IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 4, 2022, accepted May 31, 2022, date of publication June 8, 2022, date of current version June 13, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3181134*

# A Bottleneck Detection-Based Tabu Search Algorithm for the Buffer Allocation Problem in Manufacturing Systems

**SIXIAO GAO**
School of Traffic and Transportation Engineering, Central South University, Changsha, Hunan 410075, China
e-mail: gaosixiao@csu.edu.cn

This work was supported in part by the International Postdoctoral Exchange Fellowship Program (Talent-Introduction Program) under Grant YJ20210201, and in part by the High Performance Computing Center of Central South University.

**ABSTRACT** Buffer allocation is an important research issue in the design and optimization of manufacturing systems. One objective of buffer allocation is to maximize the throughput subject to the total buffer capacity in manufacturing systems. Previous studies have proposed many approaches to solve the buffer allocation problem, and tabu search has been proven to be effective in obtaining a near-optimal solution. To further improve the computational efficiency, this study proposes a bottleneck detection-based tabu search algorithm to efficiently solve the buffer allocation problem of manufacturing systems with sufficient solution quality. In particular, the tabu search algorithm integrated with diversification strategies is proposed to maintain satisfactory solution quality of the buffer allocation. The bottleneck detection cooperates with the variable neighborhood structure to search for proper locations for allocating buffers, decreasing the computational time of the buffer allocation. In addition, an existing throughput evaluation method is integrated with the proposed approach to evaluate neighborhood solutions by calculating the manufacturing system throughput. Numerical examples show that the proposed approach can obtain the near-optimal solution more efficiently than the simple tabu search and adaptive tabu search algorithms. This study improves the computational efficiency of buffer allocation in manufacturing systems without losing solution quality, contributing to efficient resource reconfiguration and process management.

**INDEX TERMS** Buffer allocation, tabu search, bottleneck detection, computational efficiency.

## I. INTRODUCTION

The buffer allocation problem (BAP) is one of the most challenging problems in the design and optimization of manufacturing systems. One objective of buffer allocation is to maximize manufacturing system performance subject to the total buffer capacity because buffers provide temporary storage areas for parts and maintain their flow through machines without blocking phenomena. However, reasonable buffer allocation in a manufacturing system is difficult because excessive buffer capacity causes system redundancy, while insufficient buffer capacity leads to blocking.

Generative methods and evaluative methods are used cooperatively to solve the BAP [1]. Generative methods search for candidate buffer allocation solutions, whereas evaluative methods evaluate the system performance corresponding to candidate solutions. This process continues until the stopping criteria are satisfied.

Commonly used evaluative methods consist of the exact method [2], [3], decomposition methods [4]–[6], aggregation methods[7], [8], and simulation methods [9], [10]. The exact method is only applicable to small-scale problems because of its computational complexity. Simulation methods can obtain system performance with high accuracy; however, long setting and simulation times limit their application in solving the BAP, which requires a large number of evaluations for candidate solutions. Decomposition and aggregation methods can quickly approximate the system performance of a manufacturing system with sufficient accuracy. Therefore, they are widely used in integration with generative methods to solve the BAP.

The associate editor coordinating the review of this manuscript and approving it for publication was Ugur Guvenc.

Commonly used generative methods are divided into four categories, including enumeration methods, dynamic programming, search algorithms, and metaheuristics [11]. Enumeration methods search all feasible solutions to obtain the optimal buffer allocation solution [12]. Wang *et al.* [13] proposed a complete enumeration method to search for the buffer allocation solution. Although their methods can obtain the theoretical optimal solution, a long computational time is typically needed, which limits its application for a large-scale manufacturing system. Dynamic programming can obtain the buffer allocation solution with a high solution quality by decomposing the BAP into subproblems [14]. Previous studies [12], [14] presented the advantage of dynamic programming to solve the BAP. However, similar to enumeration methods, dynamic programming is usually applied to a manufacturing system with a small number of stations because of its long computational time for large-scale problems. Search algorithms obtain a near-optimal solution by searching for neighborhood solutions according to guidance information [15]–[17]. The different guidance information, such as bottlenecks, determines the effectiveness of search algorithms. Cruze *et al.* [18] used the blocking probability to search for bottleneck locations for buffer allocation. Experimental results demonstrated the effectiveness of the search algorithm based on the blocking probability. Gao *et al.* [19], [20] presented a bottleneck indicator for guidance buffer allocation and used the variable neighborhood search algorithm to decrease the computational time. Their algorithms quickly obtain near-optimal solutions; however, their solution quality needs further improvement. Metaheuristic methods typically exploit the search space using learning or evolutionary strategies and have been widely used in recent years [21], [22]. Zhou *et al.* [23] presented a particle swarm optimization algorithm to maximize the availability of a manufacturing system by allocating buffers. Koyuncuoglu and Demir [24] proposed two population-based algorithms to solve the BAP and compared their solution quality. Generally, sufficiently good buffer allocation solutions can be obtained by metaheuristics; however, their solution quality and computational efficiency may change violently for different problems. In addition to the four kinds of commonly used algorithms, a new trend to solve the BAP is to integrate one algorithm with another one or other strategies to intensify the search process and obtain better solution quality. Kose and Kilincci [25] proposed a hybrid approach of a genetic algorithm and simulated annealing to improve the buffer allocation solutions of an open manufacturing system. Weiss, Matta, and Stolletz [26] integrated a rule-based local search algorithm with the new individual lower bounds to allocate buffers into a manufacturing system with limited supply.

The tabu search algorithm (TS) is a metaheuristic and is widely used in solving optimization problems [27], [28]. Previous studies [29]–[33] applied the TS to optimize manufacturing system resources and verified the effectiveness of the TS in allocating buffers into manufacturing systems. Furthermore, previous studies [34]–[36] have attempted to integrate the TS with other algorithms to improve the solution quality of buffer allocation. Numerical examples demonstrated the validity of their algorithms. Although previous studies have verified the effectiveness of the TS in solving BAPs of manufacturing systems, the computational time of the TS was unsatisfactory[32]. Therefore, a bottleneck detection (BD)-based TS (BD-TS) is proposed to efficiently solve the buffer allocation problem of a manufacturing system with sufficient solution quality. The objective of buffer allocation is to maximize the throughput of the manufacturing system subject to the total buffer capacity. The TS continues to search for candidate neighborhood solutions to obtain the near-optimal solution of the buffer allocation to maximize the throughput. Diversification strategies are proposed to maintain the solution quality of the TS. The BD, in cooperation with the variable neighborhood structure, is proposed to guide effective searching for candidate neighborhood solutions and to decrease the computational time to obtain the near-optimal solution. The performance of the BD-TS is investigated by comparing it to the simple tabu search (STS) and adaptive tabu search (ATS) based on numerical examples.

The contribution of the study is that we improve the computational efficiency of the BAP without losing solution quality, which facilitates the efficient design and optimization of manufacturing systems in real applications.

The remainder of this paper is structured as follows. Section 2 presents the problem statement. The methodology is introduced in Section 3. Section 4 presents the experimental design and comparison algorithms. Numerical results are presented to test the effectiveness of the proposed approach in Section 5. Section 6 provides some conclusions and future work.

## II. PROBLEM STATEMENT

### A. ASSUMPTIONS

A manufacturing system that consists of $I$ stations in the study is modeled as a stochastic model in Fig. 1. Each station has a machine and a buffer. The assumptions in the study are as follows:

- Materials in a manufacturing system are discrete jobs.
- External jobs enter a manufacturing system at an arrival rate according to a Poisson process.
- Machines have unique performance that is denoted by service rates.
- Buffers have finite capacity. The capacity consists of the storage area of the buffer and machine. Because there is at least one storage area in the machine and no buffer is typically set in the first machine in a manufacturing system, the first buffer capacity is always one, representing the storage area for the first machine.
- Blocking after service (BAS) [37] is utilized to describe job blocking behavior in the study. Here, the BAS is defined as follows: the part upon completion of service in a station cannot enter its subsequent station when no available space exists in its subsequent station. If the space in its subsequent station becomes

available, the part in the station enters its subsequent station.

### B. NOTATIONS

To help understand the study contents clearly, some notations in the proposed BD-TS are as follows:

- $\lambda_i$ denotes the arrival rate at which jobs enter a manufacturing system from station $i$ per unit time. $a_i$ denotes the effective arrival rate.
- $\mu_i$ denotes the service rate at which jobs leave machine $i$ per unit time.
- $i$ represents a station in a manufacturing system. $I$ denotes the number of stations.
- $N$ denotes the total buffer capability in a manufacturing system.
- $K_i$ denotes the maximum capability of buffer $i$.
- $\theta_i$ denotes the throughput at which jobs leave a manufacturing system from station $i$ per unit time.
- $\theta_{INITIAL}$ denotes the throughput after initialization.
- $\theta_{BEST}$ denotes the final best throughput.
- $\theta_{MAX}$ denotes the current maximum throughput after the solution update.
- $\bar{b}$ denotes a solution vector of the buffer allocation, and $\bar{b} = (b_2, b_3, \cdots, b_I)$.
- $b_i$ denotes the allocated buffer capacity in buffer $i$.
- $\bar{b}_{BEST}$ denotes the final best solution vector.
- $\bar{b}_{CURRENT}$ denotes the current best solution vector.
- $\bar{b}_{TABU}$ denotes the tabu solution vector.
- $\Pi$ denotes the set that stores neighborhood solution vectors.
- $\bar{p}$ denotes a configuration vector of blocking probability at each station, and $\bar{p} = (p_{b_2}, p_{b_3}, \cdots, p_{b_I})$.
- $p_{b_i}$ denotes the blocking probability in the case of $b_i$ buffer capacity allocated in buffer $i$.
- $\bar{p}_{CURRENT}$ denotes the current best configuration vector of blocking probabilities.
- $\bar{p}_{BEST}$ denotes the final best configuration vector of blocking probabilities.
- $\bar{p}_{TABU}$ denotes the tabu configuration vector of blocking probabilities.
- $j$ denotes an iteration. $J$ is the maximum number of iterations.
- $j_{BEST}$ denotes the convergence iteration number.
- $t$ denotes the computational time.
- $t_{BEST}$ denotes the computational time when the whole buffer allocation process stops.
- $n_{nei}$ denotes the number of neighborhood solutions. $N_{nei}$ is the maximum number of neighborhood solutions.
- $\bar{p}_{SUB1}$ denotes the subconfiguration vector of blocking probability obtained from low blocking probabilities in $\bar{p}_{CURRENT}$.
- $\bar{p}_{SUB2}$ denotes the subconfiguration vector of blocking probability obtained from high blocking probabilities in $\bar{p}_{CURRENT}$.
- $P_A$ denotes the probability that the location where buffer capacity is decreased is selected in a buffer corresponding to a blocking probability in $\bar{p}_{SUB1}$.
- $P_B$ denotes the probability that the location where buffer capacity is increased is selected in a buffer corresponding to a blocking probability in $\bar{p}_{SUB2}$.
- $l$ denotes the tabu list.
- $tt$ denotes the tabu tenure.
- $tt_{CURRENT}$ denotes the number of tabu moves in $l$.

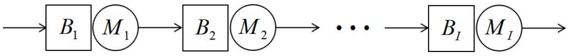

**FIGURE 1. Manufacturing system involving *I* stations.**

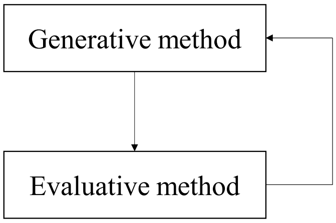

**FIGURE 2. Solution process of the buffer allocation problem.**

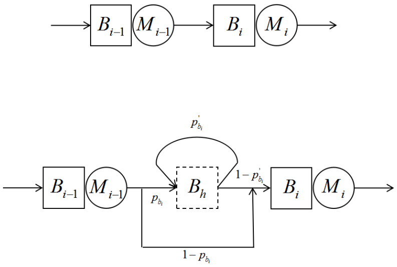

**FIGURE 3. Two stations with BAS.**

### C. DEFINITION OF THE BUFFER ALLOCATION PROBLEM

In this study, the objective of buffer allocation is to maximize the throughput of a manufacturing system subject to the total buffer capacity. The mathematical description is formulated as follows:

$$\begin{aligned} &\text{Find } \bar{b} = (b_2, b_3, \cdots, b_I) \text{ to} \\ &\text{maximize } \theta_I \\ &\text{Subject to } \sum\nolimits_{i=2}^{I} b_i = N \qquad (1) \\ &\qquad\qquad b_i \geq 1 (2 \leq i \leq I) \qquad (2) \\ &\qquad\qquad b_i \text{ is a nonnegative integer} \end{aligned}$$

where $\theta_I$ denotes the throughput of the manufacturing system. Equation (1) shows the constraint for the total buffer capacity. The total buffer capacity is allocated to $I-1$ buffers because $b_1$ is always one according to the assumption. Equation (2) shows the lower bound for each buffer allocation because there is at least one storage space for the machine in a station.

## III. METHODOLOGY

The buffer allocation problem is solved by both generative methods and evaluative methods, as shown in Fig. 2. The generalized expansion method (GEM) was proposed to calculate the throughput of queueing network systems [38]. Previous studies [39], [40] have verified its effectiveness as an evaluative method. In this study, the GEM is utilized as an evaluative method to calculate the throughput of a manufacturing system. The BD-TS is proposed as a generative method to search for the near-optimal solution of the buffer allocation.

### A. THROUGHPUT EVALUATION BY GEM

The GEM can accurately analyze the job flow process through the blocking probabilities of stations in a manufacturing system. Therefore, the GEM is utilized to calculate the throughput of the manufacturing system in the study. The GEM includes the following four steps:

*Step 1:* Network reconfiguration. Fig. 3 shows an approximation model for the BAS between two stations. For a pair of two stations in the manufacturing system, an artificial buffer with infinite capacity is added between the two stations to represent an additional delay due to blocking. Jobs upon completion of service in machine $i-1$ enter artificial buffer $h$ in the case of no available space in buffer $i$, with blocking probability $p_{b_i}$. The jobs enter buffer $i$ successfully with probability $1-p_{b_i}$. Here, $p_{b_i}$ and $1-p_{b_i}$ are used as routing probabilities. The blocked jobs in the artificial buffer $h$ enter buffer $i$ with probability $1-p'_{b_i}$ after a delay. If buffer $i$ is still unavailable, the blocked jobs will remain in the artificial buffer $h$ with reroute probability $p'_{b_i}$ and wait for another delay. This process continues until the space is available in buffer $i$.

*Step 2:* Parameter estimation. This step mainly calculates the parameters $p_{b_i}$, $p'_{b_i}$, and visual service rate of the artificial buffer $\mu_h$ (reciprocal of delay in artificial buffer $h$) using (3), (4), and (5), respectively.

$$p_{b_i} = \frac{(1-\rho)\rho^{K_i}}{1-\rho^{K_i+1}} \tag{3}$$

$$p'_{b_i} = \left[\frac{\mu_{i+1}+\mu_h}{\mu_h} - \frac{\lambda\left(r_2^{K_i}-r_1^{K_i}\right)-(r_2^{K_i-1}-r_1^{K_i-1})}{\mu_h\left(r_2^{K_i+1}-r_1^{K_i+1}\right)-\left(r_2^{K_i}-r_1^{K_i}\right)}\right]^{-1} \tag{4}$$

$$\mu_h = \mu_{i+1} \tag{5}$$

where $\rho$ denotes the utilization of a machine; $K_i$ denotes the maximum capacity of buffer $i$; and $r_1$ and $r_2$ are roots to $\lambda - (\lambda+\mu_h+\mu_{i+1})\,x + \mu_h x^2 = 0$

*Step 3:* Feedback estimation. This step approximates the arrival rate $\lambda_{i+1}$ by solving (3)-(11).

$$\lambda = \lambda_{i+1} - \lambda_h(1-p'_{b_i}) \tag{6}$$

$$\lambda_{i+1} = a_i(1-p_{b_i}) \tag{7}$$

$$z = (\lambda+2\mu_h)^2 - 4\lambda\mu_h \tag{8}$$

$$r_1 = \frac{\left[(\lambda+2\mu_h)-z^{1/2}\right]}{2\mu_h} \tag{9}$$

$$r_2 = \frac{\left[(\lambda+2\mu_h)+z^{1/2}\right]}{2\mu_h} \tag{10}$$

$$\rho = \frac{a_i}{\mu_h} \tag{11}$$

*Step 4:* Throughput calculation. Based on Steps 1, 2, and 3, every pair of machines and their subsequent buffers in the manufacturing system are analyzed until the arrival rate of the final station $\lambda_I$ is obtained. Then, the throughput of the manufacturing system can be calculated by substituting $\lambda_I$, $p_{b_I}$ and $\mu_I$ into (12) and (13).

$$\theta_I = \mu_I \frac{\rho_I(1-\rho_I^{K_I+1})}{(1-\rho_I^{K_I+1})} \tag{12}$$

$$\rho_I = \frac{\lambda_I}{\mu_I} \tag{13}$$

### B. BD-TS

STS typically generates candidate neighborhood solutions by randomly choosing two stations and changing the buffer capacity of the two stations, which may lead to invalid and unstable neighborhood solution generations and affect the computational efficiency of TS. However, the BD-TS detects possible bottleneck stations based on the proposed bottleneck indicator of each station. Then, candidate neighborhood solutions are generated by increasing the buffer capacity in the possible bottleneck stations while decreasing the buffer capacity in the other nonbottleneck stations. By this method, neighborhood solution generation becomes more effective. The BD-TS framework is shown in Fig. 4. The essential steps of the BD-TS, including parameter initialization, bottleneck indicator calculation, neighborhood criterion by the BD, best solution selection by the TS criterion, diversification strategy, and stopping criterion, are presented in detail.

#### 1) PARAMETER INITIALIZATION

All parameters of the BD-TS are initialized first. In particular, an initial solution is generated by randomly allocating the total buffer capacity $N$ into $I-1$ buffers. Here, the solution refers to the solution vector of the buffer allocation for a manufacturing system.

#### 2) BOTTLENECK INDICATOR CALCULATION

The bottleneck is the station with the weakest work performance in a manufacturing system. Because the buffer capacity increment in the bottleneck can improve both its performance and the throughput of the manufacturing system, it is effective to allocate buffer capability based on the bottleneck. To detect possible bottlenecks, the bottleneck indicator is defined by the blocking probability, which shows the degree of blocking at each station in a manufacturing system. Therefore, the higher the blocking probability of a station is, the more likely the station is to be the bottleneck of the manufacturing system. In this study, we suppose that the external arrival rate follows a Poisson distribution, and the service rates of machines follow an exponential distribution. Therefore, according to the literature [41], the blocking probability formulation can be calculated by:

$$p_{b_i} = \frac{(1-\rho_i)\rho_i^{b_i}}{1-\rho_i^{b_i+1}} \tag{14}$$

$$\rho_i = \frac{a_i}{\mu_i} \tag{15}$$

where $\rho_i$ is the utilization of station $i$ and $a_i$ is the effective arrival rate of station $i$. By GEM, the effective arrival rate $a_i$ of each station and the initial throughput $\theta_{INITIAL}$ of a manufacturing system can be obtained. Equations (14) and (15) are used to calculate the blocking probabilities of stations and to update $\bar{p}_{CURRENT}$ and $\bar{p}_{BEST}$. Furthermore, $\theta_{BEST}$ is updated by $\theta_{INITIAL}$.

### 3) NEIGHBORHOOD SOLUTION CREATION BY THE BD

Neighborhood solution creation consists mainly of the selection of the search space and neighborhood structure and the method of move representation. Here, the search space is the range of feasible solutions, and the neighborhood structure includes the selected feasible solutions. Furthermore, move representation generates neighborhood solutions by changing the buffer capacity allocation at two stations in a manufacturing system. The vital design components in the TS, including the search space, variable neighborhood structure, and move representation, are presented in detail.

#### *a: SEARCH SPACE AND VARIABLE NEIGHBORHOOD STRUCTURE*

BD-TS considers all the feasible solutions as the search space in the study. Regarding the neighborhood structure, BD-TS considers a subset of all feasible solutions. Because the variable neighborhood structure is effective in decreasing search time and improving buffer allocation efficiency [28], the number of neighborhood structures is set as

$$n_{nei} = I + tt_{CURRENT} \tag{16}$$

#### *b: MOVE REPRESENTATION*

The move in the BD-TS can be defined by

$$\begin{aligned}&(b_2, \cdots b_v, \cdots, b_w, \cdots, b_I)\\ &\quad\rightarrow (b_2, \cdots b_v - \Delta b, \cdots, b_w + \Delta b, \cdots, b_I)\end{aligned} \tag{17}$$

which means that the $\Delta b$ buffer capacity decreased in buffer $v$ and that the same amount of buffer capacities increased in buffer $w$. Here, $v$ and $w$ denote two different randomly selected buffers. $b_v$ and $b_w$ are the capacities of the two selected buffers. Furthermore, $\Delta b$ is a variable. We define that $\Delta b = 1$ in the regular process. However, if the diversification strategy condition is met,

$$\Delta b = randint(2, \beta) \tag{18}$$

$$\beta = floor(\frac{N}{I-1}) \tag{19}$$

where $randint(2, \beta)$ denotes a random integer in the range of 2 and $\beta$; $floor(\frac{N}{I-1})$ denotes an integer that is no more than $\frac{N}{I-1}$.

A previous study [42] randomly selected move locations, which led to unstable and invalid move representations and solution updates. We select the move locations by the BD. Based on the blocking probabilities of stations, possible bottlenecks of a manufacturing system are detected. Then, neighborhood solutions of the buffer allocation are generated by increasing the buffer capacity in the possible bottlenecks and decreasing the same buffer capacity in the other non-bottleneck locations. Fig. 5 shows the move representation for the BD-TS. The specific process is described as follows:

*Step 1:* Based on the blocking probabilities of stations, $I - 1$ buffers in a manufacturing system are decomposed equally into Group A with the subconfiguration vector of blocking probability $\bar{p}_{SUB1} = \{p_k, \cdots\}$ and Group B with the subconfiguration vector of blocking probability $\bar{p}_{SUB2} = \left\{p_{b_l}, \cdots\right\}$. If $I - 1$ is odd, $I/2$ buffers are in Group A; otherwise, $(I-1)/2$ buffers are in Group A. $p_{b_k}$ in $\bar{p}_{SUB1}$ are lower than $p_{b_l}$ in $\bar{p}_{SUB2}$. The buffers in Group B are possible bottlenecks in the manufacturing system.

*Step 2:* The buffer where capacity will be decreased is decided by selecting the corresponding blocking probability in Group A with probability $P_A$ and in Group B with probability $1 - P_A$. After the buffer is decided in Group A or Group B, the specific buffer is randomly selected in Group A or Group B. Here, a reselection operation is used to satisfy constraint (2). Another buffer will be chosen if the selected buffer capacity is lower than one. The value of $P_A$ is set to 60% and 70% in this study.

*Step 3:* The buffer where the same number of buffer capacities will be increased is decided by selecting the corresponding blocking probability in Group B with probability $P_B$ and in Group A with probability $1-P_B$. After the buffer is decided in Group A or Group B, the specific buffer is selected randomly in Group A or Group B. Here, the reselection operation is used to satisfy that the buffer selected in Step 3 should not be the same as the buffer selected in Step 2. The value of $P_A$ is problem-dependent. The value of $P_B$ is set to 60% and 70% in this study.

### 4) BEST SOLUTION SELECTION BY THE TS CRITERION

By GEM, the throughput and the blocking probability configurations corresponding to all neighborhood solutions for a manufacturing system can be obtained. The maximum throughput among the obtained throughputs is used to update $\theta_{MAX}$. Then, $\theta_{MAX}$ is compared to $\theta_{BEST}$, which updates $\theta_{BEST}$, $\bar{b}_{BEST}$, $\bar{p}_{BEST}$, $\bar{b}_{CURRENT}$, $\bar{p}_{CURRENT}$ and $l$ according to tabu tenure and aspiration criteria.

#### *a: TABU TENURE*

Tabu tenure is the length of the tabu list, which has an important effect on the solution quality of the TS [27]. To improve computational efficiency, the dynamic rule for the tabu tenure may be useful. Therefore, in the study, the tabu tenure can be expressed as

$$tt = j \tag{20}$$

Here, $j$ is the number of iterations.

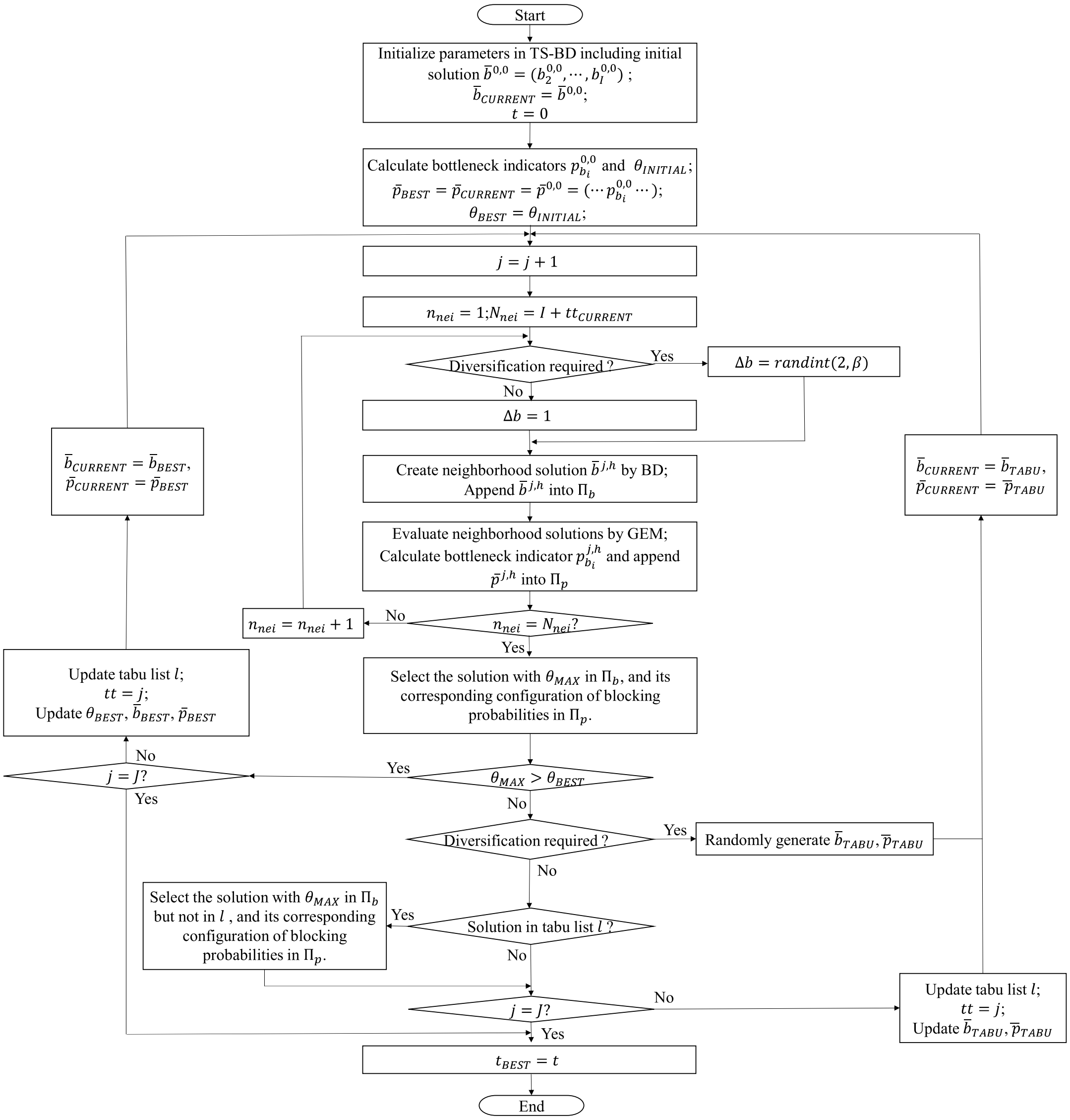
Start
Initialize parameters in TS-BD including initial solution $\bar{b}^{0,0} = (b_2^{0,0}, \cdots, b_I^{0,0})$ ;
$\bar{b}_{CURRENT} = \bar{b}^{0,0}$;
$t = 0$
Calculate bottleneck indicators $p_{b_i}^{0,0}$ and $\theta_{INITIAL}$;
$\bar{p}_{BEST} = \bar{p}_{CURRENT} = \bar{p}^{0,0} = (\cdots p_{b_i}^{0,0} \cdots)$;
$\theta_{BEST} = \theta_{INITIAL}$;
$j = j + 1$
$n_{nei} = 1; N_{nei} = I + tt_{CURRENT}$
Diversification required ?
Yes
$\Delta b = randint(2, \beta)$
No
$\Delta b = 1$
Create neighborhood solution $\bar{b}^{j,h}$ by BD;
Append $\bar{b}^{j,h}$ into $\Pi_b$
Evaluate neighborhood solutions by GEM;
Calculate bottleneck indicator $p_{b_i}^{j,h}$ and append $\bar{p}^{j,h}$ into $\Pi_p$
$\bar{b}_{CURRENT} = \bar{b}_{BEST}$,
$\bar{p}_{CURRENT} = \bar{p}_{BEST}$
$\bar{b}_{CURRENT} = \bar{b}_{TABU}$,
$\bar{p}_{CURRENT} = \bar{p}_{TABU}$
$n_{nei} = n_{nei} + 1$
No
$n_{nei} = N_{nei}$?
Yes
Update tabu list $l$;
$tt = j$;
Update $\theta_{BEST}$, $\bar{b}_{BEST}$, $\bar{p}_{BEST}$
Select the solution with $\theta_{MAX}$ in $\Pi_b$, and its corresponding configuration of blocking probabilities in $\Pi_p$.
No
$j = J$?
Yes
Yes
$\theta_{MAX} > \theta_{BEST}$
No
Diversification required ?
Yes
Randomly generate $\bar{b}_{TABU}$, $\bar{p}_{TABU}$
No
Select the solution with $\theta_{MAX}$ in $\Pi_b$ but not in $l$ , and its corresponding configuration of blocking probabilities in $\Pi_p$.
Yes
Solution in tabu list $l$ ?
No
No
$j = J$?
Yes
Update tabu list $l$;
$tt = j$;
Update $\bar{b}_{TABU}$, $\bar{p}_{TABU}$
$t_{BEST} = t$
End

**FIGURE 4.** Framework of the BD-TS.

### b: ASPIRATION CRITERIA

In the study, we use the aspiration criteria to allow a move even though the move is in the tabu list in the case of updating the best solution.

### 5) DIVERSIFICATION STRATEGY

As indicated by Glover *et al.* [28], the diversification strategy is helpful in the case of "humps" in the neighborhood structure. There are two kinds of diversification strategies used in this study. The first strategy extends the search space by restarting the solution process, and the second strategy helps explore more promising solutions by changing move values. The specific contents of the two strategies are as follows:

- If the best throughput has not been improved within a certain number of iterations, a random restart will be applied. The value of the iteration number is problem dependent and set to one-sixteenth of the maximum number of iterations according to the preliminary tests.
- If the best throughput has not been achieved in a certain number of trials in the neighborhood structure, $\Delta b$ will be changed by (17), (18), and (19). The value of the trials is problem dependent and set to one-fifth of the neighborhood structure size according to the preliminary tests.

### 6) STOPPING CRITERION

Commonly used stopping criteria include a fixed number of iterations or fixed computation time; no better solution is generated after a fixed number of iterations; and an objective throughput is met. The BD-TS stops if the number of iterations reaches a fixed value or no better solutions are

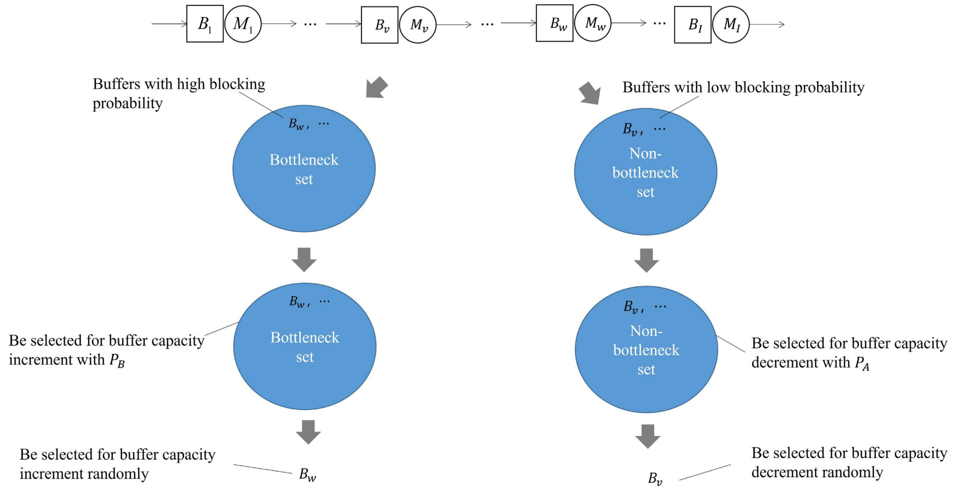
$B_1$
$M_1$
$B_v$
$M_v$
$B_w$
$M_w$
$B_I$
$M_I$
Buffers with high blocking probability
Buffers with low blocking probability
$B_w$, ⋯
Bottleneck set
$B_v$, ⋯
Non-bottleneck set
Be selected for buffer capacity increment with $P_B$
Be selected for buffer capacity decrement with $P_A$
Be selected for buffer capacity increment randomly
$B_w$
$B_v$
Be selected for buffer capacity decrement randomly

**FIGURE 5.** Move representation for the BD-TS.

obtained in a certain number of iterations, which were also used in previous studies [24], [30], [31], [35]. The values of the two parameters are problem-dependent. The first value is set to 20 times the station size in a manufacturing system, while the second value is set to one-eighth of the first value. These numbers are determined by the preliminary tests.

## IV. COMPUTATIONAL EXPERIMENT

To verify its effectiveness, the BD-TS is compared to the STS [31] and ATS [31]. The vital settings of the STS and ATS are as follows:

- The STS does not use any additional strategies or local search methods; meanwhile, the ATS adopts intensification and diversification strategies. The intensification strategy that decreases move values to search for a more promising solution intensively is only used for manufacturing systems with more than 20 stations. The diversification strategy restarts the solution process if no better solutions are generated in a certain number of iterations and penalizes frequently performed moves.
- The number of tabu tenures in the STS is $tt = \sqrt{N_{nei}}$, where $N_{nei}$ is the number of all neighborhood solutions of the current solution. The number of tabu tenures in the ATS is dynamic and is predefined as its maximum and minimum values. If no better solutions are generated in a move representation, the number of tabu tenures increases by one; otherwise, the number of tabu tenures decreases by one. The initial value of $tt$ in the ATS is problem dependent and set to $\sqrt{N_{nei}}$.
- The STS and ATS consider all feasible candidate solutions in the move representation.
- Move representations of the BD-TS, STS, and ATS are the same. However, the move value of the STS is set to one; meanwhile, the move value of the ATS depends on the problem scale (%1 of the total buffer capacity) and can be decreased to one using the intensification strategy.
- To make the BD-TS, STS, and ATS more comparable, the input parameters and stopping criteria of the three algorithms are the same. The input parameters consist of the external arrival rate, service rates of machines, and machine number.
- The GEM is used to calculate the throughput of a manufacturing system and evaluate buffer allocation solutions produced by the BD-TS, STS, and ATS.

Three experiments are included in the numerical examples to test the BD-TS for the buffer allocation problem in manufacturing systems with small, medium, and large scales. Table 1 shows the input parameters of the three experiments.

Each experiment consists of six input patterns, where five replicate calculations are run. The average computational time for the five replicate calculations is used to evaluate the computational efficiency of the BD-TS, STS, and ATS. The parameter for evaluating the computational efficiency improvement is denoted by

$$\varepsilon = \left| \frac{t_{BD-TS} - t_{TS}}{t_{TS}} \right| \tag{21}$$

where $t_{BD-TS}$ denotes the average computational time of the BD-TS for the five replication calculations and $t_{TS}$ denotes the average computational time of the STS or ATS for the five replication calculations.

Since the performance of a metaheuristic may be affected by the choice of initial solutions, two kinds of initialization schemes are used in Experiment 1, including random initialization (RI) and service rate initialization (SRI). The RI generates an initial solution randomly, whereas the SRI allocates larger buffer capacities into the stations with higher service rates.

The order of service rates may also affect the performance of the proposed BD-TS. Therefore, four kinds of the service rate order, including increasing order, decreasing order, average order, and random order, are tested in Experiment 1. Here, the increasing order represents that the service rate increases from the first station to the last one, whereas the decreasing order converse. The average order means all service rates of stations are identical. The random order allocates the service rates of stations randomly.

The proposed BD-TS, STS, and ATS were all written in Python 3.6 and executed for all experiments on a computer with a 3.2 GHz Intel Core 4 Duo central processing unit (CPU).

## V. NUMERICAL RESULTS

### A. SMALL-SCALE MANUFACTURING SYSTEM

#### 1) COMPUTATIONAL EFFICIENCY ANALYSIS

Tables 2 and 3 and Fig. 6 show the results of the BD-TS, STS, and ATS in Experiment 1. Tables 4 and 5 present the results of the BD-TS, STS, and ATS for Pattern A in the case of different initialization schemes and service rate orders. The following conclusions can be drawn:

- The BD-TS can improve the computational efficiency by 20%-70% compared to the STS and ATS, even though $P_A$ and $P_B$ are set to different values, resulting from the move representation conducted by the BD and the variable neighborhood structure. The variable neighborhood structure decreases the selected feasible solutions. The BD detects possible bottlenecks where the buffer capacity will be increased with high probability and guides the TS to more efficiently search for better neighborhood solutions. However, both the STS and ATS use all feasible solutions as the neighborhood structure. Therefore, the BD-TS can obtain the near-optimal solution with less computational time.
- The computational efficiency improvement increases if a higher total buffer capacity is allocated into the manufacturing system. A higher buffer capacity corresponds to a larger search space, which increases the difficulty of searching for better neighborhood solutions. Because of the advantage of allocating buffer capacity reasonably and decreasing the selected feasible solutions, the BD-TS can achieve better computational efficiency.
- The initialization scheme affects the computational efficiency of the BD-TS, TS, and ATS. Compared to the SRI, the BD-TS, TS, and ATS take a longer computational time to obtain the buffer allocation solution in the case of the RI. The RI may generate initial solutions that may be far different from the near-optimal solution, leading to more trials to obtain it. Furthermore, the BD-TS achieves better computational efficiency than the TS and ATS in the cases of the RI and SRI.
- In the cases of different service rate orders, the computational efficiency of the BD-TS is the best in the algorithms tested, although the service rate order can cause fluctuations in the computational efficiency of the BD-TS, TS, and ATS.

#### 2) SOLUTION QUALITY ANALYSIS

- The BD-TS, STS, and ATS achieve the same near-optimal solution, which shows that they have the same solution quality, although the BD-TS achieves much better computational efficiency than the STS and ATS. Based on two diversification strategies, the BD-TS enlarges the change value in the move representation and searches for a further neighborhood solution, whereas its solution process can be restarted if it is trapped in a local optimal solution. Consequently, the solution quality of the BD-TS has been improved, although its neighborhood structure is a subset of all feasible solutions.
- For balanced manufacturing systems whose station service rates are the same, the high throughput is usually generated by allocating a relatively small amount of buffer capacity in the first few stations, a relatively large amount of buffer capacity in the final few stations, and the other amounts of buffer capacity evenly into the other stations. Because the external arrival rate is relatively sufficient, the actual performance of the first few stations is mainly affected by blocking phenomena. However, the subsequent stations are more likely to encounter both starving and blocking phenomena because of the blocking phenomena in the first few stations. Therefore, the subsequent stations require greater buffer capacities. Because the final station is never blocked, it is a good choice to enlarge the throughput by increasing the buffer capacity of the final few stations and temporarily storing more jobs in them. Then, the blocked jobs can enter the machine of the final station directly and rapidly leave the system upon completion of service pending its space availability. Furthermore, the allocated buffer capacities gradually increase from the first station to the final station. This phenomenon becomes more apparent with increasing external arrival rates.
- For unbalanced manufacturing systems whose station service rates are different, both buffer capability and service rates affect the actual performance of a station considering starving and blocking in a manufacturing system. If all stations have the same or balanced actual performance, which means that no bottleneck effects occur in the manufacturing system, the best throughput will typically be generated. Therefore, greater buffer capacities are allocated to stations whose service rates are low to achieve balanced actual performance. However, the final station in an unbalanced manufacturing system also has a relatively high buffer capacity for the same reason as in balanced manufacturing systems

### B. MEDIUM-SCALE MANUFACTURING SYSTEM

#### 1) COMPUTATIONAL EFFICIENCY ANALYSIS

Tables 6 and 7 and Fig. 7 show the results of the BD-TS, STS, and ATS in Experiment 2. The following conclusions can be drawn:

**TABLE 1.** Input parameters of the three experiments.

| Experiments | Items | Parameters |
|---|---|---|
| Small-scale manufacturing system | External arrival rate | $\lambda_1 = 1 \; or \; 2$ (jobs/s) |
| | Service rates of machines | $\mu_1, \mu_2, \cdots, \mu_{10}$ (jobs/s) |
| | Number of stations | $I = 10$ |
| | Total buffer capacity | $N = 30 \; or \; 60$ |
| Medium-scale manufacturing system | External arrival rate | $\lambda_1 = 1 \; or \; 2$ (jobs/s) |
| | Service rates of machines | $\mu_1, \mu_2, \cdots, \mu_{30}$ (jobs/s) |
| | Number of stations | $I = 30$ |
| | Total buffer capacity | $N = 90 \; or \; 180$ |
| Large-scale manufacturing system | External arrival rate | $\lambda_1 = 1 \; or \; 2$ (jobs/s) |
| | Service rates of machines | $\mu_1, \mu_2, \cdots, \mu_{50}$ (jobs/s) |
| | Number of stations | $I = 50$ |
| | Total buffer capacity | $N = 150 \; or \; 300$ |

**TABLE 2.** Throughput, buffer allocation solutions, and computational time of the BD-TS, STS, and ATS in Experiment 1.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{10}, N)$ | | A-(1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1,9, 2.0, 30) | B-(1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1,9 2.0, 60) | C-(1, 1.5, 1.5, ···, 1.5, 30) | D-(1, 1.5, 1.5, ···, 1.5, 60) | E-(2, 1.5, 1.5, ···, 1.5, 30) | F-(2, 1.5, 1.5, ···, 1.5, 60) |
|---|---|---|---|---|---|---|---|
| BD-TS ($P_A = P_B =$ 70%) | $\theta_{BEST}$ (Jobs/s) | 0.6973 | 0.9165 | 0.6836 | 0.9092 | 0.7786 | 1.0797 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6],4) | (8,8,7,7,[6*3],5,7) | ([3*7],4,5) | ([6*5],[7*3],8) | (2,[3*6],4,6) | (4,5,6,6,[7*4],11) |
| | $t_{BD-TS}$ (s) | 0.7275 | 1.9769 | 0.5772 | 2.1280 | 0.7117 | 1.3161 |
| BD-TS ($P_A = P_B =$ 60%) | $\theta_{BEST}$ (Jobs/s) | 0.6973 | 0.9165 | 0.6836 | 0.9092 | 0.7786 | 1.0797 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6],4) | (8,8,7,7,[6*3],5,7) | ([3*7],4,5) | ([6*5],[7*3],8) | (2,[3*6],4,6) | (4,5,6,6,[67*4],11) |
| | $t_{BD-TS}$ (s) | 0.5633 | 1.4320 | 0.6193 | 1.8341 | 0.7920 | 1.3723 |
| STS | $T_{BEST}$ (Jobs/s) | 0.6973 | 0.9165 | 0.6836 | 0.9092 | 0.7786 | 1.0797 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6],4) | (8,8,7,7,[6*3],5,7) | ([3*7],4,5) | ([6*5],[7*3],8) | (2,[3*6],4,6) | (4,5,6,6,[7*4],11) |
| | $t_{STS}$ (s) | 1.2489 | 3.3621 | 1.3436 | 3.6132 | 1.4761 | 2.5905 |
| ATS | $T_{BEST}$ (Jobs/s) | 0.6973 | 0.9165 | 0.6836 | 0.9092 | 0.7786 | 1.0797 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6],4) | (8,8,7,7,[6*3],5,7) | ([3*7],4,5) | ([6*5],[7*3],8) | (2,[3*6],4,6) | (4,5,6,6,[7*4],11) |
| | $t_{ATS}$ (s) | 1.2079 | 2.4447 | 1.3126 | 3.0092 | 1.2744 | 2.7812 |

Note: [e*f ] denotes a combination of the buffer capacity and buffers, indicating that the buffer capacity is allocated in contiguous buffers. For example, a buffer allocation solution (4,4,[3*6],4) represents (4,4,3,3,3,3,3,3,4).

**TABLE 3.** Computational efficiency improvement in Experiment 1.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{10}, N)$ | A-(1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1,9, 2.0, 30) | B-(1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1,9 2.0, 60) | C-(1, 1.5, 1.5, ···, 1.5, 30) | D-(1, 1.5, 1.5, ···, 1.5, 60) | E-(2, 1.5, 1.5, ···, 1.5, 30) | F-(2, 1.5, 1.5, ···, 1.5, 60) |
|---|---|---|---|---|---|---|
| $\varepsilon$ between BD-TS ($P_A = P_B = 70\%$) and STS | 41.75% | 41.20% | 58.64% | 41.11% | 51.79% | 49.19% |
| $\varepsilon$ between BD-TS ($P_A = P_B = 70\%$) and ATS | 39.77% | 19.14% | 56.03% | 29.28% | 44.15% | 52.68% |
| $\varepsilon$ between BD-TS ($P_A = P_B = 60\%$) and STS | 54.89% | 57.41% | 53.87% | 49.24% | 46.35% | 47.03% |
| $\varepsilon$ between BD-TS ($P_A = P_B = 60\%$) and ATS | 53.36% | 41.42% | 52.82% | 39.05% | 37.855 | 50.66% |

- The BD-TS can still achieve better computational efficiency than the STS and ATS, although the number of stations increases. In addition, the computational efficiency improves from 65%-90%, which is higher than that in Experiment 1. This results from the increase in the number of stations and the total buffer capacity, which significantly enlarges the search space. The variable neighborhood structure and the BD help decrease more selected feasible neighborhood solutions.
- Compared to unbalanced manufacturing systems, the BD-TS tends to have better computational efficiency in balanced manufacturing systems in the case of a lower total buffer capacity. Because of more obvious bottlenecks in unbalanced manufacturing systems, more iterations are required to increase buffer capacities in the bottlenecks based on the move representation in which only one buffer capacity is changed. This process requires a longer computational time. However, in the

**TABLE 4.** Throughput, buffer allocation solutions, and computational time of the BD-TS, STS, and ATS for Pattern A in the cases of different initialization schemes and service rate orders.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{10}, N)$ | | (1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1,9, 2.0, 30) | | (1, 1.5, 1.5, ···, 1.5, 30) | | (1, 2.0, 1.9, 1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1,2, 1.1, 30) | | (1, 1.3, 1.5, 1.1, 2.0, 1.7, 1.6, 1.4, 1.9, 1,2, 1.8, 30) | |
|---|---|---|---|---|---|---|---|---|---|
| | | RI | SRI | RI | SRI | RI | SRI | RI | SRI |
| BD-TS ($P_A = P_B =$ 70%) | $\theta_{BEST}$ (Jobs/s) | 0.6973 | 0.6973 | 0.6836 | 0.6836 | 0.6836 | 0.6836 | 0.6860 | 0.6860 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6], 4) | (4,4,[3*6], 4) | ([3*7],4,5) | ([3*7],4, 5) | (2,[3*6],4, 6) | (2,[3*6],4, 6) | (3,4,[3*5],4,4) | (3,4,[3*5],4, 4) |
| | $t_{BD-TS}$ (s) | 0.6275 | 0.4431 | 0.5772 | 0.5409 | 0.6549 | 0.4198 | 0.6169 | 0.4208 |
| BD-TS ($P_A = P_B =$ 60%) | $\theta_{BEST}$ (Jobs/s) | 0.6973 | 0.6973 | 0.6836 | 0.6836 | 0.6557 | 0.6557 | 0.6860 | 0.6860 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6], 4) | (4,4,[3*6], 4) | ([3*7],4,5) | ([3*7],4, 5) | ([3*7],4,5) | ([3*7],4,5) | (3,4,[3*5],4,4) | (3,4,[3*5],4, 4) |
| | $t_{BD-TS}$ (s) | 0.5633 | 0.5707 | 0.6193 | 0.5294 | 0.6934 | 0.4467 | 0.5434 | 0.3621 |
| STS | $T_{BEST}$ (Jobs/s) | 0.6973 | 0.6973 | 0.6836 | 0.6836 | 0.6557 | 0.6557 | 0.6860 | 0.6860 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6], 4) | (4,4,[3*6], 4) | ([3*7],4,5) | ([3*7],4, 5) | ([3*7],4,5) | ([3*7],4,5) | (3,4,[3*5],4,4) | (3,4,[3*5],4, 4) |
| | $t_{STS}$ (s) | 1.2489 | 1.0312 | 1.3436 | 0.8936 | 1.5676 | 1.4132 | 1.2194 | 1.0423 |
| ATS | $T_{BEST}$ (Jobs/s) | 0.6973 | 0.6973 | 0.6836 | 0.6836 | 0.6557 | 0.6557 | 0.6860 | 0.6860 |
| | $\bar{b}_{BEST}$ | (4,4,[3*6], 4) | (4,4,[3*6], 4) | ([3*7],4,5) | ([3*7],4, 5) | ([3*7],4,5) | ([3*7],4,5) | (3,4,[3*5],4,4) | (3,4,[3*5],4, 4) |
| | $t_{ATS}$ (s) | 1.2079 | 1.0222 | 1.3126 | 0.8851 | 1.2658 | 1.0741 | 1.2803 | 1.0369 |

**TABLE 5.** Computational efficiency improvement for Pattern A in the cases of different initialization schemes and service rate orders.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{10}, N)$ | (1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1,9, 2.0, 30) | | (1, 1.1, 1.5, 1.5, ···, 1.5, 30) | | (1, 2.0, 1.9, 1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1,2, 1.1, 30) | | (1, 1.3, 1.5, 1.1, 2.0, 1.7, 1.6, 1.4, 1.9, 1,2, 1.8, 30) | |
|---|---|---|---|---|---|---|---|---|
| | RI | SRI | RI | SRI | RI | SRI | RI | SRI |
| $\varepsilon$ between BD-TS $(P_A = P_B = 70\%)$ and STS | 41.75% | 57.03% | 58.64% | 39.47% | 58.22% | 70.29% | 49.41% | 59.63% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 70\%)$ and ATS | 39.77% | 56.65% | 56.03% | 38.89% | 48.26% | 60.92% | 51.82% | 59.42% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 60\%)$ and STS | 54.89% | 44.66% | 53.87% | 40.76% | 55.73% | 68.39% | 55.44% | 65.26% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 60\%)$ and ATS | 53.36% | 44.17% | 52.82% | 40.19% | 45.22% | 58.41% | 57.56% | 65.08% |

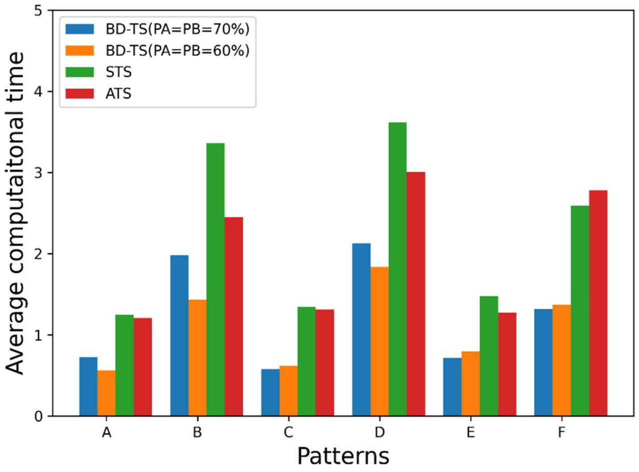

**FIGURE 6.** Average computational time of the BD-TS, STS, and ATS in Experiment 1.

case of a higher total buffer capacity, more buffer capacities are allocated into stations, and the phenomena of the obvious bottlenecks in unbalanced manufacturing systems decrease.

- The external arrival rate affects the computational efficiency of the BD-TS. A higher arrival rate leads to more jobs and blockings in the manufacturing system, which increases the difficulty of detecting the possible bottleneck for the BD. Therefore, the BD-TS requires more computational time to obtain a solution with sufficient quality.
- In Experiment 1, the ATS and TS have close computational efficiency. However, the ATS generates better computational efficiency than the TS in Experiment 2. With the increase of total buffer capacity and station number, the intensification and diversification strategies of the ATS contribute to searching for promising solutions in smaller solution areas and extending search space. Consequently, the computational time of the ATS decreases significantly.

### 2) SOLUTION QUALITY ANALYSIS

- The BD-TS, STS, and ATS achieve the same near-optimal solution, although the number of stations in the manufacturing system increases. This proves that the

**TABLE 6.** Throughput, buffer allocation solutions, and computational time for the BD-TS, STS, and ATS in Experiment 2.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{30}, N)$ | | A-(1, [1.1*10], [1.2*10], [1.3*10], 90) | B-(1, [1.1*10], [1.2*10], [1.3*10], 180) | C-(1, 1.5, 1.5, ⋯, 1.5, 90) | D-(1, 1.5, 1.5, ⋯, 1.5, 180) | E-(2, 1.5, 1.5, ⋯,1.5, 90) | F-(2, 1.5, 1.5, ⋯,1.5, 180) |
|---|---|---|---|---|---|---|---|
| BD-TS ($P_A =$ $P_B =$ 70%) | $\theta_{BEST}$ (Jobs/s) | 0.3889 | 0.6601 | 0.4794 | 0.7939 | 0.5052 | 0.8582 |
| | $\bar{b}_{BEST}$ | ([3*7],[4*2], [3*19], 4) | ([6*5],[7*4], [6*19], 8) | ([3*27],4,5) | ([6*25],[7*3],9) | (2,[3*25],4,4,5) | (3,5,5,[6*17], [7*8],9) |
| | $t_{BD-TS}$ (s) | 65.2264 | 117.2904 | 53.7846 | 116.3881 | 62.7407 | 110.5520 |
| BD-TS ($P_A =$ $P_B =$ 60%) | $\theta_{BEST}$ (Jobs/s) | 0.3889 | 0.6601 | 0.4794 | 0.7939 | 0.5052 | 0.8582 |
| | $\bar{b}_{BEST}$ | ([3*7],[4*2], [3*19], 4) | ([6*5],[7*4], [6*19], 8) | ([3*27],4,5) | ([6*25],[7*3],9) | (2,[3*25],4,4,5) | (3,5,5,[6*17], [7*8],9) |
| | $t_{BD-TS}$ (s) | 84.9040 | 133.7556 | 66.6157 | 131.4784 | 65.7225 | 152.5230 |
| STS | $\theta_{BEST}$ (Jobs/s) | 0.3889 | 0.6601 | 0.4794 | 0.7939 | 0.5052 | 0.8582 |
| | $\bar{b}_{BEST}$ | ([3*7],[4*2], [3*19], 4) | ([6*5],[7*4], [6*19], 8) | ([3*27],4,5) | ([6*25],[7*3],9) | (2,[3*25],4,4,5) | (3,5,5,[6*17], [7*8],9) |
| | $t_{STS}$ (s) | 380.2537 | 1300.8433 | 294.0526 | 596.7445 | 643.9596 | 744.5822 |
| ATS | $\theta_{BEST}$ (Jobs/s) | 0.3889 | 0.6601 | 0.4794 | 0.7939 | 0.5052 | 0.8582 |
| | $\bar{b}_{BEST}$ | ([3*7],[4*2], [3*19], 4) | ([6*5],[7*4], [6*19], 8) | ([3*27],4,5) | ([6*25],[7*3],9) | (2,[3*25],4,4,5) | (3,5,5,[6*17], [7*8],9) |
| | $t_{ATS}$ (s) | 286.1645 | 696.5552 | 288.5479 | 382.1472 | 428.6733 | 444.3709 |

BD-TS can maintain the solution quality for a manufacturing system with a medium scale.

- For balanced manufacturing systems, the buffer allocation tendency can still be described as a relatively small amount of buffer capacity in the first few stations, a relatively large amount of buffer capacity in the buffer of the final few stations, and even distribution of the remaining buffer capacity into the other stations.
- For unbalanced manufacturing systems, the service rates of stations fluctuate only slightly considering 30 stations. The bottleneck effect due to the unbalanced service rates is small, and there is no need to allocate more buffer capacities into the stations whose service rates are low. Therefore, the buffer allocation tendency is similar to that for balanced manufacturing systems.

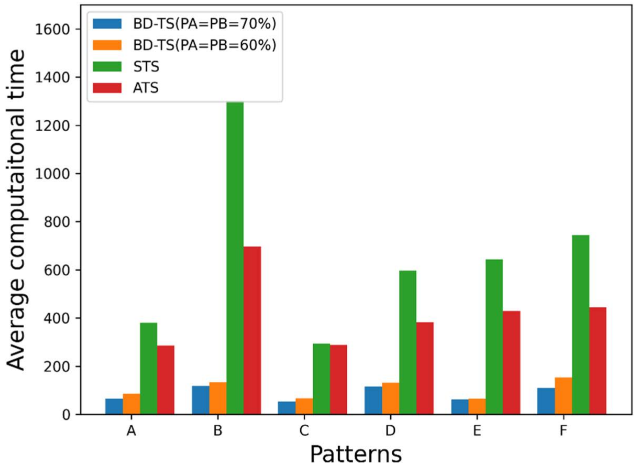

**FIGURE 7.** Average computational time of the BD-TS, STS, and ATS in Experiment 2.

## C. LARGE-SCALE MANUFACTURING SYSTEM

### 1) COMPUTATIONAL EFFICIENCY ANALYSIS

Tables 8 and 9 and Fig. 8 show the results of the BD-TS, STS, and ATS in Experiment 3. The following conclusions can be drawn:

- The BD-TS achieves better computational efficiency than STS and ATS. The computational efficiency improves from 70%-87%, which is close to that in Experiment 2. This means that the upper limit of the computational efficiency improvement for BD-TS is approximately 90%. Furthermore, for the large-scale problem, the computational time of the STS and ATS is around one hour. This computational efficiency is hardly acceptable in current industrial applications. However, the computational time of the BD-TS is under 12 minutes. With the increase in the problem scale, the advantage of the BD-TS will be more obvious.
- Similar to Experiment 2, the BD-TS tends to have better computational efficiency in balanced manufacturing systems in the case of a lower total buffer capacity. However, in the case of a higher total buffer capacity, the computational efficiency for unbalanced manufacturing systems decreases because the BD-TS obtains the local optimal solution, and the diversification strategy must be used.
- Compared to Experiments 1 and 2, the search space of the buffer allocation increases significantly. The BD-TS becomes more easily trapped in a local optimal solution; meanwhile, diversification strategies are used more frequently. This results in more iteration numbers and longer computational time to obtain the near-optimal solution. Therefore, the average computational time of five replicate calculations of the BD-TS in Patterns A, C, D, and F fluctuates obviously.

### 2) SOLUTION QUALITY ANALYSIS

- The BD-TS, STS, and ATS achieve the same near-optimal solution for the large-scale problem.

**TABLE 7.** Computational efficiency improvement in Experiment 2.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{30}, N)$ | A-(1, [1.1*10], [1.2*10], [1.3*10], 90) | B-(1, [1.1*10], [1.2*10], [1.3*10], 180) | C-(1, 1.5, 1.5, ⋯, 1.5, 90) | D-(1, 1.5, 1.5, ⋯, 1.5, 180) | E-(2, 1.5, 1.5, ⋯,1.5, 90) | F-(2, 1.5, 1.5, ⋯,1.5, 180) |
|---|---|---|---|---|---|---|
| $\varepsilon$ between BD-TS $(P_A = P_B = 70\%)$ and STS | 82.845 | 90.98% | 81.71% | 80.50% | 90.26% | 85.16% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 70\%)$ and ATS | 77.21% | 83.16% | 81.36% | 69.54% | 85.43% | 75.12% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 60\%)$ and STS | 77.67% | 89.72% | 77.35% | 77.97% | 89.79% | 79.52% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 60\%)$ and ATS | 70.33% | 80.80% | 76.91% | 65.59% | 84.67% | 65.68% |

**TABLE 8.** Throughput, buffer allocation solutions, and computational time of the BD-TS, STS, and ATS in Experiment 3.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{50}, N)$ | | A-(2, [1.5*9],2,[ 1.5*9], 2,[1.5*9],2, [1.5*9],2, [1.5*9],2, 150) | B-(2, [1.5*9],2,[ 1.5*9], 2,[1.5*9],2, [1.5*9],2, [1.5*9],2, 300) | C-(1, 1.5, 1.5, ⋯,1.5, 150) | D-(1, 1.5, 1.5, ⋯,1.5, 300) | E-(2, 1.5, 1.5, ⋯,1.5, 150) | F-(2, 1.5, 1.5, ⋯, 1.5, 300) |
|---|---|---|---|---|---|---|---|
| BD-TS $(P_A = P_B = 70\%)$ | $T_{BEST}$ (Jobs/s) | 0.4216 | 0.7871 | 0.3990 | 0.7321 | 0.4121 | 0.7699 |
| | $\bar{b}_{BEST}$ | (2,[3*44],[4*4]) | (3,[5*3],[6*4],5, [6*9],5, [6*9],5, [6*3],[7*6],5, [7*10]) | ([3*47],4,5) | ([6*44],[7*4],8) | (2,[3*45],4,4,5) | (3,[5*3],[6*35],[7*9],9) |
| | $t_{BD-TS}$ (s) | 301.9663 | 589.3453 | 266.4677 | 665.3851 | 281.8876 | 699.8312 |
| BD-TS $(P_A = P_B = 60\%)$ | $T_{BEST}$ (Jobs/s) | 0.4216 | 0.7871 | 0.3990 | 0.7321 | 0.4121 | 0.7699 |
| | $\bar{b}_{BEST}$ | (2,[3*44],[4*4]) | (3,[5*3],[6*4],5, [6*9],5, [6*9],5, [6*3],[7*6],5, [7*10]) | ([3*47],4,5) | ([6*44],[7*4],8) | (2,[3*45],4,4,5) | (3,[5*3],[6*35],[7*9],9) |
| | $t_{BD-TS}$ (s) | 376.4183 | 586.6122 | 263.2690 | 568.1390 | 309.2983 | 490.1430 |
| STS | $T_{BEST}$ (Jobs/s) | 0.4216 | 0.7871 | 0.3990 | 0.7321 | 0.4121 | 0.7699 |
| | $\bar{b}_{BEST}$ | (2,[3*44],[4*4]) | (3,[5*3],[6*4],5, [6*9],5, [6*9],5, [6*3],[7*6],5, [7*10]) | ([3*47],4,5) | ([6*44],[7*4],8) | (2,[3*45],4,4,5) | (3,[5*3],[6*35],[7*9],9) |
| | $t_{STS}$ (s) | 1778.4715 | 3609.9318 | 1758.3310 | 3680.4704 | 1603.1553 | 3781.5220 |
| ATS | $T_{BEST}$ (Jobs/s) | 0.4216 | 0.7871 | 0.3990 | 0.7321 | 0.4121 | 0.7699 |
| | $\bar{b}_{BEST}$ | (2,[3*44],[4*4]) | (3,[5*3],[6*4],5, [6*9],5, [6*9],5, [6*3],[7*6],5, [7*10]) | ([3*47],4,5) | ([6*44],[7*4],8) | (2,[3*45],4,4,5) | (3,[5*3],[6*35],[7*9],9) |
| | $t_{ATS}$ (s) | 1079.5467 | 2986.5651 | 968.1219 | 2831.3543 | 1034.8938 | 2907.0176 |

**TABLE 9.** Computational efficiency improvement in Experiment 3.

| Input patterns $(\lambda_1, \mu_1, \mu_2, \cdots, \mu_{50}, N)$ | A-(2, [1.5*9],2,[ 1.5*9], 2,[1.5*9],2, [1.5*9],2, [1.5*9],2, 150) | B-(2, [1.5*9],2,[ 1.5*9], 2,[1.5*9],2, [1.5*9],2, [1.5*9],2, 300) | C-(1, 1.5, 1.5, ⋯,1.5, 150) | D-(1, 1.5, 1.5, ⋯,1.5, 300) | E-(2, 1.5, 1.5, ⋯,1.5, 150) | F-(2, 1.5, 1.5, ⋯,1.5, 300) |
|---|---|---|---|---|---|---|
| $\varepsilon$ between BD-TS $(P_A = P_B = 70\%)$ and STS | 83.02% | 83.67% | 85.07% | 81.92% | 82.42% | 81.49% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 70\%)$ and ATS | 72.03% | 80.27% | 72.48% | 76.50% | 72.76% | 75.93% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 60\%)$ and STS | 78.83% | 83.75% | 85.035 | 84.53% | 80.71% | 87.04% |
| $\varepsilon$ between BD-TS $(P_A = P_B = 60\%)$ and ATS | 65.13% | 80.36% | 72.81% | 79.93% | 70.11% | 83.13% |

- For balanced manufacturing systems, the buffer allocation tendency is similar to Experiments 1 and 2. Because the subsequent stations exhibit more starving and blocking phenomena than the previous stations, the allocated

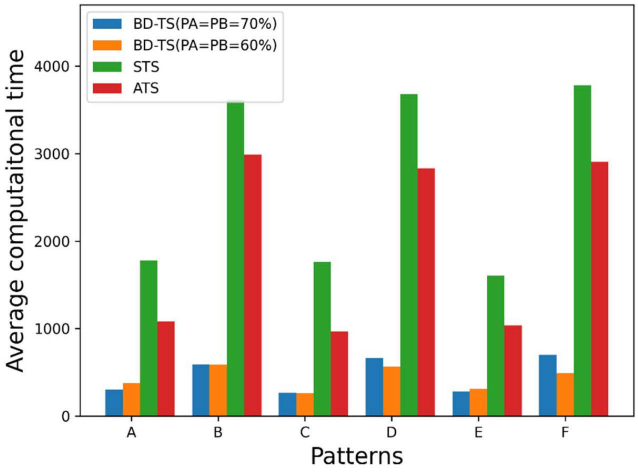

**FIGURE 8. Average computational time of the BD-TS, STS, and ATS in Experiment 3.**

buffer capacity of the previous stations is lower than that of the subsequent stations.

- For unbalanced manufacturing systems in Pattern A, the buffer allocation tendency is the same as that of balanced manufacturing systems. There are only three buffer capacities at each station on average. Compared to allocating the total buffer capacity evenly, allocating more buffer capacities to stations with low service rates may generate a larger bottleneck effect. Furthermore, in Pattern B, the total buffer capacity increases. The BD-TS tends to allocate fewer buffer capacities to stations with high service rates. However, to decrease the starve phenomena, the first station has a relatively low buffer capacity; meanwhile, to decrease the blocking phenomena, the few final stations have a relatively high buffer capacity.

### D. DISCUSSION

Regarding the computational efficiency, which is the computational time required to obtain the near-optimal solution, the BD-TS outperforms the STS and ATS for three experiments. The BD detects possible bottlenecks of a manufacturing system. Then, based on the possible bottlenecks, the BD guides move representation to efficiently search for better neighborhood solutions. Simultaneously, the variable neighborhood structure significantly decreases the selected feasible solutions. Furthermore, with the increase in the manufacturing system scale, the computational efficiency improvement of BD-TS increases. In addition, considering that the BD is used mainly in the process of neighborhood solution generation, it is possible to integrate the BD with other metaheuristics and evaluation algorithms to increase their computational efficiency. Regarding the solution quality, the BD-TS can achieve the same solution quality as the STS and ATS for the buffer allocation of manufacturing systems. Although the BD-TS only searches for a subset of all feasible solutions, diversification strategies are used to escape from the local optimal solution by enlarging the search scope and restarting the solution process.

By comparing the results in Experiments 1, 2, and 3, the computational efficiency of BD-TS changes with different $P_A$ and $P_B$; however, its solution quality remains high. The values of $P_A$ and $P_B$ are problem dependent. The high values of $P_A$ and $P_B$ seem more advantageous in improving the computational efficiency according to the experiments, although it is difficult to ascertain their specific setting strategy at this stage.

## VI. CONCLUSION AND FUTURE WORK

This study presented a BD-TS to efficiently solve the buffer allocation problem of manufacturing systems. In the BD-TS, diversification strategies were utilized to maintain the solution quality, while the BD and variable neighborhood structure were used to improve the computational efficiency. To verify its effectiveness, the BD-TS was compared to the STS and ATS in the numerical examples. Numerical examples show that the BD-TS can improve computational efficiency and achieve sufficient solution quality, although its computational efficiency fluctuates according to the setting of $P_A$ and $P_B$. Moreover, the computational efficiency improvement of the BD-TS increases with increasing problem scale, indicating the better performance of the BD-TS for large-scale manufacturing systems.

The main contribution of this study is the proposed approach to decrease the computational cost of the buffer allocation of manufacturing systems to meet efficient design requirements in real applications.

Even though the BD-TS improves the computational efficiency for the buffer allocation problem, the values of $P_A$ and $P_B$ require further study. The adaptive strategy to set $P_A$ and $P_B$ is one of our future works. In addition, buffers and service rates simultaneously affect the performance of manufacturing systems. Therefore, another direction of future work is to solve the simultaneous buffer and service rate allocation problem of manufacturing systems.

## REFERENCES

[1] L. Demir, S. Tunali, and D. T. Eliiyi, "The state of the art on buffer allocation problem: A comprehensive survey," *J. Intell. Manuf.*, vol. 25, no. 3, pp. 371–392, Jun. 2014, doi: 10.1007/s10845-012-0687-9.

[2] W. J. Gordon and G. F. Newell, "Cyclic queuing systems with restricted length queues," *Oper. Res.*, vol. 15, no. 2, pp. 266–277, Apr. 1967, doi: 10.1287/opre.15.2.266.

[3] M. Zhang, E. Pastore, A. Alfieri, and A. Matta, "Buffer allocation problem in production flow lines: A new Benders-decomposition-based exact solution approach," *IISE Trans.*, vol. 54, no. 5, pp. 421–434, Apr. 2021, doi: 10.1080/24725854.2021.1905195.

[4] S. Gao, T. Kobayashi, A. Tajiri, and J. Ota, "Throughput analysis of conveyor systems involving multiple materials based on capability decomposition," *Comput. Ind.*, vol. 132, Nov. 2021, Art. no. 103526, doi: 10.1016/j.compind.2021.103526.

[5] S. Gao, J. I. U. Rubrico, T. Higashi, T. Kobayashi, K. Taneda, and J. Ota, "Efficient throughput analysis of production lines based on modular queues," *IEEE Access*, vol. 7, pp. 95314–95326, 2019, doi: 10.1109/ACCESS.2019.2928309.

[6] Y. Alaouchiche, Y. Ouazene, and F. Yalaoui, "Economic and energetic performance evaluation of unreliable production lines: An integrated analytical approach," *IEEE Access*, vol. 8, pp. 185330–185345, 2020, doi: 10.1109/ACCESS.2020.3029761.

[7] M. Mohammadi, S. Dauzère-pérès, C. Yugma, and M. Karimi-Mamaghan, "A queue-based aggregation approach for performance evaluation of a production system with an AMHS," *Comput. Oper. Res.*, vol. 115, Mar. 2020, Art. no. 104838, doi: 10.1016/j.cor.2019.104838.

[8] Y. Bai, J. Tu, M. Yang, L. Zhang, and P. Denno, "A new aggregation algorithm for performance metric calculation in serial production lines with exponential machines: Design, accuracy and robustness," *Int. J. Prod. Res.*, vol. 59, no. 13, pp. 4072–4089, Jul. 2021, doi: 10.1080/00207543.2020.1757777.

[9] J. C. Hugan, "QUEST-queueing event simulation tool," in *Proc. Winter Simul. Conf.*, 1994, Art. no. 4864892, doi: 10.1109/WSC.1994.717248.

[10] D. G. Oljira, T. G. Abeya, G. Ofgera, and M. Gopal, "Manufacturing system modeling and performance analysis of mineral water production line using ARENA simulation," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 5, pp. 312–317, Jun. 2020, doi: 10.35940/ijeat.d8033.069520.

[11] S. Weiss, J. A. Schwarz, and R. Stolletz, "The buffer allocation problem in production lines: Formulations, solution methods, and instances," *IISE Trans.*, vol. 51, no. 5, pp. 456–485, May 2019, doi: 10.1080/24725854.2018.1442031.

[12] M. G. Huang, P. L. Chang, and Y. C. Chou, "Buffer allocation in flow-shop-type production systems with general arrival and service patterns," *Comput. Oper. Res.*, vol. 2002, doi: 10.1016/S0305-0548(00)00060-5.

[13] H. Wang and H.-P. Wang, "Optimum number of kanbans between two adjacent workstations in a JIT system," *Int. J. Prod. Econ.*, vol. 22, no. 3, pp. 179–188, Dec. 1991, doi: 10.1016/0925-5273(91)90093-9.

[14] A. C. Diamantidis and C. T. Papadopoulos, "A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines," *Math. Problems Eng.*, vol. 2004, no. 3, pp. 209–223, 2004, doi: 10.1155/S1024123X04402014.

[15] C. Shi and S. B. Gershwin, "A segmentation approach for solving buffer allocation problems in large production systems," *Int. J. Prod. Res.*, vol. 54, no. 20, pp. 6121–6141, Oct. 2016, doi: 10.1080/00207543.2014.991842.

[16] L. Demir and M. U. Koyuncuoğlu, "The impact of the optimal buffer configuration on production line efficiency: A VNS-based solution approach," *Expert Syst. Appl.*, vol. 172, Jun. 2021, Art. no. 114631, doi: 10.1016/j.eswa.2021.114631.

[17] J. Zhang, W. Ran, D. Wan, and N. Liu, "Dynamic buffer allocation and monitoring based on comprehensive activity sensitivity," *J. Syst. Manage.*, vol. 31, no. 1, pp. 150–158, 2022.

[18] F. R. B. Cruz, A. R. Duarte, and T. van Woensel, "Buffer allocation in general single-server queueing networks," *Comput. Oper. Res.*, vol. 35, no. 11, pp. 3581–3598, Nov. 2008, doi: 10.1016/j.cor.2007.03.004.

[19] S. Gao, T. Higashi, T. Kobayashi, K. Taneda, and J. Ota, "Fast buffer size design of production lines for meeting the desired throughput," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 1206–1418, doi: 10.1109/ROBIO.2018.8664907.

[20] S. Gao, T. Higashi, T. Kobayashi, K. Taneda, J. I. U. Rubrico, and J. Ota, "Buffer allocation via bottleneck-based variable neighborhood search," *Appl. Sci.*, vol. 10, no. 23, p. 8569, Nov. 2020, doi: 10.3390/app10238569.

[21] K. Kassoul, N. Cheikhrouhou, and N. Zufferey, "Buffer allocation design for unreliable production lines using genetic algorithm and finite perturbation analysis," *Int. J. Prod. Res.*, vol. 60, no. 10, pp. 3001–3017, Apr. 2021, doi: 10.1080/00207543.2021.1909169.

[22] J. O. Hernández- Vázquez, S. Hernández-González, J. I. Hernández- Vázquez, V. Figueroa-Fernández, and C. I. C. de la Fuente, "Buffer allocation problem in a shoe manufacturing line: A metamodeling approach," *Revista Facultad de Ingeniería Universidad de Antioquia*, vol. 103, pp. 175–185, Jul. 2021.

[23] B.-H. Zhou, Y.-W. Liu, J.-D. Yu, and D. Tao, "Optimization of buffer allocation in unreliable production lines based on availability evaluation," *Optim. Control Appl. Methods*, vol. 39, no. 1, pp. 204–219, Jan. 2018, doi: 10.1002/oca.2341.

[24] M. U. Koyuncuoğlu and L. Demir, "A comparison of combat genetic and big bang–big crunch algorithms for solving the buffer allocation problem," *J. Intell. Manuf.*, vol. 32, no. 6, pp. 1529–1546, Aug. 2021, doi: 10.1007/s10845-020-01647-1.

[25] S. Y. Kose and O. Kilincci, "Hybrid approach for buffer allocation in open serial production lines," *Comput. Oper. Res.*, vol. 60, pp. 67–78, Aug. 2015, doi: 10.1016/j.cor.2015.01.009.

[26] S. Weiss, A. Matta, and R. Stolletz, "Optimization of buffer allocations in flow lines with limited supply," *IISE Trans.*, vol. 50, no. 3, pp. 191–202, Mar. 2018, doi: 10.1080/24725854.2017.1328751.

[27] F. Glover, V. Campos, and R. Martí, "Tabu search tutorial. A graph drawing application," *TOP*, vol. 29, no. 2, pp. 319–350, Jul. 2021, doi: 10.1007/s11750-021-00605-1.

[28] F. Glover, M. Laguna, and R. Marti, "Principles of tabu search," *Approximation Algorithms Metaheuristics*, vol. 23, pp. 1–12, Jan. 2007.

[29] C. M. Lutz, K. R. Davis, and M. Sun, "Determining buffer location and size in production lines using tabu search," *Eur. J. Oper. Res.*, vol. 106, pp. 301–316, Apr. 1998, doi: 10.1016/S0377-2217(97)00276-2.

[30] A. Costa, A. Alfieri, A. Matta, and S. Fichera, "A parallel Tabu search for solving the primal buffer allocation problem in serial production systems," *Comput. Oper. Res.*, vol. 64, pp. 97–112, Dec. 2015, doi: 10.1016/j.cor.2015.05.013.

[31] L. Demir, S. Tunalı, and D. T. Eliiyi, "An adaptive Tabu search approach for buffer allocation problem in unreliable non-homogenous production lines," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1477–1486, Jul. 2012, doi: 10.1016/j.cor.2011.08.019.

[32] C. T. Papadopoulos, M. E. J. O'Kelly, and A. K. Tsadiras, "A DSS for the buffer allocation of production lines based on a comparative evaluation of a set of search algorithms," *Int. J. Prod. Res.*, vol. 51, no. 14, pp. 4175–4199, Jul. 2013, doi: 10.1080/00207543.2012.752585.

[33] L. Demir, S. Tunali, and A. Løkketangen, "A Tabu search approach for buffer allocation in production lines with unreliable machines," *Eng. Optim.*, vol. 43, no. 2, pp. 213–231, Feb. 2011, doi: 10.1080/0305215X.2010.481022.

[34] M. Ouzineb, F.-Z. Mhada, R. Pellerin, and I. el Hallaoui, "A hybrid method for solving buffer sizing and inspection stations allocation," in *Proc. Int. Conf. Adv. Prod. Manage. Syst.*, 2014, pp. 156–166, doi: 10.1007/978-3-662-44733-8_20.

[35] C. Su, Y. Shi, and J. Dou, "Multi-objective optimization of buffer allocation for remanufacturing system based on TS-NSGAII hybrid algorithm," *J. Cleaner Prod.*, vol. 166, pp. 756–770, Nov. 2017, doi: 10.1016/j.jclepro.2017.08.064.

[36] A. Delgoshaei, M. Parvin, and M. K. A. Ariffin, "Evaluating impact of market changes on increasing cell-load variation in dynamic cellular manufacturing systems using a hybrid Tabu search and simulated annealing algorithms," *Decis. Sci. Lett.*, vol. 5, no. 2, pp. 219–244, 2016, doi: 10.5267/j.dsl.2015.12.002.

[37] S. Balsamo, *Queueing Networks with Blocking: Analysis, Solution Algorithms and Properties*. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-642-02742-0_11.

[38] L. Kerbache and J. M. Smith, "Asymptotic behavior of the expansion method for open finite queueing networks," *Comput. Oper. Res.*, vol. 15, pp. 157–169, Jan. 1988, doi: 10.1016/0305-0548(88)90008-1.

[39] J. M. Smith, "Simultaneous buffer and service rate allocation in open finite queueing networks," *IISE Trans.*, vol. 50, no. 3, pp. 203–216, Mar. 2018, doi: 10.1080/24725854.2017.1300359.

[40] J. M. Smith, "$M/G/c/K$ blocking probability models and system performance," *Perform. Eval.*, vol. 52, no. 4, pp. 237–267, May 2003, doi: 10.1016/S0166-5316(02)00190-6.

[41] J. M. Smith and F. R. B. Cruz, "The buffer allocation problem for general finite buffer queueing networks," *IIE Trans.*, vol. 37, no. 4, pp. 343–365, Feb. 2005, doi: 10.1080/07408170590916986.

[42] A. M. Andrew, "Modern heuristic search methods," Kybernetes. vol. 27, no. 5, pp. 582–585, 1998, doi: 10.1108/k.1998.27.5.582.3.

**SIXIAO GAO** received the B.S. degree in mechanical, manufacturing, and automation engineering and the M.S. degree in mechanical engineering from Central South University, Changsha, China, in 2011 and 2014, respectively, and the Ph.D. degree in precision engineering from The University of Tokyo, Tokyo, Japan, in 2020.

He is currently a Postdoctoral Researcher with the School of Traffic and Transportation Engineering, Central South University. His research interests include the design and optimization of manufacturing and distribution systems.