Int. J. Open Problems Compt. Math., Vol. 7, No. 4, December, 2014
ISSN 1998-6262; Copyright ©ICSRS Publication, 2014
www.i-csrs.org

Efficient Approach for Selecting the Best Subset of Buffer Profile

Mohammad H. Almomani

Faculty of Science, Jerash University, 26150, JORDAN
e-mail:mh_momani@yahoo.com

Received 1 July 2014; Accepted 20 October 2014

Abstract

One of the main problems with designing a production line is to find the optimal number of buffers between workstations in order to maximizes the throughput. This problem known as buffer allocation problem. Previous work in this problem focus on selecting a single buffer profile that has the maximum throughput. The objective in this paper would be to selecting from a large number of alternatives, the best subset of buffer profiles where its throughput are at its maximum. The ordinal optimization with optimal computing budget allocation approaches will be used to isolating the best subset of buffer profile, where its throughput is maximum, from the set of all alternatives. Numerical results show that the proposed algorithm finds the best subset of the puffer allocation with high probability and small replications numbers of samples.

Keywords: *Buffer Allocation Problem, Optimal Computing Budget Allocation, Ordinal Optimization, Simulation Optimization.*

2010 Mathematics Subject Classification: 62F07, 90C06, 49K30.

1 Introduction

In this paper, we are dealing with one problem in designing a production line, which is the problem of buffer allocation. In buffer allocation problem (BAP) the goal is to allocate Q slots amongst the q intermediate buffers between $q+1$

workstations in a production line, in order to meet some specified aim. In the most BAP the performance measures for a production line are the throughput and the average work in process. The concern is, to allocate Q slots, over q buffers in order to maximize the throughput or minimize the average work in process.

The BAP is a difficult optimization problem since it is very difficult to calculate the exact value of the objective function for a given allocation. So, the objective function for this optimization problem needs to be estimated, see Chaharsooghi and Nahavandi [1]. Together, the BAP is involved with a large number of feasible allocations with respect to the total number of slots and the number of workstations in production line. Assume that there are Q slots available that needs to be allocated over the q buffers, then the number of different alternative designs is $\binom{Q+q-1}{Q}$. Each alternative is called "buffer profile", where it represents a unique combination of storage allocation with a potential to result in a different output level of the line. For instance, if the number of slots $Q = 18$ and the number of buffers $q = 5$ then we have 7315 different buffer profiles. The objective would be to select from this large set of alternatives, the best buffer profile that has the maximum throughput.

Previous work in the BAP mostly focus on selecting the single best buffer profile, see Almomani et al. [2]. In many cases, it is important to provide a set of good buffer profiles rather than selecting the best one. The purpose of this paper is using a new selection approach, to find the optimal allocation of buffers (set of buffer profile) that maximizes the throughput in a short, unbalanced and reliable production lines. The proposed approach is a combination of the ordinal optimization (OO), see Ho et al. [3], and the optimal computing budget allocation method for selecting the top m designs (OCBA-m), see Chen et al. [4]. The objective of OO in the proposed approach is to isolating a subset of good enough designs with high probability and reducing the size of the search space so that it is appropriate to apply the OCBA-m approach. We use the OCBA-m to formulate the problem as that of maximizing the probability of correctly selecting all of the top m buffer profile subject to a constraint on the total number of simulation replications available.

2 Buffer Allocation Problem

Production designs are often organized with a queuing workstations that are connected in series and are separated by buffers. Figure 1 represents a production line of 4 workstations and 3 buffers. The job moves in the direction of the arrows, from source inventory to the first workstations for served, then to the first buffer where it waits until the second workstation become empty then it moves to the second workstations, etc... until it finishes all the stations in the queue and leave the line. The challenge is, how to allocate the slots in away that will achieve the desired performance. Note that, the buffers cannot be too large because; an increase in the buffer size usually will increase the total of work in progress, time to the customer, inventory and capital. On the other hand, the buffers cannot be too small because the workstations will be untapped to meet demand.

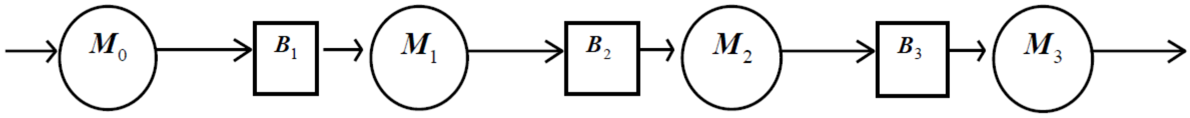

Figure 1: A production line of 4 workstations and 3 buffers

There are two types of BAP; short and longer lines as presented in Papadopoulos et al. [5]. The short line is a production line with up to 6 workstations with maximum of up to 20 slots, whereas the larger lines is otherwise. Furthermore, the BAP can be defined as balanced and unbalanced line, where the balanced line is a line with equal mean service time at each of the $q + 1$ workstations. Production line also, can be defined as a reliable or unreliable line, where in reliable line each workstation of the line cannot be failed. An illustration of the definitions can be see in Almomani et al. [2].

Other literature reviewed that relates to the BAP, include Spinellis and Papadopoulos [6] described a simulated annealing procedure for solving the BAP in reliable production lines, with the objective of maximizing the throughput. Chaharsooghi and Nahavandi [1] presented a heuristic algorithm to find the optimal allocation of buffers that maximizes throughput. Alon et al. [7] presented a stochastic algorithm for solving the BAP, based on the cross-entropy method. Yuzukirmizia and Smithb [8] proposed a new procedure to get a suboptimal buffer profile for closed queuing networks with multiple servers and finite buffers. For more details about BAP, see Alrefaei and Andradóttir [9], Daskalaki and Smith [10], Foley and Park [11], Kim et al. [12], Huang et al. [13], Diamantidis and Papadopoloulos [14].

3 Ordinal Optimization

The objective of the OO is separating a subset of good designs with high probability and reducing the required simulation time for discrete event simulation. The target of this approach is to find a good enough solution, rather than estimating the performance value of these designs accurately. Therefore, OO approach is used to select a subset that overlap with the set of the actual best $k\%$ designs with high probability.

Suppose that the correct selection (CS) is to select a subset G of g designs from the search space set Θ that contains at least one of the top $k\%$ best designs. Since we assume that Θ is large, then the probability of correct selection is given by $P(CS) \approx 1-(1-\frac{k}{100})^g$. Furthermore, suppose that the correct selection is to select a subset G of g designs that contains at least r of the best s designs. Let S be the subset that contains the actual best s designs, then the probability of correct selection can be obtained using the hypergeometric distribution as $P(CS) = P(|G \cap S| \geq r) = \sum_{i=r}^{g} \frac{\binom{s}{i}\binom{n-s}{g-i}}{\binom{n}{g}}$. Since $|\Theta| = n$, and n is large then $P(CS)$ can be approximated by the binomial random variable. Therefore, $P(CS) \approx \sum_{i=r}^{g} \binom{g}{i} \left(\frac{k}{100}\right)^i \left(1-\frac{k}{100}\right)^{g-i}$, where we assume that $s/n \times 100\% = k\%$. Another comprehensive review of OO approach can be found in Ho et al. [3], Deng and Ho [15], Lee et al. [16], Zhao et al. [17] and Ho et al. [18].

4 Computing Budget Allocation for Selecting an Optimal Subset

To improve the efficiency of OO choose the simulation lengths for different designs to minimize the total computation time. The target is to allocate the total simulation samples from all designs in a way that maximizes the probability of selecting the best design within a given computing budget, instead of allocating the computing budget among equally simulating designs. To achieve this target Chen et al. [19] proposed the optimal computing budget allocation (OCBA) approach that gives a large number of simulation samples to the designs that have a great effect on identifying the best design, whereas it gives a limited simulation sample for those designs that have little effect on identifying the best one.

Most research on the same framework has focused on selecting the single best design, see Almomani and Abdul Rahman [20], Almomani and Abdul Rahman [21], Almomani et al. [22], Almomani et al. [23], Almomani and Alrefaei [24], Alrefaei and Almomani [25], and there has been no research involving subset selection. Chen et al. [4], Chen et al. [26] fill this gap by providing an efficient allocation approach for selecting the top m designs, known as (OCBA-m) approach. They formulate the problem as that of maximizing the probability of correctly selecting all of the top m designs $P(CS_m)$ subject to a constraint on the total number of samples available, i.e.

$$\begin{aligned} &\max_{T_1,\ldots,T_n} \; P(CS_m) \\ &s.t. \quad \sum_{i=1}^{n} T_i = T \end{aligned} \tag{2.1}$$

where, T the total number of simulation samples, n the total number of designs, m the number of top designs to be selected in the optimal subset, S_m the set of m (distinct) indices indicating designs in selected subset, T_i the number of simulation samples allocated to design i and $\sum_{i=1}^{n} T_i$ denotes the total computational samples. The goal is to find a simulation budget allocation that maximizes the probability of selecting the optimal subset, defined as the set of m $(< k)$ best designs, for m a fixed number.

Suppose the m best designs are the designs with the m largest means, which is unknown and to be inferred from simulation. Let n the number of designs, and let Y_{ij} represent the j^{th} output from the design i. We assume that Y_{ij} are independent and identically distributed normal with unknown means $Y_i = E(Y_{ij})$ and variances $\sigma_i^2 = Var(Y_{ij})$. If σ_i^2 are unknown, we estimate it by the sample variances s_i^2 for Y_{ij}. Since we assume a largest mean is better, therefore we will take S_m to be the m designs with the largest sample means. Let $\bar{Y}_{i_r}$ be the r-th largest of $\{\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_n\}$, i.e. $\bar{Y}_{i_1} \geq \bar{Y}_{i_2} \geq \ldots \geq \bar{Y}_{i_n}$, where $\bar{Y}_i = \frac{1}{T_i}\sum_{i=1}^{T_i} Y_{ij}$ is the sample mean for design i. Then, the selected subset is given by $S_m = \{i_1, i_2, \ldots, i_m\}$. The correct selection is defined by S_m containing all of the m largest mean designs.

To solve (2.1) problem, Chen and Lee [27] proposed the following theorem that was useful in choosing the simulation samples for all designs in a way that maximizes the approximate $P(CS_m)$.

Theorem 4.1 *Given a total number of simulation replications T to be allocated to n competing designs whose performance is depicted by random variables with means $Y_1, Y_2, \ldots, Y_n$, and finite variances $\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2$ respectively, as $T \longrightarrow \infty$, the approximate probability of correct selection for m best ($APCS_m$) can be asymptotically maximized when $\frac{T_i}{T_j} = \left(\frac{\sigma_i/\delta_i}{\sigma_j/\delta_j}\right)^2$; for any $i, j \in \{1, 2, \ldots, n\}$ and $i \neq j$, where $\delta_i = \bar{Y}_i - c$, for c a constant.*

Note that, the parameter c impacts the quality of the $APCS_m$ to $P(CS_m)$. Since $APCS_m$ is a lower bound of $P(CS_m)$, choosing c to make $APCS_m$ as large as possible is likely to provide a better approximation of $APCS_m$ to $P(CS_m)$, see Chen et al. [4] and Chen and Lee [27]. However, in this paper we choose c to be $c = \frac{\left(\hat{\sigma}_{i_{m+1}}\bar{Y}_{i_m} + \hat{\sigma}_{i_m}\bar{Y}_{i_{m+1}}\right)}{\left(\hat{\sigma}_{i_m} + \hat{\sigma}_{i_{m+1}}\right)}$ where $\hat{\sigma}_i = \sigma_i/\sqrt{T_i}$.

5 Assumptions of the Model

This paper assume that the production line is short, reliable and is unbalanced with unlimited supply of jobs in the first workstation -the workstation will never been starved- and unlimited space after the final workstation -the workstation will never be blocked-. Jobs received service at each workstation with the service times being independently random variables following the exponential distribution with rate μ_i, for $i = 0, 1, \ldots, q$. With the model given above, the objective is to maximize the throughput, subject to a given total slots. This problem of buffer allocation can be stated as:

$$\begin{aligned} &\max \ P(\underline{B}) \\ s.t. \quad &\sum_{i=1}^{q} B_i = Q \\ &B_i \geq 0 \qquad\qquad i = 1, 2, \ldots, q \end{aligned} \tag{3.1}$$

where; $P(\underline{B})$ is the throughput of the $q+1$ workstation production line as a function of the buffer sizes vector. $\underline{B} = (B_1 \quad B_2 \quad \ldots \quad B_q)$ is the buffer vector, and B_i integer for all $i = 1, 2, \ldots, q$. Q is a fixed nonnegative integer representing the slots available in the production line.

6 Selecting an Optimal Subset Algorithm

The approach that proposed in this paper includes a combination of OO approach and OCBA-m method for identify the top m buffer profiles, when the number of alternatives is large. In the first stage, use the OO approach to selected randomly a subset G form the set of alternatives that overlaps with the set that contains the actual best $k\%$ buffer profiles with high probability. In stage two, the OCBA-m method use to identify all top m buffer profiles from the subset G that selected in the first stage. Clearly, the OO approach isolated the subset G of good enough buffer profiles with high probability and reducing the size of the set of alternatives so that it is appropriate to apply the OCBA-m method. The algorithm is described as follows:

Algorithm:

Setup: Determine the size of set G, $|G| = g$, where, G is defined as the selected subset from the set of all alternatives Θ, that satisfies $P(G$ contains at least one of the best $m\%$ buffer profiles$) \geq 1 - \left(1 - \frac{m}{100}\right)^g$. Let the number of initial simulation samples $t_0 \geq 5$, and the size of search space $|\Theta| = n$. Determine the total computing budget T, and the value of m (best top m). Let $l = 0$ and let $T_1^l = T_2^l = \ldots = T_g^l = t_0$, where l is the iteration number.

Select a subset G of g alternatives randomly from Θ. Take a random samples of t_0 observations Y_{ij} $(1 \leq j \leq t_0)$ for each buffer profile i in G, where $i = 1, 2, \ldots, g$.

Initialization: Calculate the sample mean $\bar{Y}_i$ and sample standard deviation s_i as; $\bar{Y}_i = \frac{1}{T_i^l} \sum_{j=1}^{T_i^l} Y_{ij}$ and $s_i = \sqrt{\frac{1}{T_i^l - 1} \sum_{j=1}^{T_i^l} (Y_{ij} - \bar{Y}_i)^2}$, where $i = 1, 2, \ldots, g$.

Order the buffer profiles in G according to their sample means (throughput); $\bar{Y}_{[1]} \geq \bar{Y}_{[2]} \geq \ldots \geq \bar{Y}_{[g]}$. Then select the top m buffer profiles from the set G, and represent this subset as S_m.

Stopping Rule: If $\sum_{i=1}^{g} T_i^l \geq T$, stop. Return S_m as the subset that containing all of the m best buffer profiles. Otherwise, select randomly a subset S_z of the $g - m$ alternatives from $\Theta - S_m$. Take $G = S_m \bigcup S_z$.

Simulation Budget Allocation: Increase the computing budget by Δ and compute the new budget allocation, $T_1^{l+1}, T_2^{l+1}, \ldots, T_g^{l+1}$ using $\frac{T_1^{l+1}}{\left(\frac{s_1}{\delta_1}\right)^2} = \frac{T_2^{l+1}}{\left(\frac{s_2}{\delta_2}\right)^2} = \cdots = \frac{T_g^{l+1}}{\left(\frac{s_g}{\delta_g}\right)^2}$, where $\delta_i = \bar{Y}_i - c$ and $c = \frac{\hat{\sigma}_{i_{m+1}} \bar{Y}_{i_m} + \hat{\sigma}_{i_m} \bar{Y}_{i_{m+1}}}{\hat{\sigma}_{i_m} + \hat{\sigma}_{i_{m+1}}}$ with $\hat{\sigma}_i = s_i / \sqrt{T_i^l}$, for all $i = 1, 2, \ldots, g$.

Perform additional $\max\{0, T_i^{l+1} - T_i^l\}$ simulations samples for each buffer profile i where $i = 1, 2, \ldots, g$, let $l \longleftarrow l + 1$. Go to **Initialization**.

7 Empirical Illustration

We present here numerical results obtained by applying the algorithm that presented in Section 6 to solve the *BAP*. Consider a production line involving $q+1$ workstations $M_0, M_1, \ldots, M_q$, modeled as single server queuing stations, and q intermediate buffers $B_1, B_2, \ldots, B_q$ as shown in Figure 2. Assume that there are unlimited supply of jobs in front of workstation M_0 and unlimited space after workstation M_q. Jobs receive service at each workstation with the service times at workstation M_i are being independent and exponentially distributed with rate μ_i, for all $i = 1, 2, \ldots, q$.

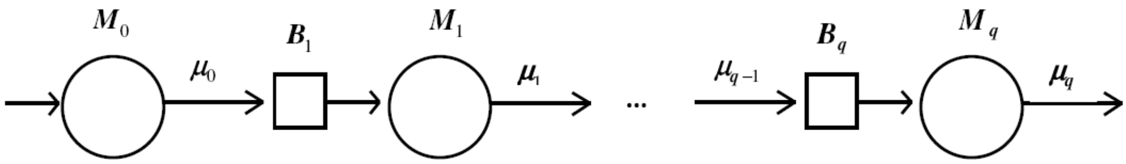

Figure 2: A production line with $q+1$ workstations, q buffers, unlimited supply jobs in front of workstation M_0, and unlimited room for all jobs departing from workstation M_q

We are interested in selecting a design that gives a maximum throughput. This mean, we are trying to solve the following maximization problem:

$$\max_{x \in \Theta} P(x) \tag{5.1}$$

where $P(x)$ is the throughput of the design, given that design x is being used and Θ is the set of all $\binom{Q+q-1}{Q}$ possible designs (alternatives). We assume that the production line here is a reliable line and we allow buffers to have zero size.

In the first example, assume that there are $Q = 15$ slots to be allocated over $q = 5$ buffers. Thus, we have 6 workstation and Θ contains 3876 different designs ($n = 3876$), and assume that $\mu_0 = \mu_1 = \mu_2 = \mu_3 = 5$ and $\mu_4 = \mu_5 = 10$, which mean that, the assumed production line in this example is unbalanced line. Furthermore, let the size of set G is $g = 50$, number of initial simulation samples $t_0 = 10$, total computing budget $T = 5000$, and the increment in simulation samples $\Delta = 30$. Suppose that our goal is selecting the top 2 design of the best 5% designs from the set Θ. Therefore, the correct selection here is selecting the 2 designs belong to set $\{\theta_1, \theta_2, \dots, \theta_{193}\}$, where θ_i, $i = 1, 2, \dots, 193$ representing the top designs that have the maximum throughput in the set Θ. The approximate analytical probability of correct selection is $P(CS) \geq 1 - (1 - 0.05)^{50} \geq 0.92$.

Table 1 contains the results of this experiment with 10 replications for selecting 2 designs of the best 5% designs, where $\sum_{i=1}^{g} T_i$ is the total sample size used in **Stopping Rule** step in the proposed algorithm, $BEST\ SUBSET$ means the index of the chosen 2 designs that being considered as the best designs, $P(\theta^*)$ is the throughput for the first selected design and $P(\theta^{**})$ is the throughput for the second selected design.

Table 1: The implementation of the proposed algorithm given the parameters $n = 3876, g = 50, \Delta = 30, k\% = 5\%, t_0 = 10, T = 5000, m = 2$

RUN	$\sum_{i=1}^{g} T_i$	$BEST\ SUBSET$	$BUFFER\ PROFILE$	$P(\theta^*)$	$P(\theta^{**})$
1	239627	{3318, 1961}	{(5 2 4 3 1), (4 2 6 2 1)}	3.88354	3.86445
2	258301	{1961, 3067}	{(5 5 3 1 1), (5 4 5 0 1)}	3.86445	3.79179
3	245247	{3344, 3126}	{(6 3 5 1 0), (4 4 4 3 0)}	3.82349	3.80021
4	229535	{2840, 3507}	{(7 0 6 1 1), (3 3 5 3 1)}	3.80984	3.90525
5	250014	{3342, 3360}	{(5 1 1 7 1), (5 5 2 3 0)}	3.84846	3.88357
6	233676	{1929, 3510}	{(6 6 2 1 0), (5 2 7 1 0)}	3.80924	3.84489
7	234135	{1930, 3370}	{(5 4 5 1 0), (4 3 3 4 1)}	3.87346	3.80966
8	241935	{2779, 2379}	{(3 5 4 3 0), (5 5 3 1 1)}	3.89886	3.84499
9	232914	{1966, 3290}	{(7 0 0 6 2), (2 6 3 4 0)}	3.79300	3.79017
10	240693	{3319, 3324}	{(6 3 4 1 1), (5 6 4 0 0)}	3.88406	3.85962

Clearly, in the first replication in Table 1, the proposed algorithm selected the designs numbered 3318 and 1961 with buffer profile (5 2 4 3 1) and (4 2 6 2 1) and the estimated throughput are 3.88354 and 3.86445, respectively. It means that we get the maximum throughput in this replication when the slots are allocated on the buffers on two ways: in the first way; the slots in B_1 is 5, the slots in B_2 is 2, the slots in B_3 is 4, the slots in B_4 is 3 and the slots in B_5 is 1, or in the second way; the slots in B_1 is 4, the slots in B_2 is 2, the slots in B_3 is 6, the slots in B_4 is 2 and the slots in B_5 is 1. For the sake of comparison, we have simulated all the 3876 designs for a long simulation run and found that the best design is 2816 with buffer profile (4 5 4 2 0) and the throughput is 3.94214. Clearly, the throughput for the 2 selected design is very closed to the throughput for the best design. Since, the 2 selected design in the first replication is belong to the best 5% designs from the set of 3876 designs, so we can call this set as a correct selection subset. In the other word, the top 2 designs that are selected in this replication are 3318 and 1961.

In the second example, consider slots $Q = 20$, to be allocated over $q = 5$ buffers. The set of all alternatives Θ contains 10626 different designs. We applied the proposed algorithm with the following parameters; $n = 10626$, $g = 100$, $t_0 = 10$, $T = 10000$ and $\Delta = 50$. The goal is selecting the top 4 designs from the best 3% designs from set Θ. Therefore, the correct selection here is selecting the 4 designs that belong to the set $\{\theta_1, \theta_2, \ldots, \theta_{318}\}$, where θ_i, $i = 1, 2, \ldots, 318$ represents the top designs in the set Θ. The approximate analytical probability of correct selection is $P(CS) \geq 1 - (1 - 0.03)^{100} \geq 0.95$. The results of the first 10 replications of this experiment are recorded in Table 2 for selecting 4 designs of the best 3% designs, the $P(\theta)$ here is the average of throughput for the 4 selected designs. We have simulated all the 10626 designs for a long simulation runs and found that the best design is numbered 7394 with a buffer profile (5 6 6 2 1) with throughput 4.10753.

Table 2: The implementation of the proposed algorithm given the parameters $n = 10626, g = 100, \Delta = 50, k\% = 3\%, t_0 = 10, T = 10000, m = 4$

$\sum_{i=1}^{g} T_i$	*BEST SUBSET*	*BUFFER PROFILE*	$P(\theta)$
2211127	{6369,6561,7335,7327}	{(8 4 5 3 0),(7 9 2 2 0),(5 6 4 5 0),(7 1 6 3 3)}	4.03099
2300132	{6451,5565,5606,6570}	{(3 4 6 5 2),(4 5 2 5 4),(5 4 5 0 6),(4 7 3 3 3)}	4.00604
2198462	{5610,5655,6618,7336}	{(5 3 6 4 2),(4 5 2 9 0),(6 0 5 4 5),(3 4 3 3 7)}	4.03030
2200043	{6569,7338,6575,5744}	{(3 9 5 1 2),(3 3 4 2 8),(8 4 1 5 2),(9 0 7 3 1)}	4.01354
2188779	{6520,6690,5613,8019}	{(4 4 5 3 4),(7 5 2 2 4),(5 3 4 5 3),(5 3 6 3 3)}	4.05654
2213453	{9587,8109,7327,5411}	{(4 5 1 6 4),(4 4 4 2 6),(5 5 5 0 5),(3 3 3 3 8)}	4.04306
2200765	{8749,7474,8487,9835}	{(4 5 6 2 3),(5 7 4 2 2),(7 6 5 0 2),(8 8 4 0 0)}	4.04633
2159997	{7474,6448,7431,5297}	{(2 5 4 4 5),(9 4 4 2 1),(3 5 6 2 4),(9 7 0 2 2)}	3.99662
2223101	{7334,6517,7440,9589}	{(3 10 5 0 2),(3 8 4 2 3),(8 4 5 1 2),(9 8 0 3 0)}	4.10604
2211453	{8744,8069,8158,8161}	{(5 4 6 3 2),(4 9 2 5 0),(6 4 5 0 5),(3 7 3 3 4)}	3.97463

In the first replication in Table 2, the proposed algorithm selected the designs numbered 6369, 6561, 7335 and 7327 with buffer profile (8 4 5 3 0),(7 9 2 2 0),(5 6 4 5 0) and (7 1 6 3 3), respectively, and the estimated average of throughput 4.03099. Actually, the values of the throughput for the designs 6369, 6561, 7335 and 7327 are 4.00306, 4.05654, 4.0371 and 4.02729, respectively. Clearly in this replication we get the maximum throughput when the slots are allocated on the buffers on 4 ways: in the first way; the slots in B_1 is 8, the slots in B_2 is 4, the slots in B_3 is 5, the slots in B_4 is 3 and the slots in B_5 is 0, or in the second way; the slots in B_1 is 7, the slots in B_2 is 9, the slots in B_3 is 2, the slots in B_4 is 2 and the slots in B_5 is 0, or in the third way; the slots in B_1 is 5, the slots in B_2 is 6, the slots in B_3 is 4, the slots in B_4 is 5 and the slots in B_5 is 0, or in the fourth way; the slots in B_1 is 7, the slots in B_2 is 1, the slots in B_3 is 6, the slots in B_4 is 3 and the slots in B_5 is 3. Clearly, the throughput for the 4 selected design is very closed to the throughput for the best design, which is 7394 with a buffer profile (5 6 6 2 1) with throughput 4.10753. Since, the 4 selected design in the first replication is belong to the best 3% designs from the set of 10626 designs, so we can call this set as a correct selection subset. In the other word, the top 4 designs that are selected in this replication are 6369, 6561, 7335 and 7327.

These two experiments are then repeated for 100 replications, and the results are summarized in Table 3, with $\overline{\sum_{i=1}^{g} T_i}$ represents the average number of the total sample size in the **Stopping Rule** in the proposed algorithm. Clearly, the proposed algorithm that presented in Section 6 selected the best buffer profile with high $P(CS)$ and is closed to the analytical values. In the same time, the number of simulation samples that are being used are relatively small.

Table 3: The performance of the proposed algorithm over 100 replications

Number of workstations	Number of Buffers	Slots	n	$\overline{\sum_{i=1}^{g} T_i}$	Approach $P(CS)$	Analytical $P(CS)$
6	5	15	3876	250232	0.88	0.92
6	5	20	10626	2130593	0.80	0.95

8 Open Problem

The difficulty in buffer allocation problem comes from a situation that it has a huge number of feasible allocations with respect to the total number of buffer spaces and the number of stations in production line. The proposed selection approach is useful and reliable to find the optimal allocation of buffers that maximizes the mean production rate in a short, unbalanced and reliable production lines. Future research should be conducted to solve another type of production lines with a different objective function such as to minimize the average work in process by using the proposed approach, and to conduct a comparison between the proposed approach and other approaches that are being used to solve problem.

9 Conclusion

We have shown how to solved the buffer allocation problem for a finite production line, by using a new selection approach. In this approach we start with ordinal optimization approach to select a relative small subset with a probability of overlapping with the subset that contains the actual best $k\%$ designs is high. Next, we allocate the available computing budget using the OCBA-m method to identify all top m designs from the subset that selected by ordinal optimization in the first stage. The advantage of our selection approach is that, it can be used to select the best buffer profile from a large number of alternatives, this is because it use the ordinal optimization approach in order to decrease the number of the competing alternatives, to be appropriate for the optimal computing budget allocation method to selecting the best subset of buffer profiles. Numerical illustrations demonstrate that the proposed algorithm finds the top m buffer profiles in a short, unbalanced and reliable production lines. Moreover, the algorithm is able to allocate buffer profiles with maximum throughput using a relatively small simulation replications, at the same time with high probability of correct selection.

References

[1] S.K. Chaharsooghi, N. Nahavandi, Buffer allocation problem, a heuristic approach. *Scientia Iranica*, 10(4) (2003) 401–409.

[2] M.H. Almomani, R. Abdul Rahman, A. Baharum, M. H. Alrefaei, A Selection Approach for Solving the Buffer Allocation Problem. *International Journal of the Physical Sciences*, 7(3) (2012) 413–422.

[3] Y.C. Ho, R.S. Sreenivas, P. Vakili, Ordinal optimization of DEDS, *Discrete Events Dynamic Systems*, 2 (1992) 61–88.

[4] C.H. Chen, D. He, M. Fu, L.H. Lee, Efficient simulation budget allocation for selecting an optimal subset, *Informs Journal on Computing*, 20(4) (2008) 579-595.

[5] Papadopoulos, C. T., O'kelly, M. E. J., Vidalis, M. J., and Spinellis D. (2009). *Analysis and Design of Discrete Part Production Lines.* Springer.

[6] Spinellis, D. D., and Papadopoulos, C. T. (2000). A simulated annealing approach for buffer allocation in reliable production lines. *Annals of Operations Research*, **93**, 373-384.

[7] Alon, G., Kroese, D. P., Raviv, T., and Rubinstein, R. Y. (2005). Application of the Cross-Entropy Method to the Buffer Allocation Problem in a Simulation-Based Environment. *Annals of Operations Research*, **134**, 137-151,

[8] Yuzukirmizi, M., and Smith, J. M. (2008). Optimal buffer allocation in finite closed networks with multiple servers. *Computers & Operations Research*, **35**, 2579–2598.

[9] Alrefaei, M., and Andradóttir, S. (2005).Discrete stochastic optimization using variants of the stochastic ruler method. *Naval research Logistics*, **52**(4), 344–360.

[10] Daskalaki, S., and Smith, J. M. (2004). Combining routing and buffer allocation problems in series-parallel queuing networks. *Annals of Operations Research*, **125**, 47–68.

[11] Foley, R. D., and Park, B. C. (2002). Optimal allocation of buffers and customers in a two-node cyclic network with multiple servers. *Operations Research Letters*, **30**, 19–24.

[12] Kim, D. S., Kulkarni, D. M., and Lin, F. (2002). An upper bound for carriers in a three-workstation closed serial production system operating under production blocking. *IEEE Transactions on Automatic Control*, **47**(7), 1134–1138.

[13] Huang, M. G., Chang, P. L., and Chou, Y. C. (2002). Buffer allocation in flow-shop-type productions systems with general arrival and service patterns. *Computer & Operations Research*, **29**, 103–121.

[14] Diamantidis, A. C., and Papadopoloulos, C. T. (2004). A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines. *Mathematical problems in Engineering*, **3**, 209–223.

[15] M. Deng, Y.C. Ho, An ordinal optimization approach to optimal control problems. *Automatica*, 35 (1999) 331–338.

[16] L.H. Lee, T.W.E. Lau, Y.C. Ho, Explanation of goal softening in ordinal optimization. *IEEE Transactions on Automatic Control*, 44 (1999) 94–99.

[17] Q.C. Zhao, Y.C. Ho, Q.S. Jia, Vector ordinal optimization. *Journal of Optimization Theory and Applications*, 125 (2005) 259-274.

[18] Y.C. Ho, Q.C. Zhao, Q.S. Jia, *Ordinal Optimization: Soft Optimization for Hard Problems*. Springer, 2007.

[19] C. H. Chen, E. Yucesan and S. E. Chick. Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization. *Discrete Event Dynamic Systems* 10(3) (2000) 251–270.

[20] M.H. Almomani, R.Abdul Rahman, Selecting a good stochastic system for the large number of alternatives, *Communications in Statistics-Simulation and Computation*, 41 (2012) 222–237.

[21] M.H. Almomani, R. Abdul Rahman, Selecting a good enough simulated design with opportunity cost, *International Journal of Mathematics and Computation*, 7(J10) (2010) 114–124.

[22] M.H. Almomani, A. Baharum, R. Abdul Rahman, Three-stage procedure and expected opportunity cost for selecting a good simulated design, *Research Journal of Applied Sciences*, 5(6) (2010) 417–423.

[23] M.H. Almomani, R. Abdul Rahman, A. Baharum. Selecting a Good Enough Stochastic Design. *Far East Journal of Applied Mathematics*, 53(2) (2011) 123–131.

[24] M.H. Almomani, M.H. Alrefaei, A three-stage procedure for selecting a good enough simulated system, *Journal of Applied Probability and Statistics*, 6(1&2) (2012) 13–27.

[25] M.H. Alrefaei, M.H. Almomani, Subset selection of best simulated systems. *Journal of the Franklin Institute*, 344 (2007) 495–506.

[26] C.H. Chen, D. He, M. Fu, L.H. Lee, Efficient selection of an optimal subset for optimization under uncertainty, *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*, (2007) 1-5.

[27] C.H. Chen, L.H. Lee, Stochastic Simulation Optimization: An Optimal Computing Budget Allocation. Chapter 5, World Scientific Publishing Company, 2010.