. First, we define the separability of the objective function. Secondly we derive the convergence bound of the proposed method.

A general problem of stochastic optimization can be defined as follows:

$$\min_{\theta\in\Theta}\Big[J(\theta)\equiv E\{L(\theta,\xi)\}\Big] \tag{1}$$

where Θ is the search space, θ is the design alternative, J is the objective function which is the expectation of L, the sample performance, and ξ the randomness in the system. When we are concerned with those real problems where $J(\theta)$ has large uncertainty, we must estimate $J(\theta)$ through simulation of sample performance, i.e.,

$$E\{L(\theta,\xi)\}\approx\tilde{J}(\theta)=L(x(t;\theta,\xi)) \tag{2}$$

where $x(t;\theta,\xi)$ is a system trajectory under the desgin parameter θ. Hereafter we use $E\{L(\theta)\}$ instead of $E\{L(\theta,\xi)\}$ for short.
By (1) and (2), $\tilde{J}(\theta)$ is expressed as follows:

$$\begin{aligned}\tilde{J}(\theta) &= J(\theta)+w(\theta)\\ &= E\{L(\theta)\}+w(\theta)\end{aligned} \tag{3}$$

where $w(\theta)\equiv w(t;\theta)$ is the estimation error or noise associated with the estimation of design θ. We assume that $w(\theta)$ is a zero-mean radom variable with its variance approaching zero as t goes to infinity.

Now we define the separability of the objective function as follows:

Definition 1: *An objective function $E\{L(\theta)\}$ is said to be separable if $E\{L(\theta)\}$ can be expressed as a sum of functions of each control variable, i.e.*

$$E\{L(\theta)\}=\sum_{i=1}^{n}E\{L_i(\theta_i)\},\ \theta=(\theta_1,...,\theta_n).$$

Even though the separability limits the type of systems, this class of systems includes a number of problems such as: 1) buffer allocation in parallel queueing systems where the blocking probability is the objective function; 2) queueing network with feedbacks where the mean sojourn time is the objective function; 3) cellular systems where the call loss probability of each cell depends only on the number of channels assigned to each cell [3].

With the definition of the separability, we can derive the following theorem.

Theorem 1: *When the objective function is separable for a discrete search space, the solution by ANOM and the solution by the full search method are same when t goes to infinity.*

Table 3: $L4(2^3)$ orthogonal array.

Run	θ_1	θ_2	θ_3	Result
1	1	1	1	J_1
2	1	2	2	J_2
3	2	1	2	J_3
4	2	2	1	J_4

Table 4: ANOM applied to $L4(2^3)$ orthogonal array.

	$\theta_i=1$	$\theta_i=2$
θ_1	$(J_1+J_2)/2$	$(J_3+J_4)/2$
θ_2	$(J_1+J_3)/2$	$(J_2+J_4)/2$
θ_3	$(J_1+J_4)/2$	$(J_2+J_3)/2$

Proof: We get the following equation for a separable objective function.

$$J(\theta)=E\{L(\theta)\}=\sum_{i=1}^{n}E\{L_i(\theta_i)\} \tag{4}$$

In order to minimize the objective function $J(\theta)$, we need to find the minimum of each $E\{L_i(\theta_i)\}$. Without loss of generality, we consider an example of ANOM applied to $L4(2^3)$ in Table 3. If we denote $\theta_i=j$ with θ_i^j, then the result of each run is

$$\begin{aligned}J_1&=E\{L_1(\theta_1^1)\}+E\{L_2(\theta_2^1)\}+E\{L_3(\theta_3^1)\}\\ J_2&=E\{L_1(\theta_1^1)\}+E\{L_2(\theta_2^2)\}+E\{L_3(\theta_3^2)\}\\ J_3&=E\{L_1(\theta_1^2)\}+E\{L_2(\theta_2^1)\}+E\{L_3(\theta_3^2)\}\\ J_4&=E\{L_1(\theta_1^2)\}+E\{L_2(\theta_2^2)\}+E\{L_3(\theta_3^1)\}\end{aligned}$$

Table 4 shows the procedure of ANOM. To compare the values of the objective function for θ_1, we substract the two values in the second row.

$$\begin{aligned}&\frac{(J_1+J_2)}{2}-\frac{(J_3+J_4)}{2}=\\ &E\{L_1(\theta_1^1)\}+\frac{1}{2}\sum_{i=2,3}\Big[E\{L_i(\theta_i^1)\}+E\{L_i(\theta_i^2)\}\Big]\\ &-E\{L_1(\theta_1^2)\}-\frac{1}{2}\sum_{i=2,3}\Big[E\{L_i(\theta_i^1)\}+E\{L_i(\theta_i^2)\}\Big]\\ &=E\{L_1(\theta_1^1)\}-E\{L_1(\theta_1^2)\}\end{aligned} \tag{5}$$

This shows that, by comparing the two values of each row in Table 4, we can decide exactly which value of θ_1 makes $E\{L_1(\theta_1)\}$ smaller. We get the same result for θ_2 and θ_3. The result is due to the orthogonality of the OA, and is universal for all OAs.

So we conclude that, by ANOM, we can determine the value of θ_i which makes $E\{L_i(\theta_i)\}$ minimum. Consequencely, when the objective function is separable, the solution by ANOM is same as the solution by the full search method. ■

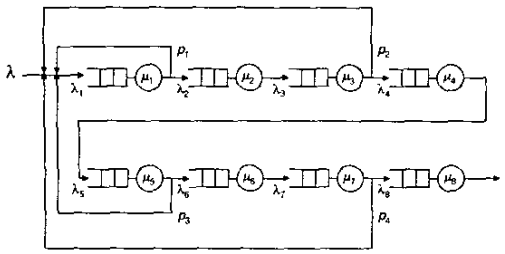

Figure 1: Queueing network with feedback.

5 Simulation Result: Queueing Network with Feedback

In this section, we present a simulation result of the proposed method. We consider a queueing network with feedback. The network is an extension of the system in [10]. The structure of this network is shown in Figure 1. Customers arrive at a rate $\lambda = 0.5$, and the values of the parameters are as follows:

$$\begin{aligned} p_1 &= \frac{1}{2},\ p_2 = \frac{1}{3},\ p_3 = \frac{1}{4},\ p_4 = \frac{1}{5} \\ \lambda_1 &= 2.5,\ \lambda_2 = 2.0,\ \lambda_3 = 2.0,\ \lambda_4 = 1.5 \\ \lambda_5 &= 1.5,\ \lambda_6 = 1.0,\ \lambda_7 = 1.0,\ \lambda_8 = 0.5 \\ \alpha_1 &= 7.0,\ \alpha_2 = 6.0,\ \alpha_3 = 5.0,\ \alpha_4 = 4.0 \end{aligned}$$

Here, p_i is the probability that customers loop back for another service. The objective is to minimize the mean sojourn time subject to the constraint. The objective function is given by (6).

$$\min[J] = \min[E\{S\}] = \min_{\lambda_i} \left[\frac{1}{\lambda} \sum_{i=1}^{8} \left[\frac{\lambda_i}{\mu_i - \lambda_i} \right] \right] \qquad (6)$$

$$s.t.\ \mu_{2i-1} + \mu_{2i} = \alpha_i,\ i = 1,2,3,4$$

As we can see in (6), the obejective function is separable. So, by Theorem 1, we can apply the proposed method. By using the constraint in (6), we can express the objective function as a function of four independent variables among μ_i, $i = 1, 2, \cdots, 8$. Here, we choose μ_1, μ_3, μ_5, and μ_7 as the control variables. To guarantee the stability of the system, the condition (7) should be satisfied.

$$\lambda_i < \mu_i\ ,\ i = 1,2,...,8 \qquad (7)$$

Therefore, the search space is bounded as follows:

$$\begin{aligned} 2.5 &< \mu_1 < 5.0\ ,\ 2.0 < \mu_3 < 4.5 \\ 1.5 &< \mu_5 < 4.0\ ,\ 1.0 < \mu_7 < 3.5 \end{aligned}$$

Table 5: Probabilities obtained by the proposed method applied to the queueing network with feedback.

customers completed service	100	200	300	400	500
Prob. to be within 10% offset	0.11	0.09	0.11	0.14	0.15
Prob. to be within 20% offset	0.28	0.32	0.32	0.36	0.39
customers completed service	600	700	800	900	1,000
Prob. to be within 10% offset	0.13	0.17	0.17	0.25	0.15
Prob. to be within 20% offset	0.42	0.44	0.52	0.49	0.50

To apply the proposed method, we must assign an appropriate number of levels to each variable. We assign $3^2 = 9$ levels to each variable here, and choose a combination of variables as a candidate of the optimal solution using an $L9(3^4)$ orthogonal array twice. Each control variable μ_i $(i = 1, 3, 5, 7)$ is discretized as follows:

$$\begin{aligned} \mu_1 &\in \{(2.55, 2.85, 3.15)(3.45, 3.75, 4.05)(4.35, 4.65, 4.95)\} \\ \mu_3 &\in \{(2.05, 2.35, 2.65)(2.95, 3.25, 3.55)(3.85, 4.15, 4.45)\} \\ \mu_5 &\in \{(1.55, 1.85, 2.15)(2.45, 2.75, 3.05)(3.35, 3.65, 3.95)\} \\ \mu_7 &\in \{(1.05, 1.35, 1.65)(1.95, 2.25, 2.55)(2.85, 3.15, 3.45)\} \end{aligned}$$

At the first stage, we take only the median values as the representation of each subset.

$$\begin{aligned} &\mu_1 \in \{2.85, 3.75, 4.65\},\ \mu_3 \in \{2.35, 3.25, 4.15\} \\ &\mu_5 \in \{1.85, 2.75, 3.65\},\ \mu_7 \in \{1.35, 2.25, 3.15\} \end{aligned}$$

After selecting the values among these using the proposed method, we apply the method once more with the values in the selected subsets. For example, if we select 3.75, 2.35, 1.85, and 3.15 for μ_1, μ_3, μ_5, and μ_7, respectively at the first stage, then we use the following subsets for the second stage.

$$\begin{aligned} &\mu_1 \in \{3.45, 3.75, 4.05\},\ \mu_3 \in \{2.05, 2.35, 2.65\} \\ &\mu_5 \in \{1.55, 1.85, 2.15\},\ \mu_7 \in \{2.85, 3.15, 3.45\} \end{aligned}$$

The size of the search space, is $9 \cdot 9 \cdot 9 \cdot 9 = 6561$ in this problem. Among the elements of the search space, the number of those whose offset from the optimal is within 10% is 196, and within 20% is 550. So, if we choose one design by blind picking, the probability that the chosen design is within 10% offset from the optimal is $196/6561 \cong 0.030$, and the probability within 20% offset from the optimal is $550/6561 \cong 0.084$. By applying the proposed method, we can reduce the size of search space to $9 \cdot 2 = 18$, which is $18/6561 \cong 1/365$ of the orginal size.

The simulation results are given in Table 5. Each of the results is an average for 100 runs. We can verify from Table 5 that the probabilies of finding a "good-enough" solution by the proposed method are much higher than the probablities by blind picking, even with a small number of customers completed service. And if we apply the proposed method several times, we can increase the probability considerably. For example, consider the result of which the number of customers com-

pleted service is 100. Table 5 shows that the probability to be within 20% offset from the optimal is 0.28. If we apply the proposed method ten times, we would get a set of ten elements, and the probability that at least one of these is within 20% offset from the optimal is 0.96. We can still reduce the size of search space to $(1/365) \cdot 10 \cong 0.03$ of the original size. This means that we can get at least one "goood-enough" design with high probability by applying the proposed method several times, and we can still reduce the simulation considerably. the search space increases combinatorially as the number of control variables increases. However, with the proposed method, combinatorial increase of the search space can be changed into arithmetic increase. Cosequently, the proposed method becomes more effecitive as the number of control variables increases.

6 Conclusion

We have discussed a new approach, ordinal optimization with orthogonal arrays, for the reduction of search space in stochastic optimization problems. One of the main problems in stochastic optimization is the problem of large search space. To resolve this problem, we suggest a combined approach, ordinal optimization with orthogonal arrays. The orthogonal array method adopts the notion of sub-optimal solutions, which coincides with the notion of "goal softening" in ordinal optimization. Furthermore, ANOM selects the level by comparing the orders of performance of levels, and this can be regarded as comparison of order in ordinal optimization.

Moreover, this method, ordinal optimization with orthogonal arrays becomes more effective especially when the search space is large because combinatorial increase of the search space is changed into arithmetic increase. Consequently, the proposed approach is a very effective method for resolving the problem of large search space in stochastic optimization and improving the efficiency of the conventional ordinal optimization approach. We expect the proposed method to be a complementary tool for the ordinal optimization approach.

References

[1] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*, Irwin, 1993.

[2] C. G. Cassandras, and C. G. Panayiotou, "Ordinal optimization for deterministic and stochastic discrete resource allocation," *Proc. of the 36th Conf. on Decision and Control*, pp.662–667, 1997.

[3] C. G. Cassandras, L. Dai, and C. G. Panayiotou, "Ordinal optimization for a class of deterministic and stochastic discrete resource allocation problems," *IEEE Trans. on Automatic Control*, vol.43, no.7, pp.881-890, 1998.

[4] R. A. Fisher, *The Design of Experiments*, Oliver and Boyd, 1935.

[5] W. Y. Fowlkes and C. M. Creveling, *Engineering Methods for Robust Product Design* , Addison-Wesley, 1995.

[6] Y. C. Ho, "Heuristics, rules of thumb, and the 80/20 proposition," *IEEE Trans. on Automatic Control*, vol.39, no.5, pp.1025-1027, 1994.

[7] Y. C. Ho, "On the numerical solutions of stochastic optimization problems," *IEEE Trans. on Automatic Control*, vol.42, no.5, pp.727–729, 1997.

[8] Y. C. Ho and M. Deng, "The problem of large search space in stochastic optimization," *Proc. of the 33rd Conf. on Decision and Control*, pp.1470–1475, 1994.

[9] Y. C. Ho and M. E. Larson, "Ordinal optimization approach to rare event probability problems," *Discrete Event Dynamic Systems:Thory and Appl.*, vol.5, pp.281-301, 1995.

[10] Y. C. Ho, R. S. Sreenivas, and P. Vakili "Ordinal optimization of DEDS," *Discrete Event Dynamic Systems:Theory and Appl.*, vol.2, pp.61–88, 1992.

[11] T. W. E. Lau and Y. C. Ho, "Universal alignment probabilities and subset selection for ordinal optimization," *J. Optimization Theory and Appl.*, vol.93, no.3, pp.455-489, 1997.

[12] L. H. Lee, T. W. E. Lau, and Y. C. Ho, "Explanation of goal softening in ordinal optimization," *IEEE Trans. on Automatic Control*, vol.44, no.1, 1999.

[13] P. J. Smith, M. Shafi, and H. Gao, "Quick simulation: a review of importance sampling techniques in communications systems," *IEEE Journal on Selected Areas in Comm.*, vol.15, pp.597-613, 1997.

[14] G. Taguchi, *Taguchi on Robust Technology Development*, ASME, New York, NY, 1993.

[15] J. K. Townsend, et al. "Simulation of rare events in communications networks," *IEEE Comm. Magazine*, August, 1998.