

Buffer clustering policy for sequential production lines with deterministic processing times

Francesca Schuler* and Houshang Darabi

Mechanical and Industrial Engineering, University of Illinois, Chicago, USA

CHRONICLE

ABSTRACT

Article history:

Received April 4 2016
Received in Revised Format
April 27 2016
Accepted May 6 2016
Available online
May 6 2016

Keywords:

*Sequential
Production
Buffer
Cluster
Deterministic
Configuration*

A sequential production line is defined as a set of sequential operations within a factory or distribution center whereby entities undergo one or more processes to produce a final product. Sequential production lines may gain efficiencies such as increased throughput or reduced work in progress by utilizing specific configurations while maintaining the chronological order of operations. One problem identified by the authors via a case study is that, some of the configurations, such as work cell or U-shaped production lines that have groups of buffers, often increase the space utilization. Therefore, many facilities do not take advantage of the configuration efficiencies that a work cell or U-shaped production line provide. To solve this problem, the authors introduce the concept of a buffer cluster. The production line implemented with one or more buffer clusters maintains the throughput of the line, identical to that with dedicated buffers, but with the clusters reduces the buffer storage space. The paper derives a time based parametric model that determines the sizing of the buffer cluster, provides a reduced time space for which to search for the buffer cluster sizing, and determines an optimal buffer clustering policy that can be applied to any N-server, N+1-buffer sequential line configuration with deterministic processing time. This solution reduces the buffer storage space utilized while ensuring no overflows or underflows occur in the buffer. Furthermore, the paper demonstrates how the buffer clustering policy serves as an input into a facility layout tool that provides the optimal production line layout.

© 2016 Growing Science Ltd. All rights reserved

1. Introduction

A sequential production line is defined as a set of sequential operations within a factory or distribution center whereby entities undergo one or more processes to produce a final product. Manufacturing and distribution facilities are facing growing competition and searching for ways to maximize production efficiency to remain competitive (Aghazadeh et al., 2011). Facilities are assessing alternate production line configurations to gain production efficiencies such as throughput increases or work in progress reduction while maintaining the chronological order of operations. A case study of a manufacturing facility (details found in section 4) desired to transition from a serial line configuration (Fig. 1) to a

* Corresponding author. Tel: +1 (224) 715-6799
E-mail: fruffo2@uic.edu (F. Schuler)

configuration in Fig. 2 called a hybrid serial-work cell configuration to gain increased throughput. As a result of this change, the facility encountered a production floor space utilization problem. In these figures, the squares without a grid pattern represent server stations where a process step occurs. The squares with a grid pattern represent buffers. The stations may be manual, semi-automated, or fully automated. The groups of buffers in Fig. 2 may vary in the number of buffers within the work cell and the number of serial stations in between the work cells as shown.

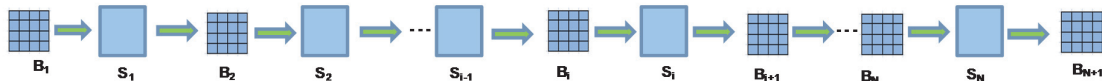


Fig. 1. N-server, N+1 buffer serial line

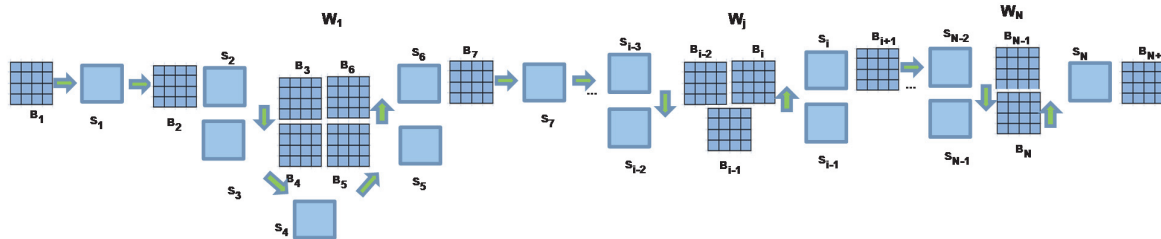


Fig. 2. Variable station work cells with one or more serial stations in between

In this scenario, because the buffers were sized separately (i.e. dedicated buffers) with respect to the serial line and grouped in the center of the work cell, the grouped buffers were not leveraging available space in the neighboring buffers during the production shift. Therefore, the work cell exceeded the typical space of the production line. The authors proposed transitioning the grouped buffers in the center of the work cell to a single buffer cluster shown in Fig. 3 which enables increased buffer utilization and reduces the size of the grouped buffers, reducing the buffer storage space. This allows the facility to benefit from efficiencies (e.g., increased throughput, work in progress reduction) by use of alternate configurations. In addition, discussed in the case study in section 4 is how sensitivity analysis of the buffer cluster size can be conducted using the models derived herein varying parameters such as the production demand and process times.

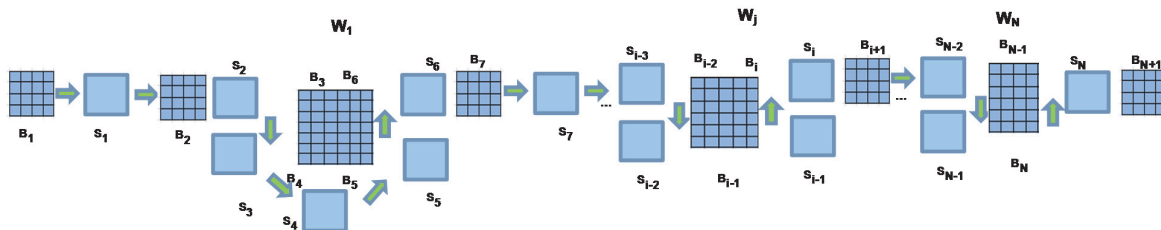


Fig. 3. Variable station work cells with serial stations in between using buffer cluster concept

In the case of this facility in a manual environment, as with many facilities globally, the use of a bar code or radio frequency identification (RFID) is utilized which aids in facilitating the buffer cluster concept. Each time a product moves from one station to another, the product bar code or RFID is scanned to ensure the prior processes are completed. Only if the prior processes are completed is that product picked from the buffer for the operator to perform the process at that station. Once the process is completed, the operator scans the product to inform the system that this process has been completed and puts the product back into the buffer. While the server or operator is processing an entity, the system via RFID identifies the next part(s) for the server / operator to pick up and notifies the operator via a light or other indicator. Thus there is no impact to the process times. Similarly, in a more fully automated environment, while the station is processing an existing entity, RFID technology identifies the next entity to process; a robot picks it up and delivers the entity to the station for processing. The buffer cluster may be partitioned and marked such that each station has a core area utilized by only that station and a shared area. Operators or robots first focus on filling their core area and then move to the shared area if needed.

Once the buffer clustering policy is identified for a production line, an activity relationship chart is created for the buffers and stations in the production line and the amount of space assigned to each activity is determined. From the space relationship diagram, one or more feasible layout concepts are generated. The optimal production line layout is then selected.

This paper proceeds with a literature review in the second section. The third section describes the problem formulation and derives the model that is used to solve the size of the buffer cluster for any N-server, N+1-buffer, sequential line. An optimization framework is derived that enables a buffer clustering policy and provides an output of the buffer sizing for that policy that ensures no buffer overflows. In the fourth section, the authors apply the model to the case study and review the results. The final section discusses the conclusions.

2. Literature review

Given the industry example discussed in the introduction, how the research at hand differs from the prior art is assessed. Quantitative analysis of production lines includes the line balancing problem (Becker & Scholl, 2006), the buffer allocation problem (Charharsooghi & Nahavandi, 2003), queuing network performance and blocking (Govil & Fu, 1999; Li et al., 2009). Literature also discusses these concepts explored with the addition of flexible manufacturing systems with varying configurations (serial, sequential, work-cells) and product types (Senanayake & Subramaniam, 2013; So, 1989).

The line balancing problem includes the positioning and sizing of the buffers and the overall throughput resulting in the buffer allocation problem. Wei et al. (1989) estimate stochastically via optimization the buffer size in a serial manufacturing system. Chaharsooghi and Nahavandi (2003) present a heuristic algorithm to find the optimal allocation of buffers that maximizes throughput. Yamashita and Altiok (1998) implement a dynamic programming algorithm that uses decomposition to minimize buffer allocation but still meeting target throughput in production lines with phase-type processing times. There are several buffer allocation strategies: (1) Equal Buffer allocates buffers equally over the line, (2) Chow's Rule (Chow, 1987) uses dynamic programming to solve the buffer allocation problem with a fixed total buffer size, (3) L&L's Rule (Liu & Lin (1994)), uses a different set of equations than Chow to estimate the throughput and (4) C&N's rule (Chan & Ng (2002)) where all possible allocations of buffers are tried and then the allocation with the highest estimated throughput is selected. Gershwin and Schor (1999) minimize the total buffer space using a production rate constraint while maximizing production rate using a total buffer space constraint. Enginarlar et al. (2000) ensure the smallest level of buffering with the desired production rate in serial lines with unreliable machines. Enginarlar et al. (2002) also introduces the concept of Lean Level of Buffering where buffer capacity is normalized and production line efficiency is achieved using exact methods for three machine lines and estimation methods for lines with more than three machines.

Queuing network performance literature includes quantitative analysis for production systems and considers production rates, average buffer levels and probabilities of blockage (Govil & Fu, 1999; Li et al., 2009). Gershwin (1987) developed a method using conservation of flow for evaluation of performance measures for lines with finite buffers. Lim et al. (1990) developed an aggregation method that converges on a production rate eliminating the need for simulations. Kouikoglou and Phillis (1991, 1994 and 1995) use a probabilistic technique that observes a limited number of events which are sufficient to determine the system performance and average buffer sizes. Morrison (2010) demonstrates that flow line models with deterministic service times using recursion to calculate the overall delay of entities in the system.

Considering flexible manufacturing systems and work cell literature, Ramirez-Serrano and Benhabib (2000) introduce a control algorithm to analyze concurrent operation of supervisors to determine absence of deadlocks within a work-cell. Outside of supervisory control, there have been several studies that

investigate the utilization of work-cell and reconfigurable manufacturing systems to increase the efficiency and capacity of production lines. Ichikawa's study (2009), for example, investigates a laptop production system and optimizing the supply of parts via material handlers from the receiving area to the cells. Another study (Aghazadeh et al., 2011) analyzes use of product-oriented layout, material handling and layout of work-cells to maximize production efficiency in areas such as average units produced per day, labor cost per unit and distance traveled to procure parts. Logendran and Karim (2003) uses a non-linear programming model consisting of binary and integer variables and a tabu search type algorithm to determine the availability of alternative locations for a work-cell and the use of alternative routes to move part loads between cells when capacity is limited. Youssef and ElMaraghy (2007) introduce a configuration selection approach that minimizes reconfiguration effort but still supports the capacity needs of production. Matta et al. (2005) use discrete event simulation and graph theory to demonstrate that technological devices moving entities from a machine to a single common buffer area of the system can improve production rates.

This paper differs from the prior literature reviewed in that it presents methods for extracting the buffer size where the buffer space is shared by several stations (via a buffer cluster) using methods derived from state space parameters with respect to time for any sequential N-server, N+1-buffer production line. The buffer sizing model is then utilized in an optimization framework that enables setting of the policy specifying the buffers that can be clustered ensuring no buffer overflows. The model provides an output of the required buffer cluster sizing for that policy and allows the facility to set the policy that minimizes space utilization of the production line.

3. Buffer cluster model

The buffer cluster model is presented in two parts. The first part (section 3.1) is the problem formulation where no clustering is assumed. This provides the model ultimately used for sizing the buffer cluster for any N-server, N+1-buffer sequential production line. In the second part (section 3.2), the model in section 3.1 is extended to a buffer cluster. Section 3.2 derives the optimization framework that determines the buffer cluster sizing required and enables a buffer clustering policy.

3.1. Buffer sizing problem formulation

In this section, the parameters used for deriving the formulation ultimately used for sizing the buffer cluster in any N-server, N+1-buffer sequential production line are defined. The number of arrivals and departures at Server S_i , $i = 1, 2, \dots, N$ by any given time t are calculated. Next the number of arrivals and departures from any buffer B_i , $i = 1, 2, \dots, N+1$ by any given time t are derived. The relationships are generated to derive the maximum number of entities (parts) a buffer will experience and the number of entities at any given time t , $B_i(t)$. $B_i(t)$ is then extended in section 3.2 to determine the buffer cluster size. Before deriving the aforementioned relationships, the notations, assumptions and definitions are listed.

Notations:

- N1) K_1 = Magnitude of inventory at B_1 at time $t=0$, $K_1 = 1, 2, \dots, N$ (Constant).
- N2) $BA_i(t)$ = Cumulative number of arrivals at buffer B_i by time t , $i=1, 2, \dots, N+1$.
- N3) $BD_i(t)$ = Cumulative number of departures from buffer B_i by time t , $i=1, 2, \dots, N+1$.
- N4) $SA_i(t)$ = Cumulative number of arrivals at server S_i by time t , $i=1, 2, \dots, N$.
- N5) $SD_i(t)$ = Cumulative number of departures from server S_i by time t , $i=1, 2, \dots, N$.
- N6) T_i , $i = 1 \dots N$ is the service time for server S_i , $i = 1 \dots N$ (This includes both the process time of the unit and the transportation time of the unit from the buffer to the station and the station to the next buffer).

Assumptions:

- A1) Each Server S_i can process at most one entity at a time (capacity = 1).
- A2) Each buffer $B_i, i = 1 \dots N+1$, has a capacity greater or equal to the starting inventory.
- A3) Service time T_i for each server S_i is deterministic.
- A4) At $t=0$ K1 is located in buffer B_1 and all other buffers are empty.
- A5) At time $t = 0, B_1$ has a departure and S_1 has an arrival.
- A6) Buffer B_1 has only departures while Buffer B_{N+1} has only arrivals and every buffer B_i in between has both departures and arrivals; $BA_1(t) = 0; BD_{N+1}(t) = 0$ as shown in Fig. 1.
- A7) If there is at least one part in B_i and S_i is idle, then with no delay, an entity (part) is moved to S_i for processing.
- A8) Machines are reliable.

Definitions:

- D1) $MT_i = \max[T_1, T_2 \dots T_i], i=1, 2, \dots N$.
- D2) $\tau_i = \sum_{j=1}^i T_j, i=1, 2, \dots N$, and $\tau_0 = 0$.
- D3) $MB_i =$ Maximum number of entities that buffer $B_i, i = 2, \dots, N$ will experience.
- D4) $\lfloor \cdot \rfloor$ is a floor function that maps a real number to the largest previous integer value.
- D5) It is trivial to see that the frequency of arrivals to server S_i is $\frac{1}{MT_i}$.

A framework utilizing deterministic process times is used as in practice data is not available to determine the correct process time probability distributions. In addition, probabilistic models do not allow production line managers to keep a pulse of the production line by knowing the status of each buffer or station at any given time. This framework enables sensitivity analysis by varying process times for one or more stations which will be shown in the case study (section 4).

This model supports adding inventory (that could be in the form of batches of varying sizes) anytime before the last item in inventory B_1 leaves the first server S_1 (i.e. a batch can be added anytime before $t = \tau_{i-1} + (K1 - 1) * MT_i$). K1 can be either the initial inventory or a summation of inventory (in the form of batches) throughout the shift.

The number of arrivals and departures from server S_i are now calculated.

Theorem 1: For the sequential system the cumulative arrivals and cumulative departures at server S_i at time t is:

$$SA_i(t) = \begin{cases} \min \left\{ K1, 1 + \left\lfloor \frac{t - \tau_{i-1}}{MT_i} \right\rfloor \right\} & \text{if } t \geq \tau_{i-1} \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

$$SD_i(t) = \begin{cases} \min \left\{ K1, 1 + \left\lfloor \frac{t - \tau_i}{MT_i} \right\rfloor \right\} & \text{if } t \geq \tau_i \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

Proof: First, Eq. (1) holds for $i=1$ is shown. At $i=1, \tau_1 = \sum_{j=1}^1 T_j = T_1 = MT_1. \tau_0 = 0$. Based on the sequential line assumption, at time $t = 0$, one part is loaded to server S_1 (recall that $K1 \geq 1$). This part is processed for $T_1 = \tau_1$ units of time and if buffer B_1 still carries a part, server S_1 is loaded again. This loading operation (arrival event) happens at time $t = \tau_1$. Continuing with this pattern, one can see that server S_1 is loaded at time stamp $0, \tau_1, 2\tau_1, \dots, K1\tau_1$, therefore the last loading of server S_1 happens at time $t = K1\tau_1$. After this time no loading occurs as all the parts in buffer B_1 have been depleted, and the total number of arrivals to S_1 remains K1. This means that the cumulative number of arrivals to server S_1 at time t can be shown by:

$$SA_1(t) = \begin{cases} \left\lfloor \frac{t}{\tau_1} \right\rfloor + 1 & 0 \leq t < K1\tau_1 \\ K1 & t \geq K1\tau_1 \end{cases}$$

and it can immediately be concluded that $SA_i(t) = \min \left\{ K1, 1 + \left\lfloor \frac{t}{MT_i} \right\rfloor \right\}$, $t \geq 0$.

This proves that Eq. (1) holds for $i=1$. Second, Eq. (1) is proven to hold for S_i where $1 < i \leq N$. By definition for $t < \tau_{i-1}$, $SA_i(t) = 0$ and for $t = \tau_{i-1}$, $SA_i(t) = 1$. Fig. 4 is based on the result of definition D5. It shows that the cumulative arrivals to server S_i at any time t . The interarrival times are MT_i . The case where $1 < SA_i(t) \leq K1$ is now considered.

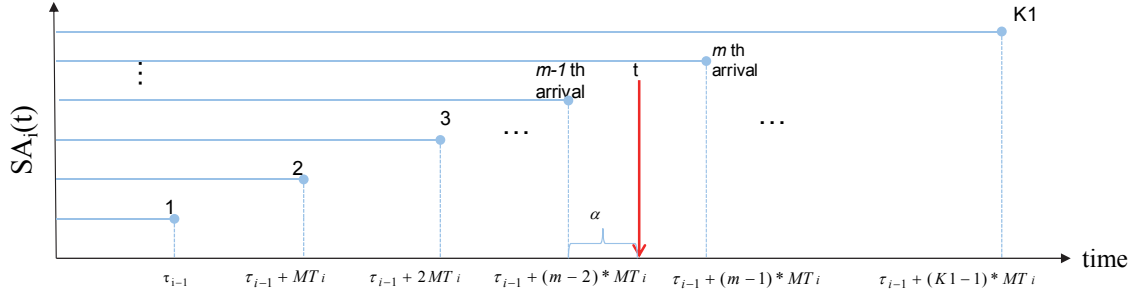


Fig. 4. Time defined for $SA_i(t)$ from 1 through $K1^{th}$ arrival

Assume that $\tau_{i-1} + (m-2)*MT_i < t < \tau_{i-1} + (m-1)*MT_i$ where $m-1$ is an integer number and is the number of arrivals before t . From the relationship that the frequency of arrivals to Server S_i is $\frac{1}{MT_i}$ and that for the $m-1$ th and m th arrivals, time is defined in the following interval: $\tau_{i-1} + (m-2)*MT_i < t < \tau_{i-1} + (m-1)*MT_i$, t is defined as:

$$t = \tau_{i-1} + (m-2)*MT_i + \alpha*MT_i \text{ where } 0 < \alpha \leq 1$$

$$t - \tau_{i-1} = ((m-2) + \alpha)*MT_i$$

When $\alpha = 1$, the coefficient $((m-2) + \alpha) = m-1 = \left\lfloor \frac{t - \tau_{i-1}}{MT_i} \right\rfloor$ and S_i experiences the m th arrival. For $0 < \alpha < 1$, the coefficient $[(m-2) + \alpha] = m-2 = \left\lfloor \frac{t - \tau_{i-1}}{MT_i} \right\rfloor$ and S_i has experienced the $m-1$ th arrival. Therefore, for $1 \leq SA_i(t) \leq K1$, $SA_i(t) = 1 + \left\lfloor \frac{t - \tau_{i-1}}{MT_i} \right\rfloor$. Based on the definition of arrival and departure of entities from server S_i one can see that because of the relationship

$$SD_i(t) = SA_i(t + T_i) \quad (3)$$

that means Eq. (2) holds. ■

Corollary 1: For the sequential system described in Fig. 1 the cumulative arrivals and cumulative departures at buffer B_i at time t are:

$$BA_i(t) = \begin{cases} \min \left\{ K1, 1 + \left\lfloor \frac{t - \tau_{i-1}}{MT_{i-1}} \right\rfloor \right\} & \text{if } t \geq \tau_{i-1} \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

$$BD_i(t) = \begin{cases} \min \left\{ K1, 1 + \left\lfloor \frac{t - \tau_{i-1}}{MT_i} \right\rfloor \right\} & \text{if } t \geq \tau_{i-1} \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

Proof: For any B_i where $i = 2, 3 \dots N+1$, the number of arrivals at B_i is equal to the number of departures from S_{i-1} at a given time t . Therefore taking Eq. (2) from the perspective of S_{i-1} and applying the condition from Eq. (6) proves Eq. (4).

$$BA_i(t) = SD_{i-1}(t) \quad (6)$$

The number of departures from B_i is equal to the number of arrivals at server S_i . Therefore, taking Eq. (1) from the perspective of arrivals at S_i and applying the condition from Eq. (7) proves Eq. (5).

$$BD_i(t) = SA_i(t) \quad \blacksquare \tag{7}$$

Corollary 2: For the sequential system, the maximum number of entities that buffer B_i will experience given starting inventory $K1$ as shown in Fig. 5 is:

$$MB_i = (K1 - 1) - \lfloor Y_i * (K1 - 1) \rfloor \tag{8}$$

Where $Y_i = \frac{MT_{i-1}}{MT_i}$ for $i = 2 \dots N$.

When $T_i > MT_{i-1}$, then $MT_{i-1} < MT_i$ and $Y_i < 1$. When $T_i \leq MT_{i-1}$, then $MT_i = MT_{i-1}$ and $Y_i = 1$. This is proven for MB_i , $i = 2, 3, \dots, N$ and when $Y_i < 1$.

When $Y_i = 1$, $MB_i = 0$, thus a buffer size = 1 is required for transport only to the next process. This is called a transport buffer and $MB_i = 1$ is assigned.

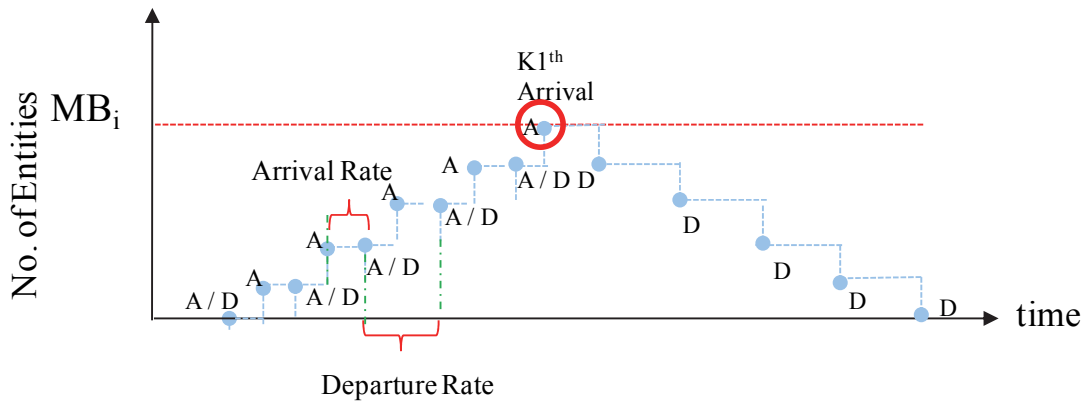


Fig. 5. Maximum number of entities that buffer B_i will experience given inventory $K1$

For B_i of interest, at any given time t the number of entities is equal to:

$$B_i(t) = BA_i(t) - BD_i(t), \quad B_i(t) \geq 0 \tag{9}$$

The time (say T) when $B_i(t)$ reaches its maximum level is derived and then entered into Eq. (9), therefore calculating MB_i . By assumption, the first departure from S_{i-1} and the first arrival to S_i happen simultaneously. After this event, because $MT_i = T_i > MT_{i-1}$ and using the relationship that the frequency of arrivals to Server S_i is $\frac{1}{MT_i}$, the departure rate from S_{i-1} will be greater than the arrival rate to S_i . This causes the accumulation in B_i to increase until the last departure ($K1$ th departure) from S_{i-1} occurs. Therefore the maximum accumulation happens when the last departure from S_{i-1} occurs.

$$T = MT_{i-1} * (K1 - 1) + \tau_{i-1} \tag{10}$$

Plugging in T from Eq. (10) into Eq. (11) and using the results of Corollary 1:

$$B_i(t) = BA_i(t) - BD_i(t) = \min \left(K1, 1 + \left\lfloor \frac{t - \tau_{i-1}}{MT_{i-1}} \right\rfloor \right) - \min \left(K1, 1 + \left\lfloor \frac{t - \tau_{i-1}}{MT_i} \right\rfloor \right) \tag{11}$$

$$= MB_i = \left\lfloor \frac{MT_{i-1} * (K1 - 1)}{MT_{i-1}} \right\rfloor - \left\lfloor \frac{MT_{i-1} * (K1 - 1)}{MT_i} \right\rfloor = (K1 - 1) - \lfloor Y_i * (K1 - 1) \rfloor \quad \blacksquare$$

3.2. Optimization framework formulation for buffer clustering policy

In this part, the buffer clustering optimization framework is derived utilizing the model from section 3.1 to provide the buffer cluster sizing and an optimal buffer clustering policy. Before deriving the

forementioned relationships, the notations, assumptions and definitions are listed. Fig. 6 which shows the inventory profile in buffer B_i is used to illustrate the new notations $K2_i$, $K3_i$, \bar{t}_{1B_i} , \bar{t}_{2B_i} and $MB_i - 1$.

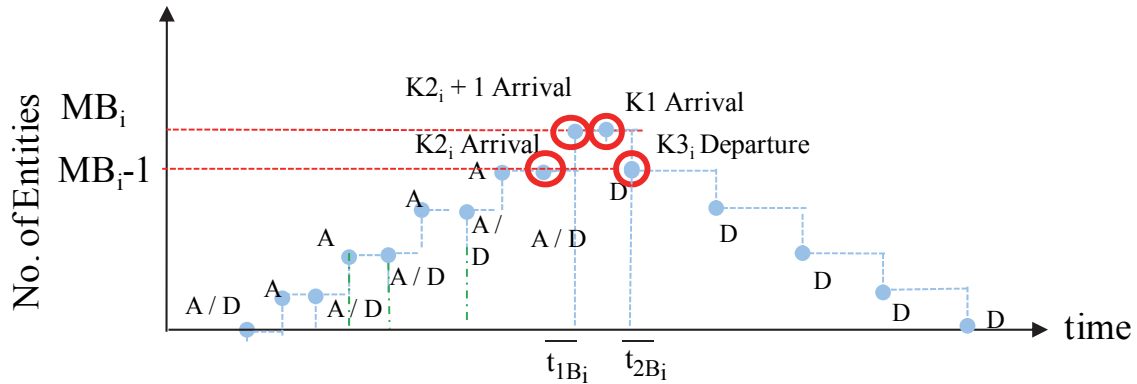


Fig. 6. Illustration of $K2_i$, $K3_i$, \bar{t}_{1B_i} , \bar{t}_{2B_i} and $MB_i - 1$

Notations:

- N7) W_j is a set of one or more buffers referred to as a buffer cluster.
- N8) BB_j is the maximum number of entities a buffer cluster W_j must be able to hold to ensure that no overflows or underflows occur in the buffer cluster.
- N9) X_j is a binary variable $\{0,1\}$ and defines whether cluster W_j must be realized $\{1\}$ or not $\{0\}$ (i.e. $X_j = 1$ determines that cluster W_j must be selected as a part of the buffer cluster policy).
- N10) $K2_i$ is the last arrival to buffer B_i that occurs when the number of entities in buffer B_i is $MB_i - 1$
- N11) $K3_i$ is the number of entities that depart from buffer B_i changing the number of entities in buffer B_i to $MB_i - 1$ from MB_i . This occurs right after the last arrival ($K1$ arrival) to buffer B_i .
- N12) \bar{t}_{1B_i} is the time of the first arrival changing the number of entities in buffer B_i to MB_i from $MB_i - 1$ ($K2_i + 1$ arrival).
- N13) \bar{t}_{2B_i} is the time of the first departure ($K3_i$) departing after the last arrival ($K1$ arrival) for buffer B_i . It is the last time the number of entities in buffer B_i equals MB_i .
- N14) q is the time the last entity departs from server S_N to buffer B_{N+1}
- N15) H is the size each entity occupies within a cell of a buffer in square meters
- N16) G is the maximum size of a buffer cluster W_j in square meters

Assumptions: (Assumptions A1 through A8 from part A hold)

- A9) Possible combinations (clusters) of buffers are given.
- A10) A buffer B_i must be either a dedicated buffer or in a single cluster; thus, if $\{2, 3, \dots, i, \dots, N\}$ denotes the index set of intermediate buffers B_i and $\{1, 2, \dots, j, \dots, C\}$ the index set of buffer clusters W_j , then

$$\bigcup_{j=1}^C W_j = \{2, 3, \dots, N\}$$

Although a buffer cluster must maintain the sequence of operations, meaning it must facilitate an entity to move in the sequence of operations from servers $S_1, S_2, S_3 \dots$ to S_N the buffers included in a cluster should not necessarily be sequential. Therefore, a buffer cluster may include non-sequential buffers and still maintain the sequence of operations. The example in Fig. 7 shows non-sequential buffers B_2 and B_5 clustered (W_1) and sequential buffers B_3 and B_4 clustered (W_2) while still maintaining the sequence of operations S_1 through S_5 as shown by the arrows.

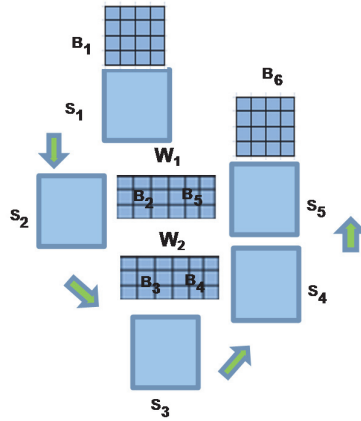


Fig. 7. Non-sequential and sequential buffer clusters maintaining sequence of operations

Eq. (8) in section 3.1 provides us with the maximum number of entities that buffer B_i will experience over a given demand $K1$ within a production shift and is used for sizing the dedicated buffers to ensure that no overflows or underflows occur. The size required for the buffer cluster combinations is needed at any given time. Thus, for the buffer cluster, the buffer sizes leveraging Eq. (11) are assessed at every given time t as shown in Eq. (12) from the time of first arrival (τ_1) of an entity in buffer B_2 to the completion of the production shift (q) (See assumption A6).

$$B_i(t) = \min \left(K1, 1 + \left\lfloor \frac{t - \tau_{i-1}}{MT_{i-1}} \right\rfloor \right) - \min \left(K1, 1 - \left\lfloor \frac{t - \tau_{i-1}}{MT_i} \right\rfloor \right) \quad \forall t \in [\tau_{i-1}, q] \quad (12)$$

From Eq. (12) the search for the buffer cluster size requires a calculation at each time step throughout the production shift for every buffer B_i . This results in a significant number of computations. Appendix B quantifies how the number of calculations can be substantially reduced. Before quantifying the savings, the authors solve for BB_j . To solve for BB_j , first solve for \bar{t}_{1Bi} and \bar{t}_{2Bi} (see Fig. 6) and then prove that the maximum buffer cluster size required occurs when one of the individual buffers B_i is at a maximum during the time interval $t \in [\bar{t}_{1Bi}, \bar{t}_{2Bi}]$. To solve for \bar{t}_{1Bi} , first solve for the last arrival ($K2_i$) that occurs at $MB_i - 1$. Eq. (8) is modified as shown in Eq. (13).

$$MB_i - 1 = (K2_i - 1) - \lfloor Y_i * (K2_i - 1) \rfloor \quad (13)$$

Then to get the first arrival that occurs at MB_i , solve for the time \bar{t}_{1Bi} that the $K2_i + 1$ arrival arrives to buffer B_i

$$\bar{t}_{1Bi} = \tau_{i-1} + ((K2_i + 1) - 1) * MT_{i-1} \quad (14)$$

To determine \bar{t}_{2Bi} , first solve for the time of the $K1$ th arrival to buffer B_i using Eq. (10) and use Eq. (5) to calculate the number of departures that have occurred by the $K1$ th arrival. Then add one to the departures (named $K3_i$) and calculate the time of this departure (\bar{t}_{2Bi}) using Eq. (15).

$$\bar{t}_{2Bi} = \tau_{i-1} + (K3_i - 1) * MT_i \quad (15)$$

The minimum size for a cluster W_j such that no overflows occur is shown in Eq. (16).

$$BB_j = \max_{t \in \cup_{i=1}^N [\bar{t}_{1Bi}, \bar{t}_{2Bi}]} \left\{ \sum_{B_r \in W_j} B_r(t) \right\} \quad \forall W_j \quad (16)$$

In Appendix A, the authors prove that the buffer cluster size required occurs when at least one of the individual buffers in the cluster W_j is at a maximum according to Eq. (16). Thus the buffer cluster size search in the time domain can be reduced by searching across all buffers only for those time steps when

a buffer is at a maximum. This results in significant computational savings. The computational savings are calculated for the case study discussed in Section 4 and shown in Appendix B.

For a given production line, there can be hundreds of buffer cluster combinations. Consider a collection of candidate buffer clusters W_j , $j = 1, \dots, C$, not necessarily disjoint, for which the minimum storage requirements BB_j have been computed via Eq. (16). Integer programming is used to find the buffer cluster combination that provides the minimum total space occupied by all clusters. From here set the objective function as shown in Eq. (17) to determine the buffer cluster(s) size across the production line where $X_j \in \{0, 1\}$ is a decision variable that determines whether the buffer cluster W_j should be realized or not. As defined earlier each buffer can only participate in one and only one realized buffer cluster. The first set of constraints Eq. (18) show that a buffer B_i can only participate in one combination buffer cluster. The second set of constraints Eq. (19) is the maximum size of a buffer cluster in square meters. The third set of constraints Eq. (20) is the binary constraints for X_j . V_j is defined to ensure constraint Eq. (18) applies only to buffers within cluster W_j .

$$V_j = \{ i, B_i \in W_j \} \quad 1 \leq j \leq C$$

Objective Function:

$$\min \sum_{j=1}^C BB_j X_j \quad (17)$$

subject to:

$$\sum_{j:i \in V_j} X_j = 1, \quad 1 \leq i \leq N \quad (18)$$

$$BB_j * H * X_j \leq G, \quad 1 \leq j \leq C \quad (19)$$

$$X_j = 0 \text{ or } 1 \quad \forall j \quad (20)$$

4. Applying model to the industry example

As discussed in the Introduction, the underlying motivation for this research was a case study where a manufacturing facility that produces mobile devices wished to change over from a serial line to a buffer cluster configuration. Table 1 shows the server stations and process times. The footprint for buffer cell holding an entity is 0.005m^2 . The maximum buffer cluster size is 1.825m^2 .

Table 1

Production line processes

Element	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Process Times (Seconds)	1	2	4	5	14	10	19

Given B_1 and B_8 are the starting and ending inventory buffers, these are not included in the analysis. Eq. (8) and Eq. (16) are used to populate Table 2 with W_j cluster sets and size for buffers B_2 through B_7 . The production line shift is 8 hours and 458 units are projected to ship by the end of the shift. There is a space constraint of 1.825 square meters for the buffer cluster size. In Table 2, the cells with bold text indicate that the cluster does not meet the space constraint (i.e. the buffer cluster size $BB_j > 365$ entities). The objective function according to Eq. (17) and constraints according to Eqs. (18 - 20) are below. As discussed in Section 3.2, a buffer cannot participate in multiple clusters at the same time. Table 3 shows the buffer storage space savings for the top buffer cluster configurations considered by the manufacturing center compared to that of the cluster that does not consider the space constraint and that for dedicated buffers. The manufacturing center desired to leverage the cluster for work cells to minimize the buffer storage space. The top buffer cluster configurations considered by the manufacturing center contained buffer clusters with three or four buffers clustered together.

Table 2

W_j Buffer cluster sets and BB_j values for each buffer cluster set

$W_1 = \{B_2\}$ $BB_1 = 229$	$W_2 = \{B_3\}$ $BB_2 = 229$	$W_3 = \{B_4\}$ $BB_3 = 92$	$W_4 = \{B_5\}$ $BB_4 = 294$
$W_5 = \{B_6\}$ $BB_5 = 1$ (Transport)	$W_6 = \{B_7\}$ $BB_6 = 121$	$W_7 = \{B_2, B_3\}$ $BB_7 = 343$	$W_8 = \{B_2, B_4\}$ $BB_8 = 251$
$W_9 = \{B_2, B_5\}$ $BB_9 = 294$	$W_{10} = \{B_2, B_6\}$ $BB_{10} = 230$	$W_{11} = \{B_2, B_7\}$ $BB_{11} = 237$	$W_{12} = \{B_3, B_4\}$ $BB_{12} = 274$
$W_{13} = \{B_3, B_5\}$ $BB_{13} = 346$	$W_{14} = \{B_3, B_6\}$ $BB_{14} = 230$	$W_{15} = \{B_3, B_7\}$ $BB_{15} = 246$	$W_{16} = \{B_4, B_5\}$ $BB_{16} = 326$
$W_{17} = \{B_4, B_6\}$ $BB_{17} = 93$	$W_{18} = \{B_4, B_7\}$ $BB_{18} = 126$	$W_{19} = \{B_5, B_6\}$ $BB_{19} = 295$	$W_{20} = \{B_5, B_7\}$ $BB_{20} = 337$
$W_{21} = \{B_6, B_7\}$ $BB_{21} = 122$	$W_{22} = \{B_2, B_3, B_4\}$ $BB_{22} = 365$	$W_{23} = \{B_2, B_3, B_5\}$ $BB_{23} = 401$	$W_{24} = \{B_2, B_3, B_6\}$ $BB_{24} = 344$
$W_{25} = \{B_2, B_3, B_7\}$ $BB_{25} = 351$	$W_{26} = \{B_2, B_4, B_5\}$ $BB_{26} = 326$	$W_{27} = \{B_2, B_4, B_6\}$ $BB_{27} = 252$	$W_{28} = \{B_2, B_4, B_7\}$ $BB_{28} = 259$
$W_{29} = \{B_2, B_5, B_6\}$ $BB_{29} = 295$	$W_{30} = \{B_2, B_5, B_7\}$ $BB_{30} = 337$	$W_{31} = \{B_2, B_6, B_7\}$ $BB_{31} = 238$	$W_{32} = \{B_3, B_4, B_5\}$ $BB_{32} = 391$
$W_{33} = \{B_3, B_4, B_6\}$ $BB_{33} = 275$	$W_{34} = \{B_3, B_4, B_7\}$ $BB_{34} = 291$	$W_{35} = \{B_3, B_5, B_6\}$ $BB_{35} = 347$	$W_{36} = \{B_3, B_5, B_7\}$ $BB_{36} = 363$
$W_{37} = \{B_3, B_6, B_7\}$ $BB_{37} = 247$	$W_{38} = \{B_4, B_5, B_6\}$ $BB_{38} = 327$	$W_{39} = \{B_4, B_5, B_7\}$ $BB_{39} = 360$	$W_{40} = \{B_4, B_6, B_7\}$ $BB_{40} = 127$
$W_{41} = \{B_5, B_6, B_7\}$ $BB_{41} = 338$	$W_{42} = \{B_2, B_3, B_4, B_5\}$ $BB_{42} = 423$	$W_{43} = \{B_2, B_3, B_4, B_6\}$ $BB_{43} = 366$	$W_{44} = \{B_2, B_3, B_4, B_7\}$ $BB_{44} = 373$
$W_{45} = \{B_2, B_3, B_5, B_6\}$ $BB_{45} = 402$	$W_{46} = \{B_2, B_3, B_5, B_7\}$ $BB_{46} = 409$	$W_{47} = \{B_2, B_3, B_6, B_7\}$ $BB_{47} = 352$	$W_{48} = \{B_2, B_4, B_5, B_6\}$ $BB_{48} = 327$
$W_{49} = \{B_2, B_4, B_5, B_7\}$ $BB_{49} = 360$	$W_{50} = \{B_2, B_4, B_6, B_7\}$ $BB_{50} = 260$	$W_{51} = \{B_2, B_5, B_6, B_7\}$ $BB_{51} = 338$	$W_{52} = \{B_3, B_4, B_5, B_6\}$ $BB_{52} = 392$
$W_{53} = \{B_3, B_4, B_5, B_7\}$ $BB_{53} = 408$	$W_{54} = \{B_3, B_4, B_6, B_7\}$ $BB_{54} = 292$	$W_{55} = \{B_3, B_5, B_6, B_7\}$ $BB_{55} = 364$	$W_{56} = \{B_4, B_5, B_6, B_7\}$ $BB_{56} = 361$
$W_{57} = \{B_2, B_3, B_4, B_5, B_6\}$ $BB_{57} = 424$	$W_{58} = \{B_2, B_3, B_4, B_5, B_7\}$ $BB_{58} = 431$	$W_{59} = \{B_2, B_3, B_4, B_6, B_7\}$ $BB_{59} = 374$	$W_{60} = \{B_2, B_3, B_5, B_6, B_7\}$ $BB_{60} = 410$
$W_{61} = \{B_2, B_4, B_5, B_6, B_7\}$ $BB_{61} = 361$	$W_{62} = \{B_3, B_4, B_5, B_6, B_7\}$ $BB_{62} = 409$	$W_{63} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $BB_{63} = 432$	

Objective Function

$$\min \sum_{i=1}^{63} BB_i * X_i$$

subject to:

Constraint for B2:

$$X_1 + X_7 + X_8 + X_9 + X_{10} + X_{11} + X_{22} + X_{23} + X_{24} + X_{25} + X_{26} + X_{27} + X_{28} + X_{29} + X_{30} + X_{31} + X_{42} + X_{43} + X_{44} + X_{45} + X_{46} + X_{47} + X_{48} + X_{49} + X_{50} + X_{51} + X_{57} + X_{58} + X_{59} + X_{60} + X_{61} + X_{63} = 1$$

Constraint for B3:

$$X_2 + X_7 + X_{12} + X_{13} + X_{14} + X_{15} + X_{22} + X_{23} + X_{24} + X_{25} + X_{32} + X_{33} + X_{34} + X_{35} + X_{36} + X_{37} + X_{42} + X_{43} + X_{44} + X_{45} + X_{46} + X_{47} + X_{52} + X_{53} + X_{54} + X_{55} + X_{57} + X_{58} + X_{59} + X_{60} + X_{62} + X_{63} = 1$$

Constraint for B4:

$$X_3 + X_8 + X_{12} + X_{16} + X_{17} + X_{18} + X_{22} + X_{26} + X_{27} + X_{28} + X_{32} + X_{33} + X_{34} + X_{38} + X_{39} + X_{40} + X_{42} + X_{43} + X_{44} + X_{48} + X_{49} + X_{50} + X_{52} + X_{53} + X_{54} + X_{56} + X_{57} + X_{58} + X_{59} + X_{61} + X_{62} + X_{63} = 1$$

Constraint for B₅:

$$X_4+X_9+X_{13}+X_{16}+X_{19}+X_{20}+X_{23}+X_{26}+X_{29}+X_{30}+X_{32}+X_{35}+X_{36}+X_{38}+X_{39}+X_{41}+X_{42}+X_{45}+X_{46}+X_{48}+X_{49}+X_{51}+X_{52}+X_{53}+X_{55}+X_{56}+X_{57}+X_{58}+X_{60}+X_{61}+X_{62}+X_{63}=1$$

Constraint for B₆:

$$X_5+X_{10}+X_{14}+X_{17}+X_{19}+X_{21}+X_{24}+X_{27}+X_{29}+X_{31}+X_{33}+X_{35}+X_{37}+X_{38}+X_{40}+X_{41}+X_{43}+X_{45}+X_{47}+X_{48}+X_{50}+X_{51}+X_{52}+X_{54}+X_{55}+X_{56}+X_{57}+X_{59}+X_{60}+X_{61}+X_{62}+X_{63}=1$$

Constraint for B₇:

$$X_6+X_{11}+X_{15}+X_{18}+X_{20}+X_{21}+X_{25}+X_{28}+X_{30}+X_{31}+X_{34}+X_{36}+X_{37}+X_{39}+X_{40}+X_{41}+X_{44}+X_{46}+X_{47}+X_{49}+X_{50}+X_{51}+X_{53}+X_{54}+X_{55}+X_{56}+X_{58}+X_{59}+X_{60}+X_{61}+X_{62}+X_{63}=1$$

Space Constraints $BB_j * 0.005 * X_j \leq 1.825, 1 \leq j \leq 63$

Binary Constraint: $X_j = 0$ or 1

Fig. 8 (part a) is the original production line configuration. Batches of 80 come from inventory and enter the production line (B₁). Batches of 80 are put on a pallet and shipped (B₈). The top cluster configurations considered by the facility based on buffer storage savings were then entered into a facility layout tool. The configuration shown in Fig. 8 (part b) was selected {B₃, B₄, B₇}, {B₂, B₅, B₆} resulting in a 39.3% buffer storage savings (1.9 square meters).

Table 3
Buffer cluster sets and buffer storage savings

Cluster Set	Size (No. Entities)	Size (m ²)	Buffer Storage Space Savings (m ²)	Space Savings %
{B ₂ }, {B ₃ }, {B ₄ }, {B ₅ }, {B ₆ }, {B ₇ }*	966	4.83		
{B ₂ , B ₃ , B ₄ , B ₅ , B ₆ , B ₇ }**	432	2.16	2.67	55.3%
{B ₂ , B ₄ , B ₅ , B ₆ , B ₇ }, {B ₃ }	590	2.95	1.88	38.9%
{B ₂ , B ₅ , B ₆ , B ₇ }, {B ₃ , B ₄ }	612	3.06	1.77	36.7%
{B ₃ , B ₅ , B ₆ , B ₇ }, {B ₂ , B ₄ }	615	3.08	1.76	36.3%
{B ₃ , B ₄ , B ₆ , B ₇ }, {B ₂ , B ₅ }	586	2.93	1.90	39.3%
{B ₂ , B ₄ , B ₅ , B ₆ }, {B ₃ , B ₇ }	573	2.87	1.97	40.7%
{B ₂ , B ₄ , B ₆ , B ₇ }, {B ₃ , B ₅ }	606	3.03	1.80	37.3%
{B ₃ , B ₅ , B ₇ }, {B ₂ , B ₄ , B ₆ }	615	3.08	1.76	36.3%
{B ₃ , B ₅ , B ₇ }, {B ₆ }, {B ₂ , B ₄ }	615	3.08	1.76	36.3%
{B ₃ , B ₄ , B ₇ }, {B ₂ , B ₅ , B ₆ }	586	2.93	1.90	39.3%
{B ₃ , B ₄ , B ₇ }, {B ₆ }, {B ₂ , B ₅ }	586	2.93	1.90	39.3%
{B ₃ , B ₄ , B ₆ }, {B ₂ , B ₅ , B ₇ }	612	3.06	1.77	36.6%
{B ₂ , B ₅ , B ₇ }, {B ₆ }, {B ₃ , B ₄ }	612	3.06	1.77	36.6%
{B ₂ , B ₄ , B ₅ }, {B ₆ }, {B ₃ , B ₇ }	573	2.87	1.97	40.7%

*Dedicated Buffers

** Optimal Buffer Cluster without space constraints

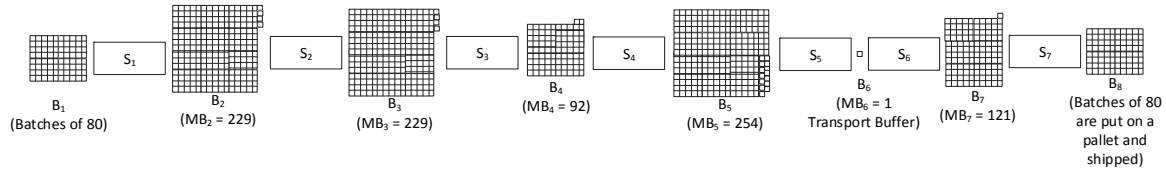


Fig. 8(a). Serial production line

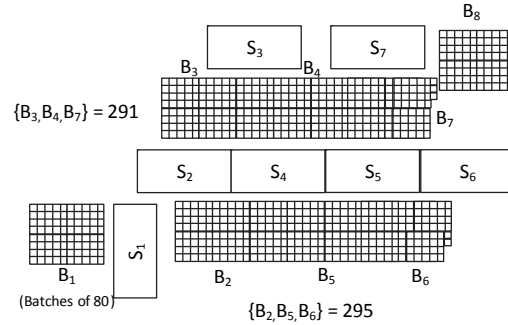


Fig. 8(b). Production line with buffer clusters

As discussed in the introduction, Eq. (16) along with the objective function and constraints in Eqs. (18-20) can be used to conduct sensitivity analysis of the buffer cluster size by varying parameters such as server process time T_i and production demand $K1$.

Table 4

W_j buffer cluster sets and BB_j values for each buffer cluster set ($T_2 = 3s$)

$W_1 = \{B_1\}$ $BB_1 = 305$	$W_2 = \{B_3\}$ $BB_2 = 115$	$W_3 = \{B_4\}$ $BB_3 = 92$	$W_4 = \{B_5\}$ $BB_4 = 294$
$W_5 = \{B_6\}$ $BB_5 = 1$ (Transport)	$W_6 = \{B_7\}$ $BB_6 = 121$	$W_7 = \{B_2, B_3\}$ $BB_7 = 343$	$W_8 = \{B_2, B_4\}$ $BB_8 = 327$
$W_9 = \{B_2, B_5\}$ $BB_9 = 363$	$W_{10} = \{B_2, B_6\}$ $BB_{10} = 306$	$W_{11} = \{B_2, B_7\}$ $BB_{11} = 313$	$W_{12} = \{B_3, B_4\}$ $BB_{12} = 183$
$W_{13} = \{B_3, B_5\}$ $BB_{13} = 294$	$W_{14} = \{B_3, B_6\}$ $BB_{14} = 116$	$W_{15} = \{B_3, B_7\}$ $BB_{15} = 140$	$W_{16} = \{B_4, B_5\}$ $BB_{16} = 326$
$W_{17} = \{B_4, B_6\}$ $BB_{17} = 93$	$W_{18} = \{B_4, B_7\}$ $BB_{18} = 126$	$W_{19} = \{B_5, B_6\}$ $BB_{19} = 295$	$W_{20} = \{B_5, B_7\}$ $BB_{20} = 337$
$W_{21} = \{B_5, B_7\}$ $BB_{21} = 122$	$W_{22} = \{B_2, B_3, B_4\}$ $BB_{22} = 365$	$W_{23} = \{B_2, B_3, B_5\}$ $BB_{23} = 401$	$W_{24} = \{B_2, B_3, B_6\}$ $BB_{24} = 344$
$W_{25} = \{B_2, B_3, B_7\}$ $BB_{25} = 351$	$W_{26} = \{B_2, B_4, B_5\}$ $BB_{26} = 385$	$W_{27} = \{B_2, B_4, B_6\}$ $BB_{27} = 328$	$W_{28} = \{B_2, B_4, B_7\}$ $BB_{28} = 335$
$W_{29} = \{B_2, B_5, B_6\}$ $BB_{29} = 364$	$W_{30} = \{B_2, B_5, B_7\}$ $BB_{30} = 371$	$W_{31} = \{B_2, B_6, B_7\}$ $BB_{31} = 314$	$*W_{32} = \{B_2, B_3, B_4, B_5\}$ $*BB_{32} = 358$
$W_{33} = \{B_2, B_3, B_4, B_6\}$ $BB_{33} = 184$	$W_{34} = \{B_2, B_3, B_4, B_7\}$ $BB_{34} = 208$	$W_{35} = \{B_2, B_3, B_5, B_6\}$ $BB_{35} = 295$	$W_{36} = \{B_2, B_3, B_5, B_7\}$ $BB_{36} = 337$
$W_{37} = \{B_2, B_3, B_6, B_7\}$ $BB_{37} = 141$	$W_{38} = \{B_2, B_4, B_5, B_6\}$ $BB_{38} = 327$	$W_{39} = \{B_2, B_4, B_5, B_7\}$ $BB_{39} = 360$	$W_{40} = \{B_2, B_4, B_6, B_7\}$ $BB_{40} = 127$
$W_{41} = \{B_2, B_5, B_6, B_7\}$ $BB_{41} = 338$	$W_{42} = \{B_2, B_3, B_4, B_5\}$ $BB_{42} = 423$	$W_{43} = \{B_2, B_3, B_4, B_6\}$ $BB_{43} = 366$	$W_{44} = \{B_2, B_3, B_4, B_7\}$ $BB_{44} = 373$
$W_{45} = \{B_2, B_3, B_5, B_6\}$ $BB_{45} = 402$	$W_{46} = \{B_2, B_3, B_5, B_7\}$ $BB_{46} = 409$	$W_{47} = \{B_2, B_3, B_6, B_7\}$ $BB_{47} = 352$	$W_{48} = \{B_2, B_4, B_5, B_6\}$ $BB_{48} = 386$
$W_{49} = \{B_2, B_4, B_5, B_6, B_7\}$ $BB_{49} = 393$	$W_{50} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $BB_{50} = 336$	$W_{51} = \{B_2, B_4, B_5, B_6, B_7\}$ $BB_{51} = 372$	$*W_{52} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $*BB_{52} = 359$
$W_{53} = \{B_3, B_4, B_5, B_6, B_7\}$ $BB_{53} = 384$	$W_{54} = \{B_3, B_4, B_5, B_6, B_7\}$ $BB_{54} = 209$	$W_{55} = \{B_3, B_5, B_6, B_7\}$ $BB_{55} = 338$	$W_{56} = \{B_4, B_5, B_6, B_7\}$ $BB_{56} = 361$
$W_{57} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $BB_{57} = 424$	$W_{58} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $BB_{58} = 431$	$W_{59} = \{B_2, B_3, B_4, B_6, B_7\}$ $BB_{59} = 374$	$W_{60} = \{B_2, B_3, B_5, B_6, B_7\}$ $BB_{60} = 410$
$W_{61} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $BB_{61} = 394$	$W_{62} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $BB_{62} = 385$	$W_{63} = \{B_2, B_3, B_4, B_5, B_6, B_7\}$ $BB_{63} = 432$	

Now the authors leverage the framework of the model and vary the process time of one server to demonstrate how the model can be used for sensitivity analysis. In this case, authors vary the process time of server S_2 to three seconds, calculate the BB_j values for each cluster set and show the buffer cluster sets in Table 4. As before, the cells with bold text indicate that the cluster does not meet the space constraint and are the same cells that did not meet the space constraint in Table 2. If the cells are in italicized text, they used to meet the space constraint, but due to the change in the process times, no longer meet the constraint. The cells with a “*” indicate that the cluster exceeded space constraint in Table 2, but now meets the constraint in Table 4. The BB_j values in red text indicate a change in the size of the cluster from Table 2. Now the authors take the configurations from Table 3 and identify in Table 5 that there are configurations that now, with S_2 equaling 3 seconds do not meet the space constraint (in bold text). It is shown that the configuration selected with a process time S_2 equaling 2 seconds, $\{B_3, B_4, B_7\}$, $\{B_2, B_5, B_6\}$, with 586 entities, achieves a total buffer size of 572 entities when the process time of S_2 is 3 seconds. This scenario in italicized text in Table 5. So the initial buffer cluster set shown in Fig. 8 can remain and still satisfy the space constraints when the process time of S_2 varies from two to three seconds.

Table 5
Buffer cluster sets and buffer storage savings with T_2 at 2 and 3 seconds

Cluster Set	Size (No. Entities) $T_2=2s$	Size (No. Entities) $T_2=3s$	Size (m^2) $T_2=2s$	Size (m^2) $T_2=3s$	Buffer Storage Space Savings (m^2) $T_2=2s$	Buffer Storage Space Savings (m^2) $T_2=3s$	Space Savings % $T_2=2s$	Space Savings % $T_2=3s$
$\{B_2\}, \{B_3\}, \{B_4\}, \{B_5\}, \{B_6\}, \{B_7\}$ *	966	928	4.83	4.64				
$\{B_2, B_3, B_4, B_5, B_6, B_7\}$ **	432	432	2.16	2.16	2.67	2.48	55.3	53.5
$\{B_2, B_4, B_5, B_6, B_7\}, \{B_3\}$	590	509	2.95	2.55	1.88	2.10	38.9	45.2
$\{B_2, B_5, B_6, B_7\}, \{B_3, B_4\}$	612	555	3.06	2.78	1.77	1.87	36.7	40.2
$\{B_3, B_5, B_6, B_7\}, \{B_2, B_4\}$	615	665	3.08	3.33	1.76	1.32	36.3	28.3
$\{B_3, B_4, B_6, B_7\}, \{B_2, B_5\}$	586	572	2.93	2.86	1.90	1.78	39.3	38.4
$\{B_2, B_4, B_5, B_6\}, \{B_3, B_7\}$	573	526	2.87	2.63	1.97	2.01	40.7	43.3
$\{B_2, B_4, B_6, B_7\}, \{B_3, B_5\}$	606	630	3.03	3.15	1.80	1.49	37.3	32.1
$\{B_3, B_5, B_7\}, \{B_2, B_4, B_6\}$	615	665	3.08	3.33	1.76	1.32	36.3	28.3
$\{B_3, B_5, B_7\}, \{B_6\}, \{B_2, B_4\}$	615	665	3.08	3.33	1.76	1.32	36.3	28.3
<i>$\{B_3, B_4, B_7\}, \{B_2, B_5, B_6\}$</i>	<i>586</i>	<i>572</i>	<i>2.93</i>	<i>2.86</i>	<i>1.90</i>	<i>1.78</i>	<i>39.3</i>	<i>38.4</i>
$\{B_3, B_4, B_7\}, \{B_6\}, \{B_2, B_5\}$	586	572	2.93	2.86	1.90	1.78	39.3	38.4
$\{B_3, B_4, B_6\}, \{B_2, B_5, B_7\}$	612	555	3.06	2.78	1.77	1.87	36.6	40.2
$\{B_2, B_5, B_7\}, \{B_6\}, \{B_3, B_4\}$	612	555	3.06	2.78	1.77	1.87	36.6	40.2
$\{B_2, B_4, B_5\}, \{B_6\}, \{B_3, B_7\}$	573	526	2.87	2.63	1.97	2.01	40.7	43.3

*Dedicated Buffers

** Optimal Buffer Cluster without space constraints

5. Conclusion

This study's results suggest that parametric time-dependent exact methods can be derived and applied with accurate results. This study derived and demonstrated usage of a time based parametric model for N-server, N+1-buffer sequential line to assist production environments in sizing buffers, in particular, buffer clusters appropriately when alternate production line configurations are desired. This study derives an optimization framework that enabled a clustering policy and provides output of the required buffer sizing for that policy. The result reduces the buffers storage space and thus the production line footprint when implemented while ensuring no bottlenecks. The research also reduced the buffer and time search space significantly reducing the number of computations. As demonstrated in the case study, the models can be used to conduct sensitivity analysis of the buffer cluster size by varying parameters such as process time or production demand.

Related studies are in progress that relax assumptions of the models in this paper and also expand configurations. In particular, studies in process consider unreliable machines. Another area of study is utilizing the model herein to consider when product size varies throughout the manufacturing process. The ability to extract state space models at any given time of interest is a rich area for Operations Research with several applications in industry.

References

- Aghazadeh, S., Hafeznezami, S., Najjar L., and Huq, Z. (2011). The influence of work-cells and facility layout on the manufacturing efficiency. *Journal of Facilities Management*, 9(3), 213-224.
- Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European journal of operational research*, 168(3), 694-715.
- Chan, F. T. S., & Ng, E. Y. H. (2002). Comparative evaluations of buffer allocation strategies in a serial production line. *The International Journal of Advanced Manufacturing Technology*, 19(11), 789-800.
- Charharsooghi, S.K. and Nahavandi, N. (2003). Buffer allocation problem, a heuristic approach. *Scientia Iranica*, 10(4), 401-409.
- Chow, W. M. (1987). Buffer capacity analysis for sequential production lines with variable process times. *International Journal of Production Research*, 25(8), 1183-1196.
- Enginarlar, E., Li, J., & Meerkov, S. M. (2005). How lean can lean buffers be?. *IIE Transactions*, 37(4), 333-342.
- Enginarlar, E., Li, J., Meerkov, S. M., & Zhang, R. Q. (2002). Buffer capacity for accommodating machine downtime in serial production lines. *International Journal of Production Research*, 40(3), 601-624.
- Gershwin, S. B. (1987). An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations research*, 35(2), 291-305.
- Gershwin, S. B., & Schor, J. E. (2000). Efficient algorithms for buffer space allocation. *Annals of Operations Research*, 93(1-4), 117-144.
- Govil, M. K., & Fu, M. C. (1999). Queueing theory in manufacturing: A survey. *Journal of manufacturing systems*, 18(3), 214.
- Ichikawa, H. (2009, December). Simulating an applied model to optimize cell production and parts supply (Mizusumashi) for laptop assembly. In *Winter Simulation Conference* (pp. 2272-2280). Winter Simulation Conference.
- Kouikoglou, V. S., & Phillis, Y. A. (1991). An exact discrete-event model and control policies for production lines with buffers. *Automatic Control, IEEE Transactions on*, 36(5), 515-527.
- Kouikoglou, V. S., & Phillis, Y. A. (1994). Discrete event modeling and optimization of unreliable production lines with random rates. *Robotics and Automation, IEEE Transactions on*, 10(2), 153-159.
- Kouikoglou, V. S., & Phillis, Y. A. (1995). An efficient discrete-event model for production networks of general geometry. *IIE transactions*, 27(1), 32-42.
- Li, J., E. Blumenfeld, D., Huang, N., & M. Alden, J. (2009). Throughput analysis of production systems: recent advances and future topics. *International Journal of Production Research*, 47(14), 3823-3851.
- Lim, J. T., Meerkov, S. M., & Top, F. (1990). Homogeneous, asymptotically reliable serial production lines: theory and a case study. *Automatic Control, IEEE Transactions on*, 35(5), 524-534..
- Liu, C. M. and Lin, C. L. (1994). Performance evaluation of unbalanced serial production lines. *International Journal of Production Research*, 32(12), 2897-2914.
- Logendran, R., & Karim, Y. (2003). Design of manufacturing cells in the presence of alternative cell locations and material transporters. *Journal of the Operational Research Society*, 54, 1059-1075.
- Matta, A., Runchina, M., & Tolio, T. (2006). Automated flow lines with shared buffer. In *Stochastic Modeling of Manufacturing Systems* (pp. 99-120). Springer Berlin Heidelberg.
- Morrison, J. R. (2010). Deterministic flow lines with applications. *Automation Science and Engineering, IEEE Transactions on*, 7(2), 228-239.

- Ramirez-Serrano, A., & Benhabib, B. (2000). Supervisory control of multiworkcell manufacturing systems with shared resources. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 30(5), 668-683.
- Senanayake, C. D., & Subramaniam, V. (2013). Analysis of a two-stage, flexible production system with unreliable machines, finite buffers and non-negligible setups. *Flexible Services and Manufacturing Journal*, 25(3), 414-442.
- So, K. C. (1989). Allocating buffer storages in a flexible manufacturing system. *International journal of flexible manufacturing systems*, 1(3), 223-237.
- Wei, K. C., Tsao, Q. Q., & Otto, N. C. (1989, December). Estimation of buffer size using stochastic approximation methods. In *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on* (pp. 1066-1068). IEEE.
- Yamashita, H., & Altiok, T. (1998). Buffer capacity allocation for a desired throughput in production lines. *IIE transactions*, 30(10), 883-892.
- Youssef, A. M., & ElMaraghy, H. A. (2007). Optimal configuration selection for reconfigurable manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 19(2), 67-106.

Appendix A. Proof for minimum cluster size

Lemma 1: Minimum size for cluster W_j such that no overflows occur takes place when at least one of the buffers B_i in cluster W_j has reached the maximum number of entities, MB_i .

Take buffers B_k and B_p that are in cluster W_j and output to servers S_k and S_p respectively where $k < p$, and the constraints $MT_{k-1} < MT_k$ and $MT_{p-1} < MT_p$ hold. Recall from Corollary 2 that when $MT_{k-1} = MT_k$ or $MT_{p-1} = MT_p$, the buffer size is 1. For the proof, the authors observe the buffer inventory profiles of B_k and B_p at three specific time intervals of the buffer inventory covering the time from the first arrival to buffer B_k to the last departure from buffer B_p as shown in Fig. 9. In addition, authors also observe when buffers B_k and B_p are sequential ($p = k+1$) and when they are not ($p > k+1$). Fig. 9 shows the case when $p = k+1$. As $p > k+1$, the buffer profiles drift apart and the overlap in Time Interval 3 decreases until no overlap exists.

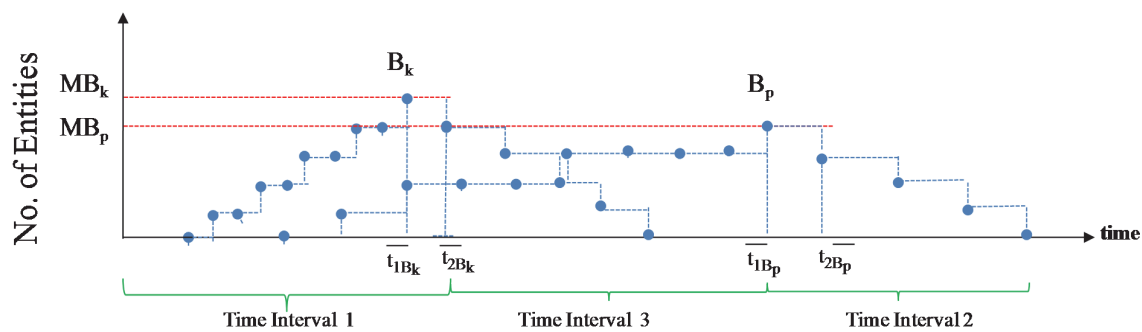


Fig. 9. Buffer profiles of B_k and B_p and time Intervals 1 through 3 for $p = k + 1$

Time Interval 1: $\tau_{k-1} < t \leq \bar{t}_{2Bk}$ (B_k and B_p are increasing; B_k has reached a maximum)

Buffer B_k has its first arrival at τ_{k-1} and increases until it reaches MB_k . Buffer B_p has its first arrival at τ_{p-1} and increases until it reaches its maximum, MB_p . For Buffers B_k and B_p where $p = k+1$, $\tau_{p-1} - \tau_{k-1} = T_k$. Thus, buffer B_p starts increasing T_k seconds after the first arrival to buffer B_k . When $p > k+1$, $\tau_{p-1} - \tau_{k-1}$ is greater than T_k meaning that time of the first arrival of B_p approaches and can exceed time interval $[\bar{t}_{1Bk}, \bar{t}_{2Bk}]$ when B_k is maximum. When buffer B_k is at its maximum (MB_k), buffer B_p is increasing in size, while after reaching MB_k , buffer B_k begins to decline. Therefore, a possible buffer cluster maximum between buffers B_k and B_p is at MB_k .

Time Interval 2: $\bar{t}_{1 B_p} \leq t \leq \tau_{p-1} + (K1-1) * MT_p$ (B_k is decreasing and reaches zero; B_p has reached its maximum and begins to decline including its last departure)

The last arrival to buffer B_p at time $t_{ApK1} = \tau_{p-1} + (K1 - 1) * MT_{p-1}$

The last departure of buffer B_k occurs at time $t_{DkK1} = \tau_{k-1} + (K1 - 1) * MT_k$

Subtract these times to get:

$$\begin{aligned} \tau_{p-1} + (K1 - 1) * MT_{p-1} - \tau_{k-1} - (K1 - 1) * MT_k &= \quad (A.1) \\ \tau_{p-1} - \tau_{k-1} + (K1 - 1) * MT_{p-1} - (K1 - 1) * MT_k &= \\ \tau_{p-1} - \tau_{k-1} - (K1 - 1) * (MT_{p-1} - MT_k) & \end{aligned}$$

For Buffers B_k and B_p where $p = k+1$ then $MT_{p-1} - MT_k = 0$ and $\tau_{p-1} - \tau_{k-1} = T_k$ then (A.1) equals T_k . Therefore, the last departure of B_k occurs T_k seconds prior to the last arrival to buffer B_p . Thus B_k reaches zero while B_p is at a maximum. When $p > k+1$, then $\tau_{p-1} - \tau_{k-1}$ is greater than T_k and $MT_k \leq MT_{p-1}$, therefore $(K1-1)*(MT_{p-1} - MT_k) \geq 0$, resulting in (A.1) being greater than T_k . Therefore, a possible buffer cluster maximum between buffers B_k and B_p occurs at MB_p .

Time Interval 3: $\bar{t}_{2 B_k} < t < \bar{t}_{1 B_p}$ (B_k is decreasing; B_p is increasing but has not reached a maximum)

At time $\bar{t}_{2 B_k}$, B_k has experienced its last arrival (K1) and it has reached a maximum MB_k . Therefore, after this time, only departures occur. Thus in essence, MB_k indicates the number of departures that are left for buffer B_k until it reaches zero. During this time interval, buffer B_p is increasing (it hasn't reached a maximum yet), meaning it has both arrivals and departures. When $p = k+1$, $MT_k = MT_{p-1}$ indicating that the number of departures remaining at buffer B_k , MB_k , is also the number of entities still to arrive at buffer B_p and they occur at the same time. However, given that B_p is increasing and hasn't reached a maximum, it is also experiencing departures at a rate of MT_p . During this time interval, the quantity of inventory of buffer B_k declines from MB_k to zero. Although Buffer B_p entities arrive at the same rate as buffer B_k departures, its inventory increases more slowly than the decline of departures from buffer B_k given buffer B_p also has entities departing at a rate of MT_p during this time interval. Therefore the sum of the inventory profiles of buffer B_k and B_p during this time interval will not exceed the maximum inventory observed in time interval (1) or (2) described above. When $p > k + 1$, buffer B_p has its first entity arriving even later than in the $p = k+1$ case. Although the decline of buffer B_k remains the same, buffer B_p starting arrival approaches the time when buffer B_k approaches MB_k and the summation of the two inventory profiles will not exceed the maximum inventory observed in time interval (1) or (2) described above. Based on results of the analysis for each of the time intervals, the union of time intervals in (A.2) for each buffer must be searched to find the maximum buffer cluster size BB_j .

$$t \in [\bar{t}_{1 B_1}, \bar{t}_{2 B_1}] \cup [\bar{t}_{1 B_2}, \bar{t}_{2 B_2}] \cup \dots [\bar{t}_{1 B_i}, \bar{t}_{2 B_i}] \cup \dots [\bar{t}_{1 B_N}, \bar{t}_{2 B_N}] \quad (A.2)$$

Appendix B. Computational and solution time savings

Table 6 shows that for this case study, 26 critical time steps were identified to measure the buffer size, resulting in 26 calculations. For Buffer B_6 , because $MT_{i-1} > MT_i$, no time interval to detect the maximum buffer size is required because buffer size required is always 1 (as discussed in Corollary 2, this is a transport buffer).

Table 7 shows the average savings in time steps processed and average solution time savings benefits based on number of buffers to cluster in the sequential line. The study starts with 6 buffers similar to the example in Section 4 and then doubles the production line size to 12 buffers and 24 buffers respectively.

K1 or production demand is also varied (from 100 to 300) such that it would cover a production shift interval spread of 8 to 12 hours.

Table 6

Number of time steps for required buffer size computations

Element B_i	B_2	B_3	B_4	B_5	B_6	B_7
\bar{t}_{1B_i}	458	917	1831	2292	0	6434
\bar{t}_{2B_i}	459	918	1836	2307	0	6438
$\bar{t}_{2B_i} - \bar{t}_{1B_i}$	1	1	5	15	0	4
Total: $B_2 - B_7 = 26$						

Table 7

Calculation and computation time savings varying K1

(A) No. Buffers	(B) Ave Time Steps	(C) Ave Solution Time 36000 time steps (sec)	(D) Ave Sum $\bar{t}_{2B_i} - \bar{t}_{1B_i}$ Time Steps	(E) Solution Time $\bar{t}_{2B_i} - \bar{t}_{1B_i}$ Calculations (sec)	(F) Savings in time steps processed $((B)-(D))/(B)\%$	(G) Savings in Solution Time $((C)-(E))/(C)\%$
6	36000	137	422	19	99%	86%
12	36000	227	921	25	97%	89%
24	36000	466	2355	33	93%	93%