A METHOD FOR SELECTING THE BEST PERFORMANCE SYSTEMS

Mohammad H. Almomani\textsuperscript{1}§, Mahmoud H. Alrefaei\textsuperscript{2}, Shahd Al Mansour\textsuperscript{3}

\textsuperscript{1}Department of Mathematics
The Hashemite University
Zarqa, JORDAN

\textsuperscript{2,3}Department of Mathematics and Statistics
Jordan University of Science and Technology
JORDAN

\textbf{Abstract:} The selection of the best stochastic systems from a set of a finite but very large alternative systems is considered in this paper. Suppose we have limited computational budget to be distributed among the alternatives in order to correctly select the best alternative. Instead of distributing these computational efforts equally likely among the alternatives, the optimal computing budget allocation (OCBA) procedure puts more effort on the promising alternatives. In this article, we propose an algorithm that combines three procedures including the Ordinal Optimization (OO), the OCBA, and the ranking and selection (R&S) to select a good enough solution. The algorithm was tested on a generic example under various parameter settings and the numerical results indicates that the algorithm behaves well under various parameter setting.

\textbf{AMS Subject Classification:} 62F07, 90C06, 49K30

\textbf{Key Words:} ordinal optimization, optimal computing budget allocation, ranking and selection

\section{1. Introduction}

Consider the problem of locating the best system among a very large num-
ber of alternative simulated systems. This problem appears in many aspects of real life such as scheduling systems, manufacturing systems, telecommunication systems, etc.. When the number of alternatives is small, then Ranking and Selection (R&S) procedures can be applied to select the best with a predetermined significance level, (see [1], [2], [3]). We consider the problem of optimizing large scale optimization problems when there is no exact formula for the objective function values but have to be simulated to get their estimates. We propose a technique for designing and analyzing experiments involving with several qualitative factors, when the objective is to select the best system, where the best system is defined as the one that has the (maximum or minimum) performance measure.

We are interested in this optimization problem

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

(1)

where $\Theta$ is huge set, $J$ is the objective function which is an expected value of a probabilistic system that can be written as $J(\theta) = E[Y(\theta, \xi)]$, where $Y(\theta, \xi)$ represents the random behavior of the system depends on the vector $\theta$ and the random effect $\xi$.

For large scale problems as in (1), the R&S procedures may not be applicable because they need a huge computational time. Some other procedures have used Simulated Annealing and Particle Swarm, these include [4]. In this case, one would change the objective to find a good enough solution rather than estimating accurately the performance value of each system. This can be achieved using the ordinal optimization procedures. In this paper, we combine three procedures including the OO, OCBA, and R&S procedures to select one of the best systems. There literature is rich in such combined procedures for selection purposes in the large scale problems, (see [5], [6], [7], [8], [9], [10], [11]). The main objective of this article is to select the best simulated system from the top $k\%$ systems, when the search space is very large.

R&S procedures play an important role in optimization via simulation, they are used to rank a finite number of simulated systems and then select the best one with a predetermined level of significance. R&S procedures assume that independent and identically normally distributed samples are available. The Indifference-Zone (IZ) procedure is used to select a system that is indifferent from the actual best system by a predetermined indifference. A good comprehensive reviews for all types of R&S procedures and other comparison procedures can be found in [12], [3] and [13].

In cases of large scale optimization problems, instead of finding the exact optimal solution, one can think of finding a good enough solution. Either a
A METHOD FOR SELECTING...

set containing some of the best alternatives or selecting an alternative that is very close to the optimal solution with high significant level. This technique is proposed by Ho et al. [14] and is called Ordinal Optimization (OO). In OO, one selects a set of $m$ alternatives from the solution set; using order statistics, one can show that the selected set overlaps with the actual best $k$ alternatives with high probability. Instead of spending much time on simulation to get the accurate estimation of each system, the aim of this technique is to relax the problem to find a good, better, or best system with little effort on simulation.

The idea of OO is to select a subset $G$ randomly from $\Theta$, then use one known optimization method to select the best system in $G$. For more details about OO and its applications see [15], [16] and [17].

The remaining of this paper is organized as follows: A background about OCBAm procedure is given in Section 2. In Section 3 we present the algorithm of a combined approach, followed by an empirical testing and discussion in Section 4. Finally, Section 5 includes concluding remarks.

2. Optimal Computing Budget Allocation

Assume that there are available simulation budget to be allocated on different alternatives in order to select the best system or a set of best systems. [19] has proposed a procedure that distribute the available budget among the different alternatives in a smart way that maximizes the probability of selecting the best system. So the alternative systems that have more impact on the final solution are given more computational time than the ones that have no impact.

Consider the problem of selecting a set $S_m$ that contains the best $m$ systems from $k$ systems. We will discuss how to allocate the available budget among the alternative systems in order to improve the probability of correct selection (CS), where the CS is to select a set $S_m$ that contains the actual best $m$ systems. [20] formulate the problem as:

$$\max_{N_1,...,N_k} P(CS_m)$$

s.t. $\sum_{i=1}^{k} N_i = T$ \hspace{1cm} (2)

where $T$ is the total simulation samples over the $k$ different alternative systems, $S_m$ is the set that contains the best $m$ systems and $N_i$ is the number of simulation samples that are allocated to system $i$. Assume that $Y_{ij}$ is the $j^{th}$ simulation sample of system $i$ for estimating the mean $Y_i$, where $j = 1, \ldots, N_i$.
and $\bar{Y}_i = 1/N_i \sum_{j=1}^{N_i} Y_{ij}$ is the estimated mean. By the Central Limit Theorem, $\bar{Y}_i$ is normally distributed with mean $Y_i$ and variance $\sigma_i^2/N_i$.

Suppose that a smallest mean is better, then $S_m$ is the $m$ systems with the smallest sample means. Define $Y_{ir}$ be the $r$-th smallest of $\{Y_1, Y_2, \ldots, Y_n\}$, i.e. $Y_{i1} \leq Y_{i2} \leq \ldots \leq Y_{in}$. Then, the selected set $S_m$ will be $S_m = \{i_1, i_2, \ldots, i_m\}$. The correct selection is defined by that $S_m$ contains all systems with the actual smallest $m$ means, i.e. $CS_m = \{\max_{i \in S_m} Y_i \leq \min_{i \notin S_m} Y_i\}$, where $Y_i$ is the (unknown) mean of system $i$.

The optimal computing budget allocation now is to maximize the probability of correct selection $P(CS_m)$ given a finite number of total computing budget $T$ as in equation 2, this procedure known as $OCBA_m$. [20] provide the following theorem for distributing the available computational time among all alternative systems.

**Theorem 1.** “Given a total number of simulation samples $T$ to be allocated to $k$ computing systems in order to select a set of the best $m$ systems whose performance is depicted by random variables with means $Y_1, Y_2, \ldots, Y_k$ and finite variances $\sigma_1^2, \sigma_2^2, \ldots, \sigma_k^2$, respectively. As $T \to \infty$, the approximate probability of correct selection can be asymptotically maximized when

$$\frac{N_i}{\sigma_i^2/\delta_i^2} = \frac{N_j}{\sigma_j^2/\delta_j^2}$$

where $N_i$ is the number of samples allocated to system $i$, $\delta_i = Y_i - c$, and $c$ is a constant satisfies $Y_{[m]} \leq c \leq Y_{[m+1]}$, where $Y_{[m]} = \max_{j \in S_m} Y_j$ and $Y_{[m+1]} = \min_{j \notin S_m} Y_j$.”

### 3. A Combined Procedure for Selecting a Good System

The proposed algorithm consists of combining three procedures including $OO$, $OCBA_m$ and $R&S$ procedures. Firstly, the $OO$ procedure is used in order to explore the search space, so a relatively small set is selected randomly, then the $OCBA_m$ procedure is used to screen out the selected subset and select a smaller subset that contains the best systems and the complement of this set is replaced by new candidates that are selected randomly from the search space. This procedure is repeated until the budget that is allocated to select a small subset is consumed. After that, when a small set is selected, the $IZ$ procedure is used to select a system among the smaller subset as the overall best system.

Consider a set $G$ of $g$ alternative systems and it is sought to select the best $k$ alternatives from $G$. Recall that from Theorem 1; $\frac{N_i}{\sigma_i^2/\delta_i^2} = \frac{N_j}{\sigma_j^2/\delta_j^2}$, where $i \neq j$,
fix an index $p$, then we have \( \frac{N_p}{\sigma_p^2/\sigma_p^2} = \frac{N_j}{\sigma_j^2/\sigma_j^2} \) so that \( N_j = \frac{\sigma_j^2}{\sigma_j^2} \frac{\Delta \sigma_p^2}{\sigma_p^2} N_p = \frac{\alpha^2_j}{\alpha^2_p} N_p \) where \( \alpha_j = \delta_j / \sigma_j \). Since \( \sum_{j \in \Theta} N_j = T \), we get \( \sum_{j \in G} \frac{\alpha^2_j}{\alpha^2_p} N_p = T \) for all \( j \in G \), let \( D_j = \alpha^2_p / \alpha^2_j \) and \( D_p = \sum_{j \in G} D_j \) then

\[
N_j = TD_j/D_p ; \quad j \in G \ , \ j \neq p \\
N_p = T/D_p
\]

(4)

Algorithm 3.1.

**Step 0:** Determine the initial samples \( t_0 \geq 5 \) and the number of samples to be distributed over the set of systems in each iteration, \( \Delta \). Determine the indifference zone \( d^* \), \( t = t_0 (1-\alpha/2)^{1/2} t_{0-1} \) from the \( t \)-distribution. Let \( l = 0 \) where \( l \) represents iteration index. Let \( R \) be the size of the inner loop (the maximum number of replications of the \( OCBA_m \) inside each iteration), let \( r = 0 \), \( T T = 0 \).

**Step 1:** Select a set \( G \) of size \( g \) from the search space \( \Theta \), let \( N_i^0 = t_0 \) for all \( i \in G \), and \( N_i^0 = 0 \) if \( i \notin G \), \( i \in \Theta \). Take random samples of \( t_0 \) observations \( Y_{ij} \) (1 \( \leq j \leq t_0 \)) for each system \( i \in G \). Compute the sample mean \( \bar{Y}_i \) and the sample variance \( s^2_i \) as; \( \bar{Y}_i = \frac{1}{N_i^0} \sum_{j=1}^{N_i^0} Y_{ij} \) and \( s_i = \frac{1}{N_i^0-1} \sum_{j=1}^{N_i^0} (Y_{ij} - \bar{Y}_i)^2 \).

**Step 2:** Increase the computing budget by \( \Delta \) and compute the new budget allocation \( N_i^{r+1} \), \( i \in G \) using equation (4).

**Step 3:** Perform additional \( \max\{0, N_i^{r+1} - N_i^r \} \) simulation samples for each system \( i \in G \). Compute the new sample means and variances, let \( r \leftarrow r+1 \), let \( TT = TT + \sum_{i \in G} \{N_i^{r+1} - N_i^r \} \), if \( TT \geq T \) go to Step 5 if \( r < R \) go to Step 2.

**Step 4:** Let \( S_m \) be the subset that contains the \( m \) best mean systems. Let \( N_i^0 = N_i^r \) for all \( i \in G \) Randomly select a subset \( S_c \) of \( g - m \) alternatives from \( \Theta - G \) and replace the worst \( g - m \) systems of the set \( G \) with \( S_c \). For \( i \in S_c \), if \( N_i^0 = 0 \), let \( N_i^0 = t_0 \), let \( TT = TT + t_0 \) take random samples of \( N_i^0 \) observations \( Y_{ij} \) (1 \( \leq j \leq N_i^0 \)) and calculate the sample mean \( \bar{Y}_i \) and sample variance \( s^2_i \) for all \( i \in G = S_m \cup S_c \). Let \( l \leftarrow l + 1 \), \( r = 0 \), go to Step 2.
Step 5: Select the set $S_m$ that contains the best systems in $G$. For $i \neq j \in S_m$.

Let $W_{ij} = t \left( \frac{s_i^2}{N_i} + \frac{s_j^2}{N_j} \right)^{1/2}$. Let $I = \{ j : 1 \leq j \leq m \text{ and } \bar{Y}_j \leq \bar{Y}_i - [W_{ij} - d^*]^{-}, \forall j \neq i \}$.

Step 6: If $I$ contains only one solution, then select it as the best system. Otherwise, for all $i \in I$, compute $N_i^{(2)} = \max\{N_i^0, \lfloor (h s_i^2)^2 \rfloor\}$, $h = h(1 - \alpha/2, t_0, |I|)$ is the Rinott [1] constant.

Step 7: For each system $i \in I$, take additional $N_i^{(2)} - N_i^0$ random samples; $Y_{ij}$. Let $\bar{Y}_i^{(2)} = \frac{\sum_{j=1}^{N_i^{(2)}} Y_{ij}}{N_i^{(2)}}$.

Step 8: Select the system $j \in I$ with the smallest $\bar{Y}_i^{(2)}$ as the required best system.

4. Empirical Testing

We test the proposed algorithm on a generic example of monotone increasing mean, where the probability of correct selection estimate by counting the number of times of success event, in which the best systems belong to the actual $k\%$ best subset out of 10 independent replications. Suppose there are 1000 systems, with the systems are normally distributed $N(10 + \frac{(i-1)}{10}, 1)$ for $i = 1, 2, \ldots, 1000$. Assume that the variances are constant, where for each system the variance is 1. The objective is to select a system with the smallest mean. Obviously system 1 has the smallest mean. Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_{1000}\}$, and $f(\theta_i) = 10 + (i - 1)/10$, where the optimization problem involves $\min_{i=1,\ldots,1000} f(\theta_i)$.

The aim is to select one system from the set $S_m$ that contains the best $k\% = 10\%$ system i.e. $S_m = \{1, 2, \ldots, 100\}$. We will consider it as a correct selection if the selected system belongs to $S_m$. Initially we implement $t_0 = 5$ and an increment of $\Delta = 10$ per stage until the total budget exceeds $N = 2500$ samples. Assume that $g = 50$, and $d^* = 0.05$. Table 1 show the results of this experiment.

Obviously, in this experiment, the proposed selection approach selected the correct system in all the first 10 replications. Moreover, from Table 1, in the first replication the selection procedure selected the set $S_m = \{4, 6, 5, 9, 0, 12, 11, 10, 21, 29\}$ as the best 10 systems from the 1000 systems and we can see here all these systems are among the the best 10% systems, so the proposed procedure...
Table 1: The performance of the selection procedure for $n = 1000$, $g = 50$, $k\% = 10\%$, $t_0 = 5$, $\Delta = 10$, $N = 2500$.

<table>
<thead>
<tr>
<th>Replication</th>
<th>Best System</th>
<th>$\sum_{i=1}^{m} N_i$</th>
<th>Best m Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2605</td>
<td>${4, 6, 5, 9, 0, 12, 11, 10, 21, 29}$</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2500</td>
<td>${2, 18, 20, 21, 19, 32, 50, 46, 44, 54}$</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>3004</td>
<td>${16, 4, 6, 9, 19, 12, 7, 13, 36, 25}$</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>2702</td>
<td>${10, 3, 6, 7, 12, 8, 27, 15, 13, 17}$</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2923</td>
<td>${0, 9, 4, 12, 11, 14, 17, 10, 26, 15}$</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>2815</td>
<td>${0, 6, 10, 5, 3, 2, 17, 9, 12, 13}$</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2794</td>
<td>${7, 9, 2, 6, 0, 10, 4, 13, 3, 15}$</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2647</td>
<td>${2, 0, 11, 7, 6, 1, 9, 15, 24, 21}$</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2962</td>
<td>${1, 11, 5, 6, 23, 14, 15, 18, 19, 17}$</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2949</td>
<td>${1, 2, 4, 8, 3, 6, 11, 20, 14, 12}$</td>
</tr>
</tbody>
</table>

made a correct selection. It is clear that the probability of correct selection is high in this experiment, note that the $P(CS) = 1$ with the average number of the sample size 2790. Also we notice that the number of sample sizes is relatively small.

Now, we test the effect of the simulation parameters to the performance of the proposed algorithm. These parameters include; the initial sample size $t_0$, the increment in each stage $\Delta$, and the total budget $T$.

**The Effect of the Initial Sample Size ($t_0$)**

The sample size in the first stage is called the initial sample size ($t_0$). It plays an important role in the performance of many selection approaches. If $t_0$ is too small, then we may get poor initial estimates of the variances of the alternatives. If it is too large, then we may spend more computational time on irrelevant alternatives, so there should be a balanced value of $t_0$ in order to get better solution. [18] and [19] suggested that as a good choice for the initial sample size, the value of $t_0$ should be between 10 and 20. We implement the proposed algorithm on three different values, including $t_0 = 5$, 10 and 20. The results are depicted in Figure 1. It is clear that $t_0 = 5$ gives better performance than the other two values and $t_0 = 10$ gives better performance than $t_0 = 20$. This is because the algorithm focusses on the alternative systems that are likely to be selected rather than spending times on the ones that are not of interest.
The increment in simulation samples (∆)

The increment in simulation samples (∆) is defined as a positive integer that represents the additional number of simulation samples in Step 3 in the proposed algorithm. In order to avoid repetition increment in the optimal computing budget allocation algorithm, ∆ cannot be too small, so that the algorithm spends much time on calculations. On the other hand, if ∆ is too large, it will result in wasting computational time on unnecessary irrelevant systems. [18] and [19] have suggested that ∆ should be between 5% and 10% of the simulated system as a good choice for the increment in simulation samples.

We implement the proposed algorithm on the above example using three different selection of the value of ∆ these include ∆ = 10, 20 and 50. We have used $t_0 = 5$ since it gives better performance. The results are depicted in Figure 2. It is clear that the algorithm gives better performance when the value of ∆ = 10 and ∆ = 20. However, when ∆ = 50 then the algorithm does not behave well. This is because the $OCBA_m$ distribute the available budget on alternatives that are not likely to be the best.

The Performance of the Algorithm as time evolves

We implement the proposed algorithm when it is sought to select either one of the best 2%, 5% or 10% and see the performance of the algorithm against the
total available budget. Clearly, the increase in the total budget $T$, will improve the efficiency of the selection approach by increasing the probability of correct selection. In particular, when $T \to \infty$ then $P(CS)$ is approaching 1. Whereas, by increasing $T$ will increase the simulation time and the number of simulation samples.

We apply the proposed selection algorithm with the setting, $t_0 = 5$ and $\Delta = 10$. These experiments are repeated 10 replications, each replication consists of 20 iterations; the probability of correct selection is shown in Figure 3. We can conclude that the probability of correct selection converge to 1 quickly.

5. Conclusion

In this paper, we have presented a selection procedure for selecting a good enough system from a large set of alternative systems. The approach consists of three stages. Firstly, we used the $OO$ procedure to select a subset $G$ randomly from the search space that contains the best simulated system with high probability, after that we used the $OCBA_m$ procedure to select the best $m$ subset of $G$ which is chosen by $OO$ procedure, then we used the $IZ$ procedure to select the best system of $m$. The proposed selection approach is applied on monotone increasing mean example with different parameter setting. We can say that the algorithm of the proposed approach is working practically well.
200

M.H. Almomani, M.H. Alrefaei, S. Al Mansour

Figure 3: The performance of the algorithm for selecting the best 2%, 5% and 10% systems and it achieves correct selection with maximum probability of correct selection. Moreover, we have discussed the impact of various simulation parameters on the performance of the algorithm. From our case study, we can say that to achieve the best performance in the context of $P(CS_m)$ for the proposed algorithm, we need a small value of $t_0$, $t_0 = 5$ and at the same time, small value of $\Delta$, $\Delta = 10$. Meanwhile, when the total budget $T$ increases the $P(CS_m)$ increase as will. The numerical results show that the algorithm selects one of the best $k\%$ systems very fast and the smaller the values of the initial sample size, the better solution we get. The number of samples to be performed and distributed over the alternatives $\Delta$ is better to be moderate.

References


