Received: 9 March 2015 | Revised: 23 December 2016 | Accepted: 30 May 2017

DOI: 10.1002/oca.2341

RESEARCH ARTICLE

WILEY

Optimization of buffer allocation in unreliable production lines based on availability evaluation

Bing-hai Zhou[1] | Yu-wang Liu[1] | Jia-di Yu[1] | Da Tao[2]

[1] School of Mechanical and Energy Engineering, Tongji University, Shanghai, PR China

[2] Department of Industrial & Manufacturing Systems Engineering, the University of Hong Kong, Hong Kong, PR China

Correspondence
Bing-hai Zhou, School of Mechanical and Energy Engineering, Tongji University, Shanghai 201804, PR China.
Email: bhzhou@tongji.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 71471135

Summary

The buffer allocation problem is an important issue in production lines design. In this paper, we present new evaluation and optimization methods to optimally allocate buffers in unreliable production lines. Through analyzing different states of the machines and buffers by Markov process and incorporating the aggregation method, we make an evaluation on the system availability, instead of the throughput rate of the line. The optimization method is proposed by combining particle swarm optimization and estimation of distribution algorithm to maximize the system availability. It generates the new populations by estimation of distribution algorithm and particle swarm optimization to take their respective advantages in global and local optimization. Numerical tests and simulations are performed to validate the performance of the evaluation and optimization methods. The results indicate the effectiveness and efficiency of the proposed methods.

KEYWORDS

aggregation method, availability, buffer allocation problem, Markov, serial production line

1 | INTRODUCTION

Buffer allocation problem (BAP) is an important optimization problem faced by manufacturing system designers, which deals with finding the optimal buffer sizes to be allocated into buffer areas in a production system.[1] For unreliable production lines, the flow of parts may be disrupted by machine failures. The effects of such variations can be reduced using interstage buffers between the machines. However, allocating buffers into a production line may result in additional costs because of increased capital investment and in-process inventory, and there is generally a physical limit to the floor space in the system. Thus, the optimal allocation of buffers among the machines so as to improve the system performance is an important manufacturing design problem.

Two methods, ie, evaluative and optimization methods, are generally used iteratively to solve the BAP. The first method is used to calculate the value of the objective function, which is usually the production rate of the system. The second method is applied to find the optimal or near optimal sizes of the buffers. These 2 methods communicate with each other in an iterative manner within a closed loop configuration.[2]

Evaluative methods, which provide estimates of various performance measures such as throughput rate and mean queue lengths, are based on analytical procedures and simulation.[3] Analytical approaches are usually applicable under certain assumptions such as specific processing time distributions or small number of station. An exact analytic formula was derived to calculate the production rate for a 2-machine–1-buffer line with deterministic processing times in the work of Buzacott.[4] In the work of Kokangul et al,[5] a mathematical model on the basis of closed-form asymptotic methods was developed, and the asymptotic results for the expected value of the stock level in a buffer was provided as a function of time. In the work of Ouazene et al,[6] a new analytical formulation called equivalent machine method was proposed to evaluate the system throughput, which considered both homogeneous and nonhomogeneous serial lines. Starting from a time-discrete recursion, Kolb and Göttlich[7] derived a time-continuous buffer allocation model supplemented with a stochastic process. Li et al[8] summarized the recent studies devoted to developing approximate analytical methods to estimate the throughput of production lines and drew the conclusion that most of these methods were based on decomposition or aggregation techniques. In the work of Gershwin,[9] an approximate decomposition approach was presented for evaluating the production rate for a class of tandem queuing systems with finite buffers. Dallery et al[10] developed the decomposition equations for the continuous material model and the DDX (Dallery-David-Xie) algorithm for the homogeneous lines to evaluate the throughput. In the work of De Koster,[11] the aggregation approach was tested to estimate the efficiency of an unreliable line with exponential failure and repair times. Other methods for such production lines are mostly based on simulation techniques. In the work of Sörensen and Janssens,[12] a petri-net–based simulation model was developed to study a continuous flow transfer line with 3 machines and 2 buffers, on the basis of which the relationship between the equipment reliability and buffer capacity was analyzed. In the work of Amiri and Mohtashami,[13] assuming the process times, time between failures and repair times were general function distributions, the authors proposed a discrete event simulation methodology for estimating production rate.

Most of the previous studies focus on the estimation of the production rate on the basis of various evaluative methods, to the best of our knowledge, ours is the first extensive study dealing with BAP to take system availability as the objective function, which has been well established in the literature of stochastic modeling and optimal maintenance.[14] In fact, the throughput of the production lines is directly affected by the line steady-state availability. Moreover, availability is also one of the principal indicators of measuring random machine failures during operations. Thus, it is of great significance to take availability into account while making optimizations for the unreliable production lines.

Regarding to the optimization methods, various optimization techniques have been proposed. Complete enumeration is the simplest one, but evidently, it is applicable to only small-sized problems. Because the number of feasible solutions grows exponentially as the number of machines and total buffer capacity increase, it is impractical to use complete enumeration for large-sized problems. Consequently, researchers widely adopt various search procedures and metaheuristics. In the work of Diamantidis and Papadopoulos,[15] a newly developed dynamic programming algorithm was proposed for the BAP in homogeneous, asymptotically reliable serial production lines. In the work of Yuzukirmizi et al,[16] the solution optimization procedure was incorporated into a nonlinear optimization scheme by using Powell search method. An analytical decomposition-type approximation, ie, degrading ceiling algorithm was used to estimate the production line throughput in the work of Nahas et al.[17] In the work of Cruz et al,[18] a Lagrangian relaxation approach was proposed for the two-moment approximation formula to obtain the optimal buffer allocation in general service time single queues. In the work of Shi and Gershwin,[19] a nonlinear programming approach was adopted to maximize the profits through buffer size optimization for production lines with a nonlinear production rate constraint.

Recently, meta-heuristic approaches have been widely used to solve the BAP. Tabu search algorithm including some problem specific features was proposed to solve the BAP in long production lines.[20,21] In the work of Amiri and Mohtoshami,[13] genetic algorithm (GA) combined to line search method was used to solve the multiobjective formulation of the BAP model. Particle swarm optimization (PSO) was used as generative technique to optimize the buffer allocation in the work of Narasimhamu et al.[22] In the work of Nahas et al,[23] ant colony optimization and simulated annealing were compared empirically to solve the new formulated BAP, where the decision variables were buffers and the number of parallel machines. To better search the solution space, the recent trend is to hybridize the metaheuristics with other methods, such as nested partitions[24] and branch and bound methods.[25] Costa et al[26] presented a parallel tabu search algorithm equipped with a proper adaptive neighborhood generation mechanism to solve the BAP. In the work of Massim et al,[27] a combined artificial immune system optimization algorithm in conjunction with a decomposition method was implemented to allocate buffers in transfer lines.

In this study, a novel framework of hybridizing PSO with estimation of distribution algorithm (EDA) is designed to obtain the optimal or near optimal solutions of the BAP. Since first introduced by Kennedy and Eberhart,[28] PSO has been widely used as an optimization method, while it suffers the weakness of local-search intensive as stated in the work of Ahn et al.[29] Fortunately, through incorporating the graphical probabilistic modeling into the linkage learning, EDA has the great advantage in automatic discovery and exploitation of problem regularities. Therefore, researchers have attempted to combine PSO with EDA while maintaining their respective strengths simultaneously.[30,31]

To the best of the authors' knowledge, this is the first study taking the system availability as the new evaluation criterion for the BAP. What is more, a hybrid optimization method that combines PSO and EDA is proposed to solve the problem in unreliable production lines. The remainder of the paper is organized as follows. Section 2 describes the BAP and the related assumptions. Section 3 presents a novel aggregation method used to evaluate the objective function. Section 4 provides the details of a hybrid PSO-EDA optimization tool for the BAP. Numerical tests and simulations as well as discussion of the results are presented in Section 5. Finally, Section 6 draws the conclusion and presents future work of our study.

2 | PROBLEM STATEMENT AND RELATED ASSUMPTIONS

In this paper, we examine the BAP in a serial production line, as depicted in Figure 1, where the rectangles represent machines, and the circles indicate buffers, denoted as M_i, i = 1 , … , n, and B_i, i = 1 , … , n − 1, respectively. The assumptions of the BAP in a serial production line are listed as follows:

(1) Each part goes through all machines and buffers in sequence, from machine M_1 to machine M_n.
(2) Machine M_i is starved at time t if $M_{i - 1}$ is down and buffer $B_{i - 1}$ is empty; machine M_i is blocked at time t if $M_{i + 1}$ is down and buffer B_i is full.
(3) Machine M_1 has an unlimited supply of parts, while machine M_n possesses an unlimited storage area. Thus, machine M_1 is never starved and machine M_n is never blocked.
(4) The processing times of all parts are constant and equal for all machines and are considered as a time unit. Transportation time is negligible.
(5) The failures are operation dependent and detected instantaneously. A machine cannot fail when it is starved or blocked. Times to failure and times to repair for the machines are independent and exponentially distributed.
(6) The machines can be fully repaired.
(7) The state transition process has Markov property (ie, the state probability distribution of future states of a machine depends only on the present state, rather than on its preceding events).

A serial production line consisting of n machines and n − 1 buffers is examined (see Figure 1). The objective is to maximize the availability of the production line by optimizing buffer allocation with a given total buffer capacity. The mathematical model for the problem is formulated as follows:Find $S = (S_1,S_2, \cdots\cdots ,S_{n-1})$ so as to

$$\text{Maximize } A = f(S) \tag{1}$$

Subject to

$$\sum_{i=1}^{n-1} S_i = Q \tag{2}$$

$$0 \le S_i \le S_{iup} \quad (i = 1, \cdots, n-1) \tag{3}$$

$$S_i \text{ nonnegative integers } (i = 1, \cdots, n-1) \tag{4}$$

where Q is the total buffer capacity available in the production line, which is a fixed nonnegative integer; $S = (S_1,S_2, \cdots\cdots ,S_{n-1})$ represents the vector of buffer capacities, S_i, $i = 1$, … , $n - 1$, which are integer numbers; and $f(S)$ denotes the availability of the production line. Constraint 3 shows upper bounds (S_{iup}) for each of the buffer locations. It should be noted that the sum of the upper bounds (S_{iup}) for all buffer locations will be larger than Q.The number of feasible buffer configurations of Q buffer capacity among the $n - 1$ intermediate buffer locations increases dramatically with Q and n, as shown in Equation 5.

$$C_{Q+n-2}^{n-2} = \frac{(Q+1)(Q+2)\ldots(Q+n-2)}{(n-2)\ !} \tag{5}$$

M_1
B_1
M_2
B_2
B_{i-1}
M_i
B_{n-1}
M_n

FIGURE 1 A serial production line

3 | EVALUATION METHOD

The basic idea of the aggregation method is to replace a 2-machine–1-buffer subsystem by a single equivalent machine that has the same availability in isolation, as illustrated in Figure 2. The obtained equivalent machine is further combined with the next equivalent machine to obtain a new aggregated machine. Finally, the whole transfer line can be aggregated in a single machine. To apply the aggregation technique as mentioned above, we have to evaluate the performance of the 2-machine–1-buffer subsystem firstly.

The following notations are used in the mathematical formulation of the model:

λ_i	failure rate of machine M_i
μ_i	repair rate of machine M_i
ω_i	processing rate of machine M_i
Sub_i	the *i*th subsystem in the production line
Sys_i	state space of Sub_i
$X_{i,j}$	the jth state of Sub_i
P_{Sys_1}	state transition matrix of Sub_i
P^s_i	steady-state probability matrix of Sub_i
$P^s_{i,j}$	steady-state probability when the state of Sub_i is $X_{i,j}$
Buf_i	state space of buffer B_i
$P_{Buf_i,j}$	state transition matrix of buffer B_i when the state of Sub_i is $X_{i,j}$
$Q^s_{i,j}$	steady-state probability matrix of buffer B_i when the state of Sub_i is $X_{i,j}$
$Q^s_{i,j,k}$	steady-state probability when the state of Sub_i is $X_{i,j}$ and the number of work-in-process in buffer B_i is k
$\mathrm{Num}_{i,j}$	number of work-in-process in buffer B_i when the state of Sub_i is $X_{i,j}$
$P_{Num_{i,j}}$	probability with which Sub_i is operational when the state of Sub_i is $X_{i,j}$ and the number of work-in-process in buffer B_i is k
A^s_i	steady-state availability of Sub_i
M^j_i	the *i*th equivalent machine in jth aggregation stage
λ^j_i	failure rate of equivalent machine M^j_i
μ^j_i	repair rate of equivalent machine M^j_i
ω^j_i	processing rate of equivalent machine M^j_i
A	steady-state availability of the production line
E	throughput of the production line
as_i	average number of work-in-process in buffer B_i
CW	average number of work-in-process in the production line

3.1 | The 2-machine–1-buffer subsystem

The 2-machine–1-buffer subsystem Sub_1 (see Figure 3) consists of an upstream machine M_1, a downstream machine M_2, and an intermediate buffer B_1. The availability of the subsystem is determined by the intermediate buffer capacity and the parameters of the 2 adjacent machines. Firstly, we classify the machines states of subsystem, describe the state transition rule, and calculate the steady-state probabilities of each subsystem state. Then the steady-state probability of each buffer state is estimated with the same process as mentioned above. Lastly, the availability of subsystem is calculated by analyzing the subsystem and buffer-state transition rules.

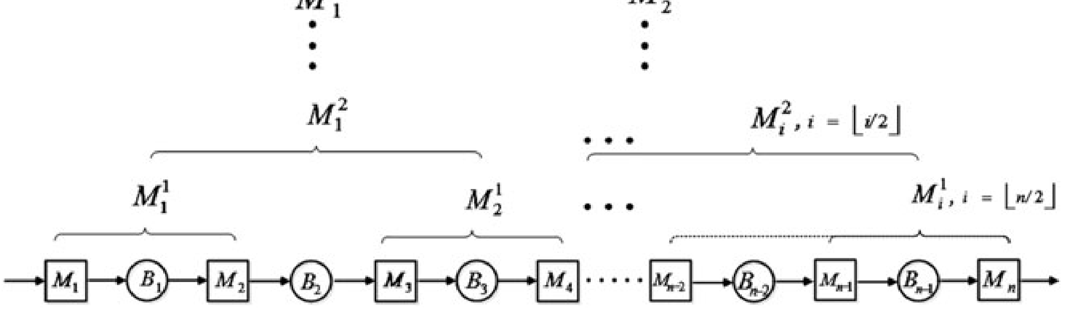

FIGURE 2 Aggregation method of a transfer line

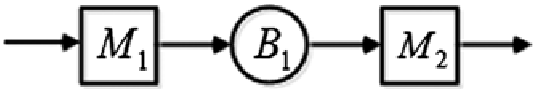

FIGURE 3 Two-machine-one-buffer subsystem *Sub*$_1$

3.1.1 | The machines state transition rule of a subsystem

Each of the 2 machines in the subsystem can be in 1 of 2 states: operational, ie, capable for producing parts; or down, ie, under repair. Therefore, the subsystem can be in 1 of 4 machines states. The state space of subsystem Sub_1 can be written as

$$\mathrm{Sys}_1 = \left\{X_{1,1}, X_{1,2}, X_{1,3}, X_{1,4}\right\} \tag{6}$$

where state $X_{1\,,\,1}$ represents that both M_1 and M_2 are operational; $X_{1\,,\,4}$ represents both M_1 and M_2 are down; state $X_{1\,,\,2}$ denotes that M_1 is operational while M_2 is down; and state $X_{1\,,\,3}$ indicates M_1 is down while M_2 is operational.

The states of Sub_1 can be transformed among each other because of the failure of the machines (see Figure 4).The state transition rule can be obtained through analyzing the failure rate and repair rate of machine M_1 and M_2 on the basis of the Markov theory. The state transition matrix of Sub_i can be written as

$$P_{\mathrm{Sys}_1} = \begin{bmatrix} (1-\lambda_1)(1-\lambda_2) & (1-\lambda_1)\lambda_2 & \lambda_1(1-\lambda_2) & \lambda_1\lambda_2 \\ (1-\lambda_1)\mu_2 & (1-\lambda_1)(1-\mu_2) & \lambda_1\mu_2 & \lambda_1(1-\mu_2) \\ \mu_1(1-\lambda_2) & \mu_1\lambda_2 & (1-\mu_1)(1-\lambda_2) & (1-\mu_1)\lambda_2 \\ \mu_1\mu_2 & \mu_1(1-\mu_2) & (1-\mu_1)\mu_2 & (1-\mu_1)(1-\mu_2) \end{bmatrix} \tag{7}$$

According to the Markov theory, the system will eventually reach a stable state in a sufficiently long time. Therefore, the balance equations of the machines states of the subsystem can be written as

$$\begin{cases} P_1^s P_{Sys_1} = P_1^s \\ \sum\limits_{j=1}^{4} P_{1,j}^s = 1 \end{cases} \tag{8}$$

where $P_1^s = \left[P_{1,1}^s, P_{1,2}^s, P_{1,3}^s, P_{1,4}^s\right]$. The steady-state probability matrix $P_1^s$of Sub_1 can be obtained by solving Equation 8.

3.1.2 | The buffer-state transition rule of a subsystem

To analyze the steady states of the buffer, we consider a birth-death Markov process with $(\mathrm{S}_1 + 1)$ states. The state space of buffer B_1 can be written as

$$\mathrm{Buf}_1 = \{0,\ 1,\ 2,\ \cdots, S_1-1, S_1\} \tag{9}$$

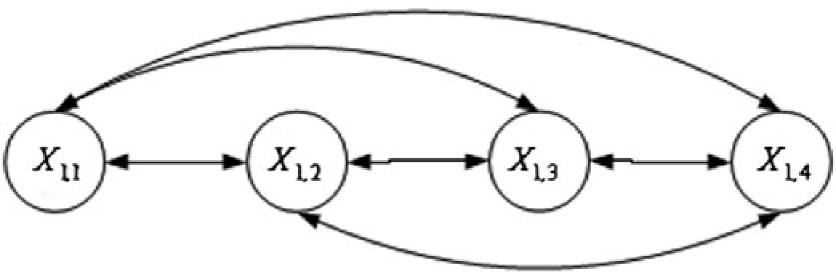

FIGURE 4 Transition of machines states for *Sub*$_1$

It should be noted that the buffer-state transition rules depend on the states of the subsystem. The following section presents the buffer-state transition rules in each of the subsystem states, except for $X_{1,4}$, in which the number of work in process in buffer B_1 remains unchanged.

1) The state of Sub_1 is $X_{1,1}$: Both of the 2 machines are operational. One part moves into buffer B_1 from upstream machine M_1, and another part moves to downstream machine M_2 from the buffer B_1. Figure 5 illustrates the buffer-state transition under the state of $X_{1,1}$.

The state transition matrix of buffer B_1 in the machines state of $X_{1,1}$ can be written as

$$P_{Buf_1,1} = \begin{bmatrix} 1-\eta & \eta & 0 & \cdots & 0 & 0 & 0 \\ \alpha & \varepsilon & \eta & \cdots & 0 & 0 & 0 \\ 0 & \alpha & \varepsilon & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha & \varepsilon & \eta \\ 0 & 0 & 0 & \cdots & 0 & \alpha & 1-\alpha \end{bmatrix} \tag{10}$$

where$\alpha=(1-\omega_1)\omega_2$,$\eta=\omega_1(1-\omega_2)$,$\varepsilon=\omega_1\omega_2+(1-\omega_1)(1-\omega_2)$

2) The state of Sub_1 is $X_{1,2}$: M_1 is operational while M_2 is down. One part moves into buffer B_1 from upstream machine M_1, and no part moves to downstream machine M_2. Figure 6 illustrates the buffer-state transition under the state of $X_{1,2}$.

The state transition matrix of buffer B_1 in the machines state of $X_{1,2}$ can be obtained as

$$P_{Buf_1,2} = \begin{bmatrix} 1-\omega_1 & \omega_1 & 0 & \cdots & 0 & 0 \\ 0 & 1-\omega_1 & \omega_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1-\omega_1 & \omega_1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

3) The state of Sub_1 is $X_{1,3}$: M_1 is down while M_2 is operational. No part moves into buffer B_1 from upstream machine M_1 and one part moves to downstream machine M_2 from buffer B_1. Figure 7 illustrates the buffer-state transition under the state of $X_{1,3}$.

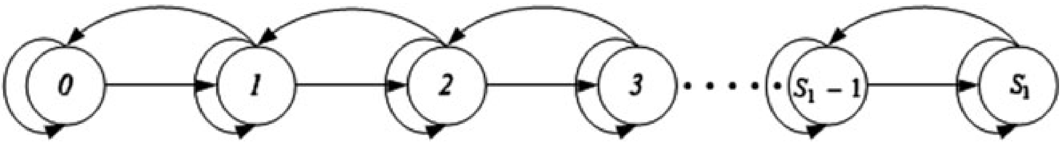

FIGURE 5 Buffer-state transition under the machines state of $X_{1,1}$

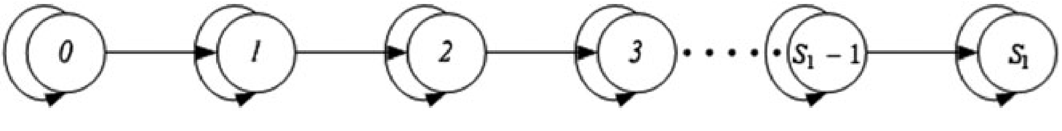

FIGURE 6 Buffer-state transition under the machines state of $X_{1,2}$

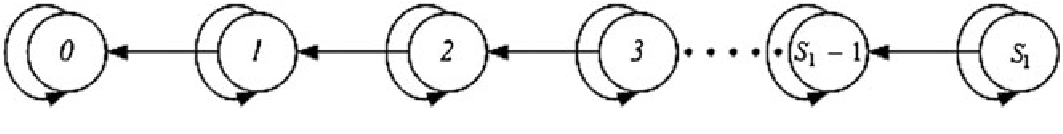

FIGURE 7 Buffer-state transition under the machines state of $X_{1,3}$

The state transition matrix of buffer B_1 in the machines state of $X_{1\,,\,3}$ can be expressed as

$$P_{Buf_1,3}=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \omega_2 & 1-\omega_2 & 0 & \cdots & 0 & 0 \\ 0 & \omega_2 & 1-\omega_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-\omega_2 & 0 \\ 0 & 0 & 0 & \cdots & \omega_2 & 1-\omega_2 \end{bmatrix} \tag{12}$$

The dimensions for the buffer-state transition matrices $P_{Buf_1,1}$, $P_{Buf_1,2}$, and $P_{Buf_1,3}$ are all $(\mathrm{S}_1+1)\times(\mathrm{S}_1+1)$. Therefore, the subsystem balance equations of the buffer states can be written as

$$\begin{cases} Q^s_{1,j}P_{\mathrm{Buf}_1,j}=Q^s_{1,j} \\ \sum\limits_{k=0}^{S_1+1}Q^s_{1,j,k}=1 \end{cases} \tag{13}$$

where $Q^s_{1,j}=\left[Q^s_{1,j,0},Q^s_{1,j,1},Q^s_{1,j,2},\cdots Q^s_{1,j,S_1-1},Q^s_{1,j,S_1}\right]$,$\mathrm{j}\in\{1,2,3\}$.

The steady-state probability matrices $Q^s_{1,j}$ ($\mathrm{j}\in\{1,2,3\}$) of Sub_1 can be obtained by solving the Equation 13 if there is no machine repair. However, the buffer may not reach stable state under equipment maintenance, in which the steady-state probability distribution should be revised.

3.1.3 | The steady-state availability of a subsystem

As a matter of fact, if one of the machines breaks down in the subsystem, the upstream and downstream machines can still work until the intermediate buffers are full and empty respectively. Therefore, the subsystem is operational under the following 3 scenarios:

1) The state of Sub_1 is $X_{1\,,\,1}$;
2) The state of Sub_1 is $X_{1\,,\,2}$ and the buffer is not full;
3) The state of Sub_1 is $X_{1\,,\,3}$ and the buffer is not empty.

When the state of Sub_1 is $X_{1\,,\,1}$, both of the 2 machines are operational; thus, the probability of the subsystem being operational equals to 1.

With equipment maintenance, the probability of the subsystem being operational under the state of $X_{1\,,\,2}$ (or $X_{1\,,\,3}$) is determined by initial state of the buffer B_1, the processing time of machine M_1 (or M_2), and the downtime of machine M_2 (or M_1).When the state of Sub_1 is $X_{1\,,\,2}$, the probability of the subsystem being operational (ie, M_1 is not blocked) can now be written as

$$P_{Num_{1,2}}=\begin{cases} 1 & \dfrac{S_1-Num_{1,2}+1}{\omega_1}\geq\dfrac{1}{\mu_2} \\ \dfrac{(S_1-Num_{1,2}+1)\times\mu_2}{\omega_1} & \dfrac{S_1-Num_{1,2}+1}{\omega_1}<\dfrac{1}{\mu_2} \end{cases} \tag{14}$$

where $\mathrm{Num}_{1\,,\,2}\in\mathrm{Z}$, $0\leq\mathrm{Num}_{1\,,\,2}\leq\mathrm{S}_1$, and $\mathrm{Num}_{1\,,\,2}$ follows the probability distribution of $Q^s_{1,1}$.When the state of Sub_1 is $X_{1\,,\,3}$, the probability of the subsystem being operational (ie, M_2 is not starved) can be written as

$$P_{Num_{1,3}}=\begin{cases} 1 & \dfrac{Num_{1,3}+1}{\omega_2}\geq\dfrac{1}{\mu_1} \\ \dfrac{(Num_{1,3}+1)\times\mu_1}{\omega_2} & \dfrac{Num_{1,3}+1}{\omega_2}<\dfrac{1}{\mu_1} \end{cases} \tag{15}$$

where $\mathrm{Num}_{1,3} \in Z$, $0 \le \mathrm{Num}_{1,3} \le S_1$, and $\mathrm{Num}_{1,3}$ follows the probability distribution of $Q_{1,1}^s$.The steady-state availability of the subsystem is calculated as

$$A_1^s = P_{1,1}^s \times \sum_{k=0}^{S_1} Q_{1,1,k} + P_{1,3}^s \times \sum_{Num_{1,3}=0}^{S_1} \left(Q_{1,1,Num_{1,3}} \times P_{Num_{1,3}}\right) + P_{1,2}^s \times \sum_{Num_{1,2}=0}^{S_1} \left(Q_{1,1,Num_{1,2}} \times P_{Num_{1,2}}\right) \tag{16}$$

3.2 | General model with n machines and n - 1 buffers

Combining with the property of reliability, the failure rate, repair rate, and processing rate of equivalent machine M_1^1 can be calculated as follows:

$$\lambda_1^1 = 1-(1-\lambda_1)(1-\lambda_2)-(1-\lambda_1)\lambda_2 \sum_{Num_{1,2}=0}^{S_1} \left(Q_{1,1,Num_{1,2}} \times P_{Num_{1,2}}\right) - \lambda_1(1-\lambda_2) \sum_{Num_{1,3}=0}^{S_1} \left(Q_{1,1,Num_{1,3}} \times P_{Num_{1,3}}\right) \tag{17}$$

$$\mu_1^1 = \frac{A_1^s \lambda_1^1}{1-A_1^s} \tag{18}$$

$$\omega_1^1 = \min(\omega_1, \omega_2) \tag{19}$$

According to the aggregation method, the whole transfer line can be aggregated in a single machine by solving Equations 6 to 19. The procedural form of the aggregation method on the basis of Markov theory (AMM) is given as follows:

Step 1. Decompose the serial production line with n machines into a set of i subsystems, where $i = \left\lfloor \frac{n}{2} \right\rfloor$. If $n = 2i$, the i subsystems are $\mathrm{Sub}_1 = \{\mathrm{M}_1,\mathrm{B}_1,\mathrm{M}_2\}$, $\mathrm{Sub}_2 = \{\mathrm{M}_3,\mathrm{B}_3,\mathrm{M}_4\} \ldots \mathrm{Sub}_\mathrm{i} = \{\mathrm{M}_{2\mathrm{i}-1},\mathrm{B}_{2\mathrm{i}-1},\mathrm{M}_{2\mathrm{i}}\}$. Ifn = 2i − 1, the i subsystems are $\mathrm{Sub}_1 = \{\mathrm{M}_1,\mathrm{B}_1,\mathrm{M}_2\}$,$\mathrm{Sub}_2 = \{\mathrm{M}_3,\mathrm{B}_3,\mathrm{M}_4\}$ … $\mathrm{Sub}_{\mathrm{i}-1} = \{\mathrm{M}_{2\mathrm{i}-3},\mathrm{B}_{2\mathrm{i}-3},\mathrm{M}_{2\mathrm{i}-2}\}$ and the last machine M_n.

Step 2. Calculate the availabilities of the i subsystems by using Equations 6 to 16 and obtain i equivalent machines in the jth aggregation stage $\left\{M_1^j, M_2^j, \cdots, M_i^j\right\}$ by solving Equations 17 to 19. Then a new serial production line is obtained, which consists of the i equivalent machines $\left\{M_1^j, M_2^j, \cdots, M_i^j\right\}$ and $i - 1$ buffers $\{\mathrm{B}_2,\mathrm{B}_4, \cdots,\mathrm{B}_{2\mathrm{i}-2}\}$ that are not aggregated in step 1.

Step 3. Update the number of machines n = i and aggregation stage j = j + 1. If $n \le 3$, which is the stopping criterion of the aggregation process, then $nc < nc_{\max}$ stop the aggregation. Otherwise, go to step 1.

Finally, a new production line $\left\{M_1^k, B_{2k}, M_2^k\right\}$ that is equivalent to the initial production line can be obtained, where $2^k \le \mathrm{n}$ and $2^{k+1} \ge \mathrm{n}$. Therefore, the availability of the initial production line can be achieved by calculating the availability of the subsystem $\left\{M_1^k, B_{2k}, M_2^k\right\}$ with Equations 6 to 19.The steady-state availability of the production line can be written as

$$A = f(S_1, S_2, \cdots\cdots, S_{n-1}) \tag{20}$$

4 | OPTIMIZATION METHOD

As mentioned in Section 1, an efficient optimization tool should be presented to find the optimal solution. Therefore, in this section, we provide the details of a hybrid PSO-EDA optimization method for the BAP.

4.1 | Encoding

Every individual of the population denotes a solution of the BAP, which contains information about the capacity of each intermediate buffer. Each individual is characterized by its position and velocity. The individuals are described as follows:

$$X = [X_1,\ X_2, \cdots, X_K],\ X_i = \left[x_i^1, x_i^2, \cdots, x_i^N\right] \tag{21}$$

$$x_i^j \in \{0,\ 1\}\ \ (i = 1,\ 2, \cdots, K; j = 1,\ 2, \cdots, N) \tag{22}$$

$$V = [V_1,\ V_2,\ \cdots,\ V_K],\ V_i = \left[v_i^1,\ v_i^2, \cdots,\ v_i^N\right] \tag{23}$$

$$0 \le v_i^j \le 1 (i = 1,\ 2, \cdots, K; j = 1,\ 2, \cdots, N) \tag{24}$$

where K denotes the dimension of each individual; N is the number of individuals; x_i^j is the position value of the ith dimension of the jth individual; v_i^j represents the probability of x_i^j taking the value 0.As illustrated in Figure 8, we use binary encoding for the individuals. The part of individual $\{x_{(i-1)k+1}, x_{(i-1)k+2}, \cdots, x_{ik}\}$ represents the capacity S_i of buffer B_i. Then the capacity S_i is described as follows:

$$S_i = 2^{k-1}x_{(i-1)k+1} + 2^{k-2}x_{(i-1)k+2} + \cdots + 2^0 x_{\text{ik}} \tag{25}$$

$$S_i \geq 0, \sum_{i=1}^{n-1} S_i = Q, (n-1)k = K \tag{26}$$

where k is the dimension of the part of individual.

4.2 | Initialization

To guarantee the diversity of the initial population, we use an initial random population of N individuals, which is uniformly distributed.

4.3 | Fitness calculation

In our study, the availability of the production line is chosen as the fitness, which is used to evaluate the performance of solutions.

4.4 | Population division

The population is randomly divided into 2 subpopulations *A* and *B*, in which the number of individuals is NA and N - NA, respectively. Then new offspring of subpopulation *A* and *B* are generated according to the probabilistic model of EDA and the individual flying model of PSO, respectively.

4.5 | Probabilistic model

The performance of the EDA is closely related to the probabilistic model.[25] In this paper, the probabilistic model is designed as a probability matrix *P* with the dimension of 2 × K.The best MP individuals in the subpopulation *A* are defined as a superior subpopulation. p_{mi}(gene) represents the element in the matrix *P* at generation gene, which is the probability when the ith dimension value of the individual is m, where m = 1 , 0; i = 1 , 2 , ⋯ , K. Then the matrix *P* is initialized according to the following equation:

$$p_{\text{mi}}(1) = \frac{1}{\text{MP}} \sum_{s=1}^{\text{MP}} I_{\text{mi}^{(s)}}(1), \forall m, i \in K \tag{27}$$

where $I_{\text{mi}^{(s)}}(\text{gene})$ is an indicator function of the *s*th individual in the superior subpopulation, shown as

$$I_{\text{mi}^{(s)}}(\text{gene}) = \begin{cases} 1, & \text{if the ith dimension of sth individual is } m \\ 0, & \text{else} \end{cases} \tag{28}$$

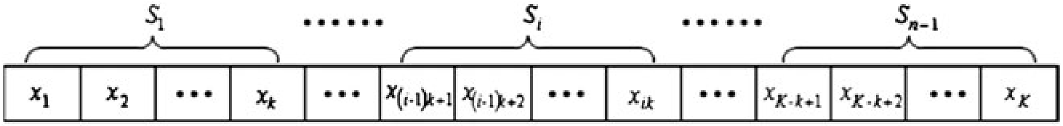

FIGURE 8 Representation of a feasible solution of the BAP

In each generation of the EDA, new individuals are generated via sampling the solution space according to the probability matrix *P*. For an individual, theith dimension value is selected with a probability of $\mathrm{p}_{\mathrm{mi}}(\text{gene})$. In the EDA, a population with NA individuals is generated.

4.6 | Update of probabilistic model

The subpopulation *A* with NA individuals determines the superior subpopulation that consists of the best MP solutions. Then the probability matrix *P* is updated according to the following equation:

$$p^{A}_{\mathrm{mi}}(\text{gene}+1) = (1-\gamma)p^{A}_{\mathrm{mi}}(\text{gene}) + \frac{\gamma}{\mathrm{MP}}\cdot\sum_{s=1}^{\mathrm{MP}} I_{\mathrm{mi}^{(s)}}(\text{gene}+1) \tag{29}$$

where $\gamma(0<\gamma<1)$ is the learning rate of *P*.

4.7 | Update of individual flying model

The velocity and position of each individual in the subpopulation *B* are updated according to the following equations:

$$v^{\text{gene}}_{\text{pbest},i} = \alpha \mathrm{x}^{\text{gene}}_{\text{pbest},i} + \beta\left(1-x^{\text{gene}}_{\text{pbest},i}\right) \tag{30}$$

$$v^{\text{gene}}_{\text{gbest}} = \alpha \mathrm{x}^{\text{gene}}_{\text{gbest}} + \beta\left(1-x^{\text{gene}}_{\text{gbest}}\right) \tag{31}$$

$$v^{\text{gene}+1}_{i} = \omega \mathrm{v}^{\text{gene}}_{i} + c_1 v^{\text{gene}}_{\text{pbest},i} + c_2 v^{\text{gene}}_{\text{gbest}} \tag{32}$$

$$x^{\text{gene}+1}_{i} = \begin{cases} 1, \ \text{rand}(\ )>v^{\text{gene}+1}_{i} \\ 0, \ \text{rand}(\ )\le v^{\text{gene}+1}_{i} \end{cases} \tag{33}$$

where rand( ) is a random function, whose value is in the range of [0, 1]; ω is the inertia weight, a parameter tuning the influence of the velocity of an individual at the last iteration; $\alpha, \beta(0<\alpha,\beta<1)$ are two uniformly distributed random numbers.

4.8 | Acceptance probabilities

To improve the local exploitation capability of the PSO-EDA, an acceptance probability of simulated annealing for updating the personal best of each individual is introduced. After a new individual is generated, firstly, calculate the objective values of old individual S and new individual $\mathrm{S}^{'}$. If corresponding availability differential $\Delta\mathrm{A}=\mathrm{f}(\mathrm{S}^{'})-\mathrm{f}(\mathrm{S})$ is nonnegative, accept the new individual as the personal best individual; otherwise, accept it with the probability of exp(-ΔA/T), where T is a global time-varying parameter called the temperature.

4.9 | Procedure of the PSO-EDA

The proposed PSO-EDA can be formally described as follows:

1. Initialize the population following the procedures in Section 4.2;
2. Evaluate the fitness value according to the procedures in Section 4.3 and obtain the best feasible solution according to the procedures in Section 4.8;
3. Divide the population into 2 subpopulations *A* and *B* according to the procedures in Section 4.4;
4. Select superior subpopulation for subpopulation *A* to build the probabilistic matrix *P* and sample the probabilistic model to generate new NA individuals according to the procedures in Section 4.5;
5. Update the probabilistic model according to the procedures in Section 4.6
6. Generate new N‐NA individuals and update individual flying model according to the procedures in Section 4.7; and
7. If the termination criterion is not met, go to step 2; otherwise, stop.

TABLE 1 Machines parameters

Machine	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$MTBF = 1/\lambda_i$	20	20	30	22	30	30	20	25	26	30	10	30	30	20	25
$MTTR = 1/\mu_i$	7	10	7	5	5	8	8	9	10	6	3	15	14	9	5
Machine	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
$MTBF = 1/\lambda_i$	45	10	20	12	25	20	25	25	40	30	20	30	22	22	25
$MTTR = 1/\mu_i$	10	4	4	3	7	6	6	8	9	10	7	7	5	10	9

5 | NUMERICAL RESULTS

To verify the effectiveness of the proposed evaluation method and the PSO-EDA–based optimization method, experiments are conducted with serial production line configurations of different total buffer capacities and machine sizes. The experiments

TABLE 2 Results of the availability and CPU time by different evaluation methods and optimization methods

Number of Machines	Total Buffer Capacity	Evaluation Method	Optimization Method	A	CPU Time, s
5	60	DDX	Degraded ceiling	0.615094	93.24
			PSO-EDA	0.616059	10.13
		AMM	PSO	0.674368	19.45
			PSO-EDA	0.674605	5.02
			EDA	0.674112	12.64
			GA	0.674101	15.03
10	120	DDX	Degraded ceiling	0.582082	199.74
			PSO-EDA	0.582231	48.92
		AMM	PSO	0.596011	79.00
			PSO-EDA	0.609398	20.01
			EDA	0.598564	36.64
			GA	0.597874	45.72
15	180	DDX	Degraded ceiling	0.537206	733.99
			PSO-EDA	0.542559	32.61
		AMM	PSO	0.567426	73.27
			PSO-EDA	0.569336	33.91
			EDA	0.568748	63.28
			GA	0.567011	56.59
20	240	DDX	Degraded ceiling	0.545536	1562.4
			PSO-EDA	0.552381	47.11
		AMM	PSO	0.560548	89.34
			PSO-EDA	0.562938	43.55
			EDA	0.561341	72.64
			GA	0.560985	79.67
25	300	DDX	Degraded ceiling	0.545814	2051.2
			PSO-EDA	0.561418	84.69
		AMM	PSO	0.560220	209.79
			PSO-EDA	0.571024	70.28
			EDA	0.563972	94.94
			GA	0.562943	111.82
30	360	DDX	Degraded ceiling	0.539524	2292.9
			PSO-EDA	0.555844	129.77
		AMM	PSO	0.552579	214.73
			PSO-EDA	0.562435	105.63
			EDA	0.556252	108.25
			GA	0.555168	152.47

results of the proposed evaluation method is compared with that of DDX algorithm proposed by Dallery et al,[10] while the results of the optimization method is compared with those of degrading ceiling algorithm presented by Nahas et al,[17] GA proposed by Amiri and Mohtashami,[13] and other methods (ie, PSO and EDA). In all the tests, it is assumed that machines are subject to breakdown, and processing rate for each machine is one time unit. The mean time to repair and the mean time between failures are denoted by MTTR and MTBF, respectively (see Table 1). All the experimental studies are programmed in C++ language and run on a PC with Inter (R) Core (TM) 2-Duo (2.00 GHz) CPU under Windows7 operating system.

5.1 | Comparison results of different evaluation methods and optimization methods

Table 2 presents the availability and CPU time obtained using combinations of varied evaluation methods and optimization methods with n machines ($5 \leq n \leq 30$). It is shown that for the same evaluation method the PSO-EDA results in slightly larger availability and requires less time to find the near optimal solutions compared with other optimization methods in all of the cases. Moreover, AMM achieves better results than DDX algorithm with PSO-EDA method.

5.2 | Results of experiments in a given number of machines

Section 5.1 shows that the PSO-EDA method results in higher solution quality and shorter CPU time. Therefore, we use PSO-EDA as the optimization method in the search of the near optimal solutions. This section presents the search results of the 2 evaluation methods (ie, DDX and AMM) in 5-, 15-, and 30-machine production lines. Table 3 presents the results of the buffer allocation solutions, availability, and CPU time for a 5-machine production line with the total buffer capacity of 20, 25, 30, 35, 40, 45, and 50. Figure 9 shows the availability (Figure 9A) and CPU time (Figure 9B) of the near optimal solutions for a 15-machine production line with the total buffer capacity of 60, 75, 90, 105, 120, 135, and 150. Figure 10 shows the availability (Figure 10A) and CPU time (Figure 10B) of the near optimal solutions for a 30-machine production line with the total buffer capacity of 120, 150, 180, 210, 240, 270, and 300.

It is shown that the availability and CPU time of near optimal solutions increase as the total buffer capacity increases for both of the DDX and AMM evaluation methods. In addition, the proposed AMM method results in higher solution quality and less CPU time.

5.3 | Results of experiments with a given total buffer capacity

Experiments are conducted to examine the relationship between the availability and the number of machines with a given total buffer capacity. Table 4 presents the results of the buffer allocation solutions, availability, and CPU time for small-size production lines with the total buffer capacity of 60 and with the number of machines of 5, 6, 7, 8, and 9. Figures 11 and 12 show the

TABLE 3 Results of the buffer allocation solutions, availability and CPU time in a 5-machine production line

Total Buffer Capacity	Evaluation Method	*A*	CPU Time, s	B1	B2	B3	B4
20	DDX	0.439504	3.3	3	9	7	1
	AMM	0.549352	0.68	2	4	4	10
25	DDX	0.456600	3.4	6	7	7	5
	AMM	0.578592	0.71	4	3	8	10
30	DDX	0.471440	3.5	7	11	9	3
	AMM	0.621042	0.98	3	2	10	15
35	DDX	0.484452	4.2	8	12	11	4
	AMM	0.629564	1.04	6	5	12	12
40	DDX	0.496026	4.3	10	13	12	5
	AMM	0.656074	1.24	5	7	13	15
45	DDX	0.506294	4.4	11	15	13	6
	AMM	0.673770	1.84	6	8	14	17
50	DDX	0.515547	4.6	13	16	14	7
	AMM	0.690058	2.24	7	7	16	20

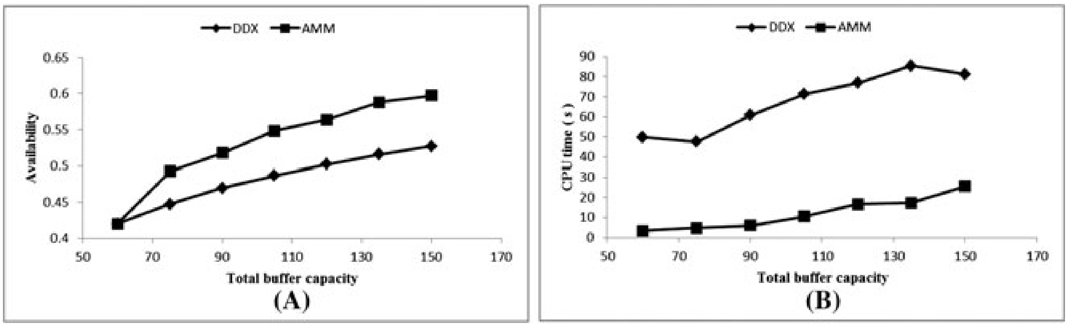

FIGURE 9 Availability and CPU time for a 15-machine production line

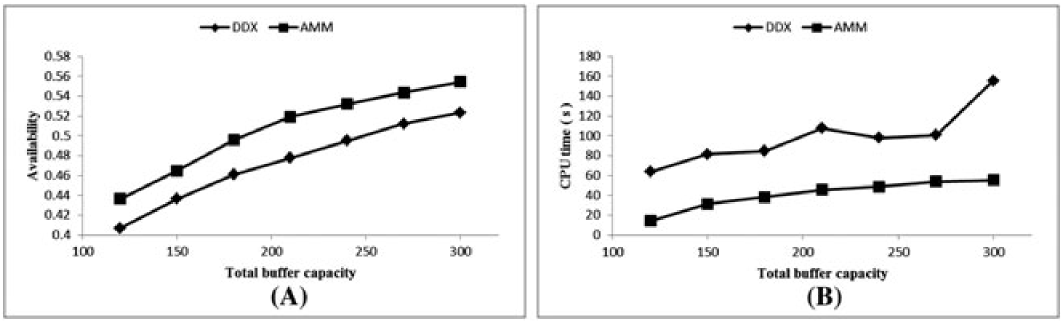

FIGURE 10 Availability and CPU time for a 30-machine production line

TABLE 4 Results of experiments for small-size problems with a total buffer capacity of 60

Number of machines	Evaluation method	*A*	CPU time(s)	B1	B2	B3	B4	B5	B6	B7	B8
n = 5	DDX	0.616059	10.13	24	22	11	3				
	AMM	0.674605	5.02	13	9	21	17				
n = 6	DDX	0.600970	15.11	18	19	11	8	4			
	AMM	0.669207	7.24	6	9	7	26	12			
n = 7	DDX	0.576695	25.78	12	15	10	8	8	7		
	AMM	0.529159	7.46	5	9	5	12	16	13		
n = 8	DDX	0.549558	31.25	8	11	9	8	8	9	7	
	AMM	0.502540	7.94	4	4	5	15	4	13	15	
n = 9	DDX	0.521154	42.09	5	9	8	6	7	9	10	6
	AMM	0.439967	8.74	5	4	5	7	10	4	12	13

availability (Figures 11A and 12A) and CPU time (Figures 11B and 12B) of the near optimal solutions for medium- and large-sized production lines with the total buffer capacity of 120 and 240, respectively.

It shows that the increase of the number of machines leads to a decrease of the availability of near optimal solutions and an increase of the CPU time for near optimal buffer solutions. In addition, the proposed AMM method yields higher solution quality and less CPU time.

5.4 | Results of simulation

This section presents simulation results of throughput and work-in-process obtained using the DDX and AMM methods for a serial production line with a given total buffer capacity of 60 (see Table 5). The simulation is conducted with ARENA simulation software. It shows that the buffer allocation solutions obtained by AMM method is better than the solutions found by DDX method because of less CW. Furthermore, the proposed AMM results in slightly larger throughput than DDX method.

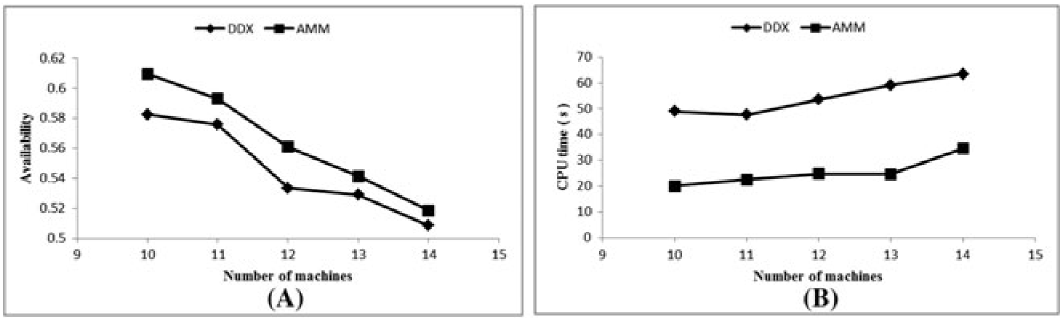

FIGURE 11 Availability and CPU time for medium-size problems with a total buffer capacity of 120

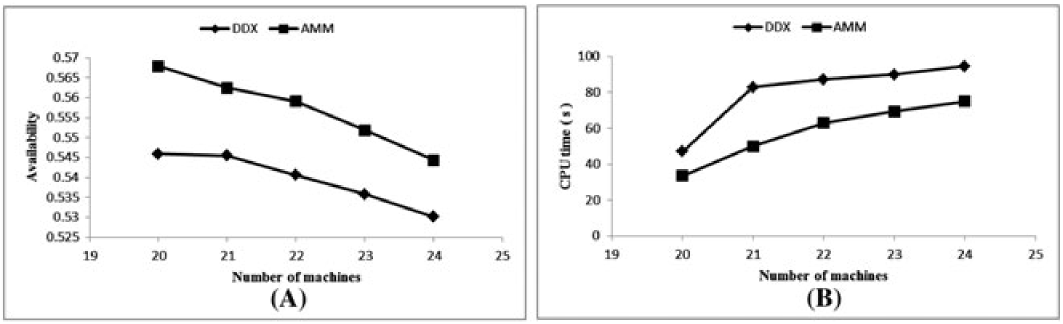

FIGURE 12 Availability and CPU time for large-sized problems with a total buffer capacity of 240

TABLE 5 Simulation results for small-size problems with a total buffer capacity of 60

Number of Machines	Evaluation Method	E	as_1	as_2	as_3	as_4	as_5	as_6	as_7	as_8	CW
n = 5	DDX	0.553	12.81	5.2	1.88	0.39					20.3
	AMM	0.620	5.75	2.36	2.67	1.57					12.4
n = 6	DDX	0.548	9.27	3.62	1.43	2.13	0.85				17.3
	AMM	0.610	3.03	1.56	1.07	1.12	1.8				8.58
n = 7	DDX	0.505	5.14	2.95	2.04	1.02	1.18	2.43			14.8
	AMM	0.525	2.35	0.94	0.72	1.07	1.92	2.38			9.38
n = 8	DDX	0.460	3.69	2.4	1.56	0.98	1.65	2.21	1.48		14.0
	AMM	0.485	1.34	0.81	0.67	0.91	0.83	1.92	1.79		8.27
n = 9	DDX	0.410	2.22	1.78	1.52	0.84	0.89	1.35	1.65	1.23	11.5
	AMM	0.438	1.9	0.7	0.84	0.86	0.67	0.72	1.02	1.65	8.36

6 | CONCLUSION

This paper presents a new evaluation criterion on the basis of the system availability, which makes it possible to focus on a novel optimization target in the classical BAP. Moreover, an evaluation method on the basis of Markov theory (AMM) and an optimization method that combines PSO and EDA are also proposed respectively. Numerical results show good accuracy and efficiency of our methods and highlight the significance of the new evaluation criteria. Meanwhile, simulation is conducted to verify the effectiveness of the buffer configurations achieved by our methods.

Future research extension of this work may be the adaptation of the proposed methods to the series-parallel production lines. Ouazene et al[32] proposed a mathematical formulation to calculate the system throughput of a buffered parallel unreliable machines production line in an operation-dependent–failure context. Our study can be extended to evaluate the system availability and allocate buffers properly under such complex context. What is more, the integrated product quality analysis of production lines with sampling inspection stations and on-line rework operations is of great interest to us.

ACKNOWLEDGEMENTS

This study was supported by the National Natural Science Foundation of China under Grant No. 71471135.

REFERENCES

1. Demir L, Tunalı S, Eliiyi DT. An adaptive tabu search approach for buffer allocation problem in unreliable non-homogenous production lines. *Computers & Operations Research*. 2012;39(7):1477-1486.
2. Köse SY, Demir L, Tunalı S, Eliiyi DT. Capacity improvement using simulation optimization approaches: a case study in the thermotechnology industry. *Engineering Optimization*. 2015;47(2):149-164.
3. Demir L, Tunali S, Eliiyi DT. The state of the art on buffer allocation problem: a comprehensive survey. *J Intell Manuf*. 2014;25(3):371-392.
4. Buzacott JA. Automatic transfer lines with buffer stocks. *International Journal of Production Research*. 1967;5(3):182-200.
5. Kokangul A, Khaniyev T, Cochran JK. Optimal control of work-in-process inventory of a two-station production line. *Optimal Control Applications and Methods*. 2010;31(3):201-211.
6. Ouazene, Y., Chehade, H., Yalaoui, A. and Yalaoui, F. Equivalent machine method for approximate evaluation of buffered unreliable production lines. *Computational Intelligence In Production And Logistics Systems (CIPLS), 2013 IEEE Workshop on. IEEE* 2013; 33–39.
7. Kolb O, Göttlich S. A continuous buffer allocation model using stochastic processes. *European Journal of Operational Research*. 2015;242(3):865-874.
8. Li J, Blumenfeld DE, Huang N, Alden JM. Throughput analysis of production systems: recent advances and future topics. *International Journal of Production Research*. 2009;47(14):3823-3851.
9. Gershwin SB. An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Oper Res*. 1987;35(2):291-305.
10. Dallery Y, David R, Xie XL. Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Transactions on Automatic Control*. 1989;34(9):943-953.
11. De Koster MBM. Estimation of line efficiency by aggregation†. *International Journal of Production Research*. 1987;25(4):615-625.
12. Sörensen K, Janssens G. Simulation results on buffer allocation in a continuous flow transfer line with three unreliable machines. *Advances in Production Engineering & Management*. 2011;6(1):15-26.
13. Amiri M, Mohtashami A. Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm. *Int J Adv Manuf Technol*. 2012;62(1–4):371-383.
14. Xie W, Liao H, Jin T. Maximizing system availability through joint decision on component redundancy and spares inventory. *European Journal of Operational Research*. 2014;237(1):164-176.
15. Diamantidis AC, Papadopoulos CT. A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines. *Math Probl Eng*. 2004;2004(3):209-223.
16. Yuzukirmizi M, Smith JMG. Optimal buffer allocation in finite closed networks with multiple servers. *Computers & Operations Research*. 2008;35(8):2579-2598.
17. Nahas N, Ait-Kadi D, Nourelfath M. A new approach for buffer allocation in unreliable production lines. *International journal of production economics*. 2006;103(2):873-881.
18. Cruz FRB, Duarte AR, Van Woensel T. Buffer allocation in general single-server queueing network. *Computers & Operations Research*. 2008;35(11):3581-3598.
19. Shi C, Gershwin SB. An efficient buffer design algorithm for production line profit maximization. *International Journal of Production Economics*. 2009;122(2):725-740.
20. Demir L, Tunalı S, Løkketangen A. A tabu search approach for buffer allocation in production lines with unreliable machines. *Engineering Optimization*. 2011;43(2):213-231.
21. Demir L, Tunalı S, Eliiyi DT, Løkketangen A. Two approaches for solving the buffer allocation problem in unreliable production lines. *Computers & Operations Research*. 2013;40(10):2556-2563.
22. Narasimhamu KL, Reddy VV, Rao CSP. Optimal buffer allocation in tandem closed queuing network with single server using PSO. *Procedia Materials Science*. 2014;5:2084-2089. https://doi.org/10.1016/j.mspro.2014.07.543
23. Nahas N, Nourelfath M, Ait-Kadi D. Selecting machines and buffers in unreliable series-parallel production lines. *International Journal of Production Research*. 2009;47(14):3741-3774.
24. Shi L, Men S. Optimal buffer allocation in production lines. *IIE Transactions*. 2003;35(1):1-10.
25. Dolgui A, Eremeev AV, Sigaev VS. HBBA: hybrid algorithm for buffer allocation in tandem production lines. *J Intell Manuf*. 2007;18(3):411-420.
26. Costa A, Alfieri A, Matta A, Fichera S. A parallel tabu search for solving the primal buffer allocation problem in serial production systems. *Computers & Operations Research*. 2015;64:97-112. https://doi.org/10.1016/j.cor.2015.05.013

27. Massim Y, Yalaoui F, Amodeo L, Châtelet É, Zeblah A. Efficient combined immune-decomposition algorithm for optimal buffer allocation in production lines for throughput and profit maximization. *Computers & Operations Research*. 2010;37(4):611-620.

28. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*. Perth, Australia: Piscataway; 1995:1942-1948.

29. Ahn CW, An J, Yoo JC. Estimation of particle swarm distribution algorithms: combining the benefits of PSO and EDAs. *Inform Sci*. 2012;192:109-119. https://doi.org/10.1016/j.ins.2010.07.014

30. Lozano JA. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Heidelberg: Springer; 2006.

31. Liu H, Gao L, Pan Q. A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flow shop scheduling problem. *Expert Systems with Applications*. 2011;38(4):4348-4360.

32. Ouazene, Y., Yalaoui, A., Chehade, H., Yalaoui, F. A new formulation of a buffered two-workstation production line with parallel unreliable machines. ISC'2012, 10th Industrial Simulation Conference 2012; 200–206.

How to cite this article: Zhou B, Liu Y, Yu J, Tao D. Optimization of buffer allocation in unreliable production lines based on availability evaluation. *Optim Control Appl Meth*. 2018;39:204–219. https://doi.org/10.1002/oca.2341