IISE Transactions

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uiie21

Decomposition-based real-time control of multi-stage transfer lines with residence time constraints

Feifan Wang & Feng Ju

To cite this article: Feifan Wang & Feng Ju (2021) Decomposition-based real-time control of multi-stage transfer lines with residence time constraints, IISE Transactions, 53:9, 943-959, DOI: 10.1080/24725854.2020.1803513

To link to this article: https://doi.org/10.1080/24725854.2020.1803513

Published online: 21 Sep 2020.

IISE TRANSACTIONS
2021, VOL. 53, NO. 9, 943–959
https://doi.org/10.1080/24725854.2020.1803513

Decomposition-based real-time control of multi-stage transfer lines with residence time constraints

Feifan Wang and Feng Ju

School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

ABSTRACT

It is commonly observed in the food industry, battery production, automotive paint shop, and semiconductor manufacturing that an intermediate product's residence time in the buffer within a production line is controlled by a time window to guarantee product quality. There is typically a minimum time limit reflected by a part's travel time or process requirement. Meanwhile, these intermediate parts are prevented from staying in the buffer for too long by an upper time limit, exceeding which a part will be scrapped or need additional treatment. To increase production throughput and reduce scrap, one needs to control machines' working mode according to real-time system information in the stochastic production environment, which is a difficult problem to solve, due to the system's complexity. In this article, we propose a novel decomposition-based control approach by decomposing a production system into small-scale subsystems based on domain knowledge and their structural relationship. An iterative aggregation procedure is then used to generate a production control policy with convergence guarantee. Numerical studies suggest that the decomposition-based control approach outperforms general-purpose reinforcement learning method by delivering significant system performance improvement and substantial reduction on computation overhead.

ARTICLE HISTORY
Received 4 November 2019
Accepted 15 July 2020

KEYWORDS
Residence time; real-time control; multi-stage transfer line; decomposition-based control

1. Introduction

Rapid advances in sensor technology, automation, and artificial intelligence have the potential to contribute to improving manufacturing efficiency and quality (Yang *et al.*, 2019). This could allow the creation of Smart Manufacturing — enabling production systems to be self-learning and self-optimizing "on the fly" using real-time information to quickly respond to production uncertainties.

One common real-time production control problem is to deal with production systems with residence time constraints for intermediate products. For instance, a semiconductor packaging and testing line consists of multiple operations. During the production run, there are certain operations (e.g., Die Attach), after which a product needs to be held for a minimum time in the buffer for outgassing purposes before being sent to the next operation. In addition, these time windows are typically upper bounded around operations, where exposing parts to the atmosphere for too long will lead to either surface oxidation or moisture absorption, thus reducing production yield (Han and Kim, 2017). Therefore, proper dispatching decisions need to be made considering parts' actual waiting time in buffers.

Similar issues are observed in many industries, including food production lines and battery manufacturing, where perishability of intermediate products is a major concern. For example, yogurt goes through several processes from raw milk to intermediate products and finally to final products, and each stage is performed under strict time limits (Amorim *et al.*, 2013). Quality deterioration could occur during the stoppage due to machine failures (Liberopoulos and Tsarouhas, 2005). Similarly in battery manufacturing, chemicals are filled into cells to form electrodes, and those processes need to be done within a certain time limit to ensure cell quality. A cell will be typically scrapped, if such a time limit is exceeded, which potentially increases production cost and wastes (Ju *et al.*, 2017).

To optimize the system performance while providing a guaranteed product quality, optimal production control and dispatch strategy are pursued to coordinate machines and product flow considering machines' random failures. Due to the complexity of production systems with residence time constraints, current studies primarily focus on small-scale problems with only two machines and one buffer (Ju *et al.*, 2017; Wang *et al.*, 2019). The basic idea is to develop a Markov Decision Process (MDP) model given machine uncertainty and derive the optimal control policy for sequential decision making in each cycle based on the real-time system state. However, it is inapplicable to extend such an approach to multi-stage transfer lines, where the system state space increases exponentially as the system scales up. Reinforcement learning can potentially solve the aforementioned production control problem for large-scale systems (Bertsekas, 2019). Through discrete event simulations, the mapping of states and actions to rewards can be learned by

CONTACT Feng Ju Feng.Ju@asu.edu

approximation architectures, such as regression and artificial neural networks (Bertsekas, 2018). However, a heavy computation overhead impedes reinforcement learning from being practically implemented for real-time applications. In addition, the learning process typically ignores the engineering domain knowledge, which makes the control solution difficult to explain and lack managerial insights.

To overcome these drawbacks, we propose a novel decomposition-based control approach by decomposing a multi-stage production system into small-scale subsystems using domain knowledge and their structural relationship. The subsystems are simple enough to derive a control solution using local information. An iterative aggregation procedure is then used to improve the derived control policy with convergence guarantee. Compared with a general-purpose reinforcement learning-based method, the decomposition-based control can deliver significant improvements on system performance and substantial reduction on computation overhead, which makes it applicable for real-time production decision making.

The rest of this article is organized as follows. Section 2 reviews the related literature. The mathematical formulation is introduced in Section 3. In Section 4, we present the framework of decomposition-based control, which includes the modeling of subsystems and the aggregation procedure. A simulation study is conducted in Section 5 to justify the performance of the decomposition-based control. Finally, the conclusions and future directions are provided in Section 6.

2. Literature review

Perishability is usually a main reason why residence time constraints need to be considered (Raafat, 1991). Different classifications are proposed to deal with perishability (Nahmias, 1982; Raafat, 1991). According to Amorim *et al.* (2013), perishability has three dimensions: physical product deterioration, authority limits, and customer value. In manufacturing environments, the dimension of authority limits is usually applied. A fixed threshold, obtained from experiments or domain knowledge, is set to represent residence time constraints. Perishability only refers to the upper bound of the residence time, and in practice, the lower bound of residence time is also often required in a production system. In a production line, a machine's random failure and repair are often the primary source of uncertainty, and can significantly impact system performance. Transfer line refers to a category of production systems, where system performance is studied under the uncertainty of machine reliability (Gershwin, 1994; Li and Meerkov, 2009; Papadopoulos *et al.*, 2019). Transfer lines with residence time constraints are studied. One direction of the study is to estimate the probability distribution of the residence time. Shi and Gershwin (2012) study the distribution of the residence time of parts in the buffer for a two-machine transfer line, and the risk of scrap is evaluated based on the derived distribution. Such a residence time, especially counting from part entry to the system to the departure from the system, is often referred to as lead time or sojourn time in the literature. For instance, Shi and Gershwin (2015) and Angius *et al.* (2016) consider lead time in a three-machine transfer line and a production system with closed loop, respectively. Shi and Gershwin (2016) extend the study on residence time distribution to transfer lines with multiple machines and obtain residence time distribution for each buffer. The studies in this direction help design buffer capacity to reduce a rate of defective parts. However, a transfer line in those studies can only identify a defective part at the end of the transfer line. It wastes resources to process a defective part, before its defect is identified. In Naebulharam and Zhang (2014), the defect is able to be detected soon after the defect is created, and quality buy rate is introduced to evaluate system performance. Lee *et al.* (2017) and Lee *et al.* (2018) consider Bernoulli lines where each machine inspects the quality of parts, and parts with a residence time larger than a limit have a certain probability of being scrapped. Another direction of the study is to take residence time into consideration in the modeling (Wang and Ju, 2020). System dynamics can be more accurately captured in this way, and it enables production control based on the residence time. However, this modeling results in a large state space to be addressed in system analysis (Ju *et al.*, 2015; Kang *et al.*, 2016).

The above-mentioned studies aim at deriving analytical performance measures of a transfer line given a system parameter setting. Performance measures include production rate and scrap rate. One way to improve the system is to stop several machines from producing each cycle according to real-time system states to reduce scrap rate without sacrificing too much of the production rate. Thus, a control problem arises and becomes worth studying. Real-time control of a two-machine transfer line is studied (Ju *et al.*, 2017; Wang *et al.*, 2019). After problem approximation, the optimal control policy is derived. Such a method works well for a small-scale problem, but is not directly extendable.

Production control is studied to achieve a desired system performance, and it has been investigated for decades. Due to a lack of access to real-time system states, early practice of production control mainly focuses on simple static system settings (Jaegler *et al.*, 2018) and event-driven/rule-based approaches (Thürer *et al.*, 2019), and those strategies are still widely used (Cao and Xie, 2015; Ju *et al.*, 2016). Supported by the Internet of Things technologies, real-time production control based on real-time system state becomes possible, and it provides potentials to further improve a production system (Jia *et al.*, 2016; Lu *et al.*, 2016). Reinforcement learning is a way to perform real-time production control and enhance production performance (Stricker *et al.*, 2018; Waschneck *et al.*, 2018). Through training, reinforcement learning enables a complex production system to find a real-time action that can improve its performance. However, such a way to control production becomes difficult in many cases for two reasons. First, training a reinforcement learning model is computationally expensive. Second, learning methods, such as artificial neural networks, are black box models, and it is difficult to combine domain knowledge

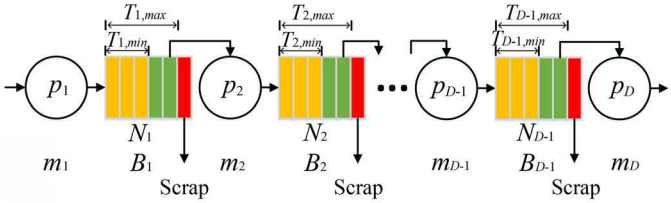


Figure 1. Illustration of the multi-stage line with residence time constraints.

into them. Another direction to control a complex production system is through a decentralized way (Lu and Ju, 2017; Wang *et al.*, 2017; Wang *et al.*, 2018). The introduction of the multi-agent system and holonic manufacturing system attempts to address the production control problem in this way (Leitão, 2009; Barbosa *et al.*, 2015; Giret *et al.*, 2016). The decentralized control aims at achieving flexible control, significantly reducing computational efforts, and improving system performance globally. However, it is difficult to have all three objectives well achieved, and there is a lack of mathematical models supporting decentralized production control, which motivates the work outlined in this article.

3. Problem formulation

3.1. System description and assumptions

For simplicity purposes, the term "multi-stage line" is used to represent the multi-stage Bernoulli transfer line with residence time constraints for the rest of this article. The multi-stage line under study is shown in Figure 1. Parts visit each machine and buffer from the left side to the right side, until they finish all the processes or get scrapped from the system. The following assumptions define the machines, the buffers, and their interactions:

(i) The multi-stage line consists of D machines, denoted by $m_1, m_2, ..., m_D$, and $(D-1)$ buffers, denoted by $B_1, B_2, ..., B_{D-1}$, where $D > 2$.

(ii) All machines are synchronized with a constant processing time (cycle time), which is the time to process a single part on a machine.

(iii) Machines are subject to failures, and each machine is assumed to be an independent Bernoulli machine. The state of a machine is determined at the beginning of a cycle. Before that, the state of machine m_i in cycle t, for $i = 1, ..., D$ and $t = 1, 2, ...$, is a random variable, denoted by $S_i(t)$, following the Bernoulli distribution with parameter p_i. Specifically, machine m_i is capable of producing a part in cycle t with probability p_i and fails to do so with probability $(1 - p_i)$. It can be represented by $P(S_i(t) = 1) = p_i$ and $P(S_i(t) = 0) = 1 - p_i$. At the beginning of cycle t, the machine state is realized, and the realized machine state is denoted by $s_i(t) \in \{0, 1\}$.

(iv) Buffer B_i has a finite capacity N_i $(1 \le N_i < \infty)$, for $i = 1, 2, ..., D-1$, and its buffer occupancy is determined at the end of a cycle and denoted by n_i. First-in-first-out policy is assumed regarding the buffer outflow process.

(v) Each part in a buffer has its residence time, and it is counted as the number of cycles, for which the part has been staying in the buffer. Residence time of a part is determined at the end of a cycle and starts at zero as the part enters a buffer at the end of a cycle. Residence time of a part in a buffer increases by one each cycle, if the part remains in the same buffer. Let $\tau_{i,j}$ denote the residence time of the jth part in buffer B_i, if such a part exists.

(vi) The maximum allowable residence time for a part in buffer B_i is characterized by $T_{i,max}$, for $i = 1, 2, ..., D-1$. A part in buffer B_i will be scrapped when its residence time reaches $T_{i,max}$. Let $T_{i,max} \ge N_i$, otherwise N_i has no effect on the multi-stage line.

(vii) The minimum required residence time for a part in buffer B_i is denoted by $T_{i,min}$, for $i = 1, 2, ..., D-1$. A part in buffer B_i is allowed to be processed by machine m_{i+1} only when its residence time reaches or exceeds $T_{i,min}$.

(viii) Machine m_i, for $i = 1, 2, ..., D-1$, is blocked during a cycle, if (a) machine m_i is up, (b) buffer B_i is full, (c) machine m_{i+1} does not produce a part in this cycle due to machine failure or blockage, and (d) there will be no part scrapped from buffer B_i. Machine m_D is never blocked. In addition, the block-before-service policy is assumed.

(ix) Machine m_i, for $i = 2, ..., D$, is starved during a cycle, if machine m_i is up, and no part in buffer B_{i-1} has residence time greater than or equal to $T_{i-1,min}$. Machine m_1 is never starved.

(x) At the end of each cycle, a machine can be stopped to prevent it from producing in the next cycle. One can also have a machine unchanged, and thus the machine will work as a Bernoulli machine in the next cycle. It is always beneficial not to stop the last machine, so we only consider actions on machine m_i, for $i = 1, 2, ..., D-1$. Let $a_i(t) \in \{1, 0\}$, for $i = 1, 2, ..., D-1$ and $t = 0, 1, ...$, denote the action on machine m_i at the end of cycle t. The action $a_i(t) = 0$ makes machine m_i not work in cycle $(t+1)$. The action $a_i(t) = 1$ represents that machine m_i is unchanged. The action on the whole system is represented by $\boldsymbol{a}(t) = [a_1(t) \quad a_2(t) \quad \cdots \quad a_{D-1}(t)]^T$. The action space is denoted by $\mathcal{A} = \{0, 1\}^{D-1}$.

3.2. Performance measures

To evaluate the multi-stage line, we introduce the performance measures of interest as follows.

- *Production rate of machine m_i*, $PR_i(t)$, for $t = 1, 2, \ldots$ and $i = 1, \ldots, D$: the expected number of parts produced by machine m_i in cycle t;
- *Scrap rate of buffer B_i*, $SR_i(t)$, for $t = 1, 2, \ldots$ and $i = 1, \ldots, D-1$: the expected number of scrapped parts from buffer B_i in cycle t;

- *Scrap rate of the multi-stage line*, $SR(t)$ for $t = 1, 2, \ldots$: the expected number of scrapped parts from the multi-stage line in cycle t.

Remark 1. *Both Ju* et al. *(2017) and Zhang* et al. *(2013) study Bernoulli lines and are consistent with the early work of Li and Meerkov (2009) in the problem formulation of Bernoulli lines. Ju* et al. *(2017) and Zhang* et al. *(2013) have a small difference in the definition of performance measures due to the concern of transient analysis. In Ju* et al. *(2017), the authors above performance measures in one cycle in the current cycle; these are derived from the system state of the previous cycle. In Zhang* et al. *(2013), the authors derive performance measures in one cycle from the system state in the current cycle; these can be observed at the end of the next cycle. One cycle lag of performance measures is the only difference between the two studies. In the current article, we follow the definition from Ju* et al. *(2017).*

Remark 2. *Scrap rate of the multi-stage line is the summation of scrap rates of all buffers. Thus, we have* $SR(t) = \sum_{i=1}^{D-1} SR_i(t)$ *for all t.*

For the system under consideration, it is desired to maximize the production rate $PR_D(t)$ and minimize scrap rate $SR(t)$ simultaneously. The objective of the study is therefore to maximize $(PR_D(t) - \omega SR(t))$ through the actions defined in assumption (x), where ω is a positive constant to balance the trade-off between production rate $PR_D(t)$ and scrap rate $SR(t)$.

3.3. System dynamics and optimization model

Let $\mathcal{H}_i$, for $i = 1, 2, \ldots, D-1$, be a collection of all subsets of set $\{0, 1, \ldots, T_{i,max} - 1\}$ that have cardinality smaller than or equal to N_i. Specifically,

$$\mathcal{H}_i = \{h | h \subset \{0, 1, \ldots, T_{i,max} - 1\} \text{ and } |h| \leq N_i\}, \tag{1}$$

for $i = 1, 2, \ldots, D-1$, which is the state space for buffer B_i. $H_i(t) \in \mathcal{H}_i$, for $t = 0, 1, \ldots$, is defined to be the state of buffer B_i at the end of cycle t, and $H_i(t)$ represents a set of residence times of parts in buffer B_i. The occupancy of buffer B_i at the end of cycle t can be represented by $n_i = |H_i(t)|$. We follow the convention that machine state is determined at the beginning of a cycle, buffer state is determined at the end of a cycle, and the system state is determined at the end of a cycle. Thus, system state is represented by the states of all buffers. The state of a multi-stage line at the end of cycle t can be defined by $H(t) = (H_1(t), H_2(t), \ldots, H_{D-1}(t))$, which belongs to the state space of the multi-stage line, denoted by $\mathcal{H} = \otimes_{i=1}^{D-1} \mathcal{H}_i$. In addition, we define two other collections as follows:

$$\mathcal{H}_{i,min} = \{H_i \in \mathcal{H}_i | \sup H_i \geq T_{i,min}\}, \tag{2}$$

$$\mathcal{H}_{i,max} = \{H_i \in \mathcal{H}_i | \sup H_i = T_{i,max} - 1\}, \tag{3}$$

for $i = 1, 2, \ldots, D-1$. If buffer B_i is not empty, $\sup H_i$ is equal to the residence time of the first part in the buffer and we have $\tau_{i,1} = \sup H_i$. If buffer B_i is empty, then the set H_i is empty and we have $\sup H_i = -\infty$. $\mathcal{H}_{i,min}$ is a collection of states of buffer B_i that the first part in the buffer has residence time greater than or equal to $T_{i,min}$, whereas $\mathcal{H}_{i,max}$ is a collection of states of buffer B_i that the first part in the buffer has residence time equal to $(T_{i,max} - 1)$.

A discounted infinite horizon dynamic optimization problem is considered, and the objective is to maximize the discounted cumulative production rate while minimizing the scrap rate in the long term. Given the known initial state $H(0)$, the objective function is

$$\max \quad E\left\{\sum_{t=1}^{\infty} \lambda^{t-1}\left(\widetilde{PR}_D(t) - \omega \widetilde{SR}(t)\right)\right\}, \tag{4}$$

where $\widetilde{PR}_i(t)$, for $i = 1, 2, \ldots, D$, $\widetilde{SR}_i(t)$, for $i = 1, 2, \ldots, D-1$, and $\widetilde{SR}(t)$ are random variables, and $\lambda \in [0, 1)$ is the discount factor. We have $PR_i(t) = E\left[\widetilde{PR}_i(t)\right]$, $SR_i(t) = E\left[\widetilde{SR}_i(t)\right]$ and $SR(t) = E\left[\widetilde{SR}(t)\right]$. We start with machine m_D to formulate the system dynamics. The production and scrap of the last machine at time $(t+1)$, for $t = 0, 1, \ldots$, are represented as follows:

$$\chi_{\mathcal{H}_{D-1,min}}(H_{D-1}(t)) S_D(t+1) = \widetilde{PR}_D(t+1), \tag{5}$$

$$\chi_{\mathcal{H}_{D-1,max}}(H_{D-1}(t))(1 - S_D(t+1)) = \widetilde{SR}_{D-1}(t+1), \tag{6}$$

where a characteristic function $\chi_X(x)$ is used. Specifically,

$$\chi_X(x) = \begin{cases} 1 & \text{if } x \in X, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Equation (5) means that machine m_D will finish producing a part at the end of cycle $(t+1)$, if there is at least a part in buffer B_{D-1} with residence time greater than or equal to $T_{D-1,min}$ at the end of cycle t and machine m_D is up during cycle $(t+1)$. Equation (6) represents that a part will be scrapped from buffer B_{D-1} if there exists a part in buffer B_{D-1} with residence time equal to $(T_{D-1,max} - 1)$ at the end of cycle t and the machine is down during cycle $(t+1)$. Then, the state of buffer B_{D-1} is updated as follows:

$$H'_{D-1}(t) = \begin{cases} H_{D-1}(t) & \text{if } \widetilde{PR}_D(t+1) + \widetilde{SR}_{D-1}(t+1) = 0, \\ H_{D-1}(t) \setminus \sup H_{D-1}(t) & \text{otherwise,} \end{cases} \tag{8}$$

$$H''_{D-1}(t) = F\left(H'_{D-1}(t)\right), \tag{9}$$

for $t = 0, 1, \ldots$. Equation (8) suggests that the part with the largest residence time in buffer B_{D-1} is removed if a part in this buffer is either produced or scrapped. In Equation (9), we introduce an operator $F()$ on the set. For two sets X and X' such that $X' = F(X)$, $x + 1 \in X'$ is satisfied for any element $x \in X$, and $x - 1 \in X$ is satisfied for any element $x \in X'$. Equation (9) means that the residence time of each part increases by one.

In a similar way, the production rate and scrap rate of machine m_{i+1}, for $i = 1, 2, \ldots, D-2$, are expressed as follows:

$$\begin{aligned} &\chi_{\mathcal{H}_{i,min}}(H_i(t)) \chi_{\mathbb{R}_{>0}}(N_{i+1} - |H'_{i+1}(t)|) S_{i+1}(t+1) a_{i+1}(t) \\ &= \widetilde{PR}_{i+1}(t+1), \end{aligned} \tag{10}$$

$$\chi_{\mathcal{H}_{i,max}}(H_i(t))\left(1-\chi_{\mathbb{R}_{>0}}\left(N_{i+1}-|H'_{i+1}(t)|\right)S_{i+1}(t+1)a_{i+1}(t)\right)=\widetilde{SR}_i(t+1), \tag{11}$$

for $t=0,1,\ldots$. If machine m_{i+1} finishes producing a part at the end of cycle $(t+1)$, suggested by Equation (10), four conditions should be met. First, there is at least one part in buffer B_i with residence time greater than or equal to $T_{i,min}$ at the end of cycle t. Second, there is no blockage in buffer B_{i+1}. Third, machine m_{i+1} is up during cycle $(t+1)$. Finally, machine m_{i+1} is not turned down. If there is one part in buffer B_i with residence time equal to $(T_{i,max}-1)$ and at least one of the last three conditions above is not satisfied, then a part is scrapped from buffer B_i, suggested by Equation (11). Then, we update the states of those buffers, shown in Equation (12) and Equation (13) below:

$$H'_i(t)=\begin{cases} H_i(t) & \text{if } \widetilde{PR}_{i+1}(t+1)+\widetilde{SR}_i(t+1)=0,\\ H_i(t)\setminus \sup H_i(t) & \text{otherwise,}\end{cases} \tag{12}$$

$$H''_i(t)=F\left(H'_i(t)\right), \tag{13}$$

for $i=1,2,\ldots,D-2$ and $t=0,1,\ldots$. Equation (12) and Equation (13) are similar to Equation (8) and Equation (9), respectively. If a machine produces a part, the number of parts in its downstream buffer will increase by one. After considering both inflow and outflow of a buffer, we can determine the state of a buffer in the next cycle as follows:

$$H_{i+1}(t+1)=\begin{cases} H''_{i+1}(t) & \text{if } \widetilde{PR}_{i+1}(t+1)=0,\\ H''_{i+1}(t)\cup\{0\} & \text{otherwise,}\end{cases} \tag{14}$$

for $i=1,2,\ldots,D-2$ and $t=0,1,\ldots$. Equation (14) suggests that a new part with residence time equal to zero is added to buffer B_{i+1} at the end of cycle $(t+1)$ if machine m_{i+1} successfully produces a part at the end of cycle $(t+1)$. In addition, we have:

$$\chi_{\mathbb{R}_{>0}}\left(N_1-|H'_1(t)|\right)S_1(t+1)a_1(t)=\widetilde{PR}_1(t+1), \tag{15}$$

$$H''_1(t)=F\left(H'_1(t)\right), \tag{16}$$

$$H_1(t+1)=\begin{cases} H''_1(t) & \text{if } \widetilde{PR}_1(t+1)=0\\ H''_1(t)\cup\{0\} & \text{otherwise,}\end{cases} \tag{17}$$

for $t=0,1,\ldots$. Equation (15), Equation (16) and Equation (17) are for the first machine and first buffer, and they are similar to Equation (10), Equation (9) and Equation (14), respectively. Finally, by Remark 2, we have:

$$\widetilde{SR}(t+1)=\sum_{i=1}^{D-1}\widetilde{SR}_i(t+1), \tag{18}$$

for $t=0,1,\ldots$.

4. Decomposition-based control framework

4.1. Complexity of multi-stage line

The production control problem introduced in Section 3 cannot be analyzed directly, due to the large state space. The total number of system states of a multi-stage line, denoted by M, is provided as follows:

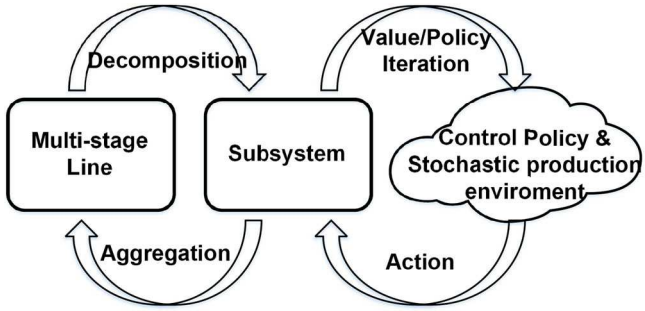


Figure 2. Concept of decomposition-based control.

$$M=\prod_{i=1}^{D-1}\sum_{j=0}^{N_i}\binom{T_{i,max}}{j}. \tag{19}$$

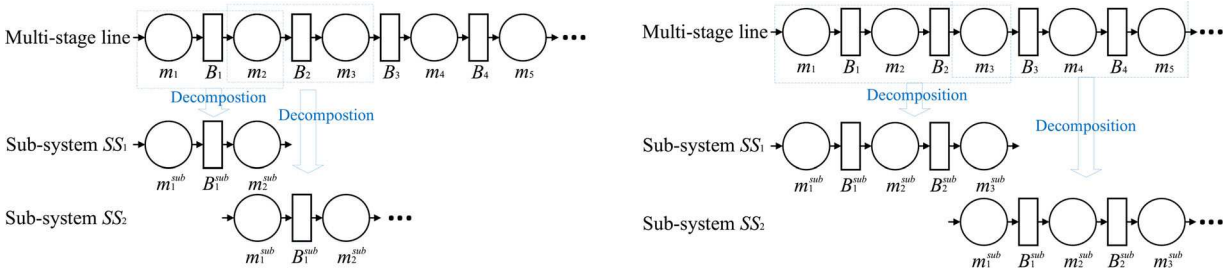
Consider a single buffer first. If we fix the buffer occupancy to be j, the number of combinations for a buffer is equal to the number of ways to choose j different residence times from $T_{i,max}$ options, which is represented by

$$\binom{T_{i,max}}{j}.$$

Then, the total number of system states can be calculated by considering all buffers and all possible buffer occupancies. For example, for a multi-stage line that has seven machines and six buffers with buffer capacity $N_i=6$ and maximum allowable residence time $T_{i,max}=8$, for $i=1,2,\ldots,6$, the number of system states is as large as 2.3×10^{14} according to Equation (19). To deal with this level of complexity, one common approach is to use reinforcement learning to perform production control by approximately mapping system states and actions to rewards. However, these methods result in a long training time and suffer from interpretability. In addition, the approximation architecture can quickly deteriorate as the problem scale continues to increase. To tackle these issues, we propose a novel approach, decomposition-based control. We hypothesize that, by leveraging the system decomposition, we can effectively optimize production performance in real-time.

4.2. Overview of the decomposition-based control approach

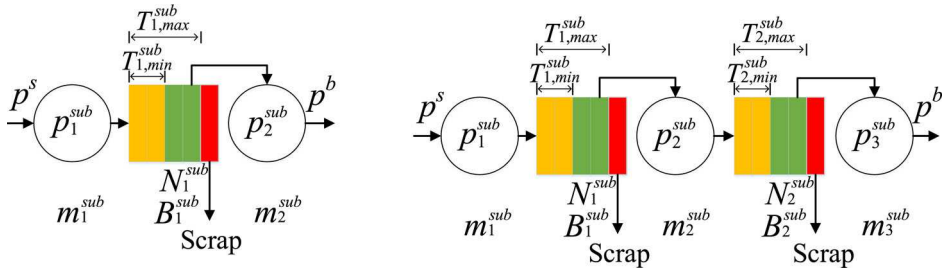
Instead of analyzing and controlling a multi-stage line as a whole, we propose the decomposition-based control approach. The concept of decomposition-based control is shown in Figure 2. A multi-stage line is decomposed into subsystems, and a structural relationship between subsystems is defined. Under a properly defined structural relationship, each subsystem is assumed to behave like its corresponding part in the multi-stage line. Each subsystem is modeled independently as an MDP model. Since the state space of a subsystem is small enough, the control policy for each subsystem can be derived through value iteration or policy iteration. Each subsystem takes action by observing its local environment. The control policy of a multi-stage line is a combination of all control policies derived from all subsystems. However, as a control policy is implemented,



(a) Two-machine-one-buffer subsystems

(b) Three-machine-two-buffer subsystems

Figure 3. System decomposition with two- or three-machine subsystems.



(a) A two-machine-one-buffer subsystem

(b) A three-macine-two-buffer subsystem

Figure 4. Models for subsystems.

the original structural relationship between subsystems changes. Due to this change, the behavior of a subsystem does not truly represents its corresponding part in the multi-stage line. It requires subsystems to update their relationship according to the current control policy, which is part of the aggregation procedure. A new iteration starts, since the current control policy may not be optimal as the relationship between subsystems is updated. The MDP model for each subsystem with updated relationship is developed, and new control policy is derived. After several iterations, this process converges, and each subsystem has a similar behavior as its corresponding part in the multi-stage line. The control policy for each subsystem can achieve a global improvement.

In the following subsections, the system decomposition and modeling of subsystems will be introduced in details, and a novel aggregation-based procedure will be provided to generate the control policy.

4.3. System decomposition

A subsystem, isolated from a multi-stage line, serves as a building block to support the decomposition-based control, and it can be a two-machine-one-buffer subsystem or a three-machine-two-buffer subsystem, shown in Figure 3. Figure 3(a) shows how a multi-stage line with D machines is decomposed into $(D-1)$ two-machine-one-buffer subsystems. Each subsystem consists of two machines, m_1^{sub} and m_2^{sub}, and a buffer B_1^{sub}. The ith subsystem of a multi-stage line is denoted by SS_i, for $i = 1, 2 \cdots, D-1$. The control model for a two-machine-one-buffer subsystem is to determine when to turn machine m_1^{sub} down based on the state of the subsystem. Figure 3(b) shows how a multi-stage line with D machines is decomposed into $(D-1)/2$ three-machine-two-buffer subsystems. Each three-machine-two-buffer subsystem consists of three machines, denoted by m_1^{sub}, m_2^{sub} and m_3^{sub}, and two buffers, denoted by B_1^{sub} and B_2^{sub}. Similar to a two-machine-one-buffer subsystem, the control model for a three-machine-two-buffer subsystem is to control machine m_1^{sub} and m_2^{sub} according to the state of the subsystem.

We decompose a transfer line into subsystems, since the system as a whole is infeasible to analyze directly, due to its large state space. Decomposition, as an approximation-based method, can compromise modeling accuracy, when too many subsystems are involved. Also to consider the complexity of the subsystem itself, we find the balance to be the use of a three-machine-two-buffer subsystem as a general building block in the decomposition method. One two-machine-one-buffer subsystem will be utilized to handle the systems with an even number of machines.

4.4. Descriptive model of subsystem

The relevant parameters to model a subsystem are presented in Figure 4. The ith machine in a subsystem is denoted by m_i^{sub}, and it is a Bernoulli machine with parameter p_i^{sub}. The ith buffer in a subsystem is denoted by B_i^{sub}. Buffer B_i^{sub} is described by buffer capacity N_i^{sub}, maximum allowable residence time $T_{i,max}^{sub}$ and minimum required residence time $T_{i,min}^{sub}$. Neighboring subsystems are mutually influenced. Such influence is modeled by starvation probability, p^s, and blockage probability, p^b. The probability that machine m_1^{sub}

is not able to produce, due to the starvation from its upstream buffer is denoted by p^s. If buffer B_1^{sub} has available space, the probability that the first machine can produce is $p_1^{sub}(1-p^s)$. Machine m_2^{sub} in a two-machine-one-buffer subsystem and machine m_3^{sub} in a three-machine-two-buffer subsystem are shared by its downstream subsystem, which is illustrated in Figure 3. The probability p^b represents the probability that machine m_2^{sub} in a two-machine-one-buffer subsystem or machine m_3^{sub} in a three-machine-two-buffer subsystem is not allowed to work, either due to downstream blockage or the control policy of the downstream subsystem. Thus, if there is at least one part in buffer B_1^{sub} of a two-machine-one-buffer subsystem or buffer B_2^{sub} of a three-machine-two-buffer subsystem with residence time larger than or equal to the minimum required residence time, the probability that the part can be produced and leave the subsystem is $p_2^{sub}(1-p^b)$ and $p_3^{sub}(1-p^b)$ for a two-machine-one-buffer subsystem and three-machine-two-buffer subsystem, respectively. Assumption (ix) suggests that machine m_1 is never starved, so p^s is always equal to zero for the first subsystem. Similarly, the last machine of the last subsystem is never blocked, and thus p^b in the last subsystem is always equal to zero.

4.5. Markov decision model for the subsystem

When a two-machine-one-buffer subsystem is isolated, the subsystem can be viewed as a two-machine transfer line with two Bernoulli machines with parameters $p_1^{sub}(1-p^s)$ and $p_2^{sub}(1-p^b)$, respectively. Similarly, a three-machine-two-buffer subsystem can be viewed as a three-machine transfer line with three Bernoulli machines with parameters $p_1^{sub}(1-p^s)$, p_2 and $p_3^{sub}(1-p^b)$, respectively. The modeling of two-machine-one-buffer subsystem shares similarities with the modeling of a three-machine-two-buffer subsystem. In this subsection, we only show how to model a three-machine-two-buffer subsystem without repeating it for two-machine-one-buffer subsystem.

- Decision epochs: $t = 0, 1, \ldots$.
- System state: $h^{sub}(t) = (n_1^{sub}, \tau_1^{sub}, n_2^{sub}, \tau_2^{sub}) \in \mathcal{H}^{sub}$. n_1^{sub} and n_2^{sub} are the buffer occupancy of buffer B_1^{sub} and buffer B_2^{sub}, respectively. τ_1^{sub} and τ_2^{sub} are the residence time of the first part in buffer B_1^{sub} and buffer B_2^{sub}, respectively, if the buffer is not empty. Let $\tau_i^{sub} = 0$, for $i = 1, 2$, if $n_i^{sub} = 0$. The state space of a subsystem is denoted by $\mathcal{H}^{sub}$.
- Action: $\boldsymbol{a}^{sub}(t) = \begin{bmatrix} a_1^{sub}(t) & a_2^{sub}(t) \end{bmatrix}^T \in \mathcal{A}^{sub}$, where $a_i^{sub}(t) \in \{1, 0\}$, for $i = 1, 2$, at any time t. The action $a_i^{sub}(t) = 0$ makes machine m_i^{sub} not work in cycle $(t+1)$, and the action $a_i^{sub}(t) = 1$ keeps machine m_i^{sub} unchanged. The action space of a subsystem is denoted by $\mathcal{A}^{sub}$.
- Reward: the reward at time $(t-1)$ is denoted by $r(h^{sub}(t-1), \boldsymbol{a}^{sub}(t-1))$. Specifically,

$$r\left(h^{sub}(t-1), \boldsymbol{a}^{sub}(t-1)\right) = \widetilde{PR}^{sub}(t) - \omega \widetilde{SR}^{sub}(t), \quad (20)$$

where $\widetilde{PR}^{sub}(t)$ and $\widetilde{SR}^{sub}(t)$ are random variables representing the production of the last machine m_3^{sub} and the scrap from both buffers, respectively.

- Expected total discounted reward of policy π^{sub}:

$$v^{\pi^{sub}} = E^{\pi^{sub}}\left\{\sum_{t=0}^{\infty} \lambda^t r\left(h^{sub}(t), \boldsymbol{a}^{sub}(t)\right)\right\}, \quad (21)$$

where $\lambda \in [0, 1)$ is the discount.

The optimal control policy of a subsystem can be expressed as

$$\pi^* \in \arg\max_{\pi^{sub}} E^{\pi^{sub}}\left\{\sum_{t=0}^{\infty} \lambda^t r\left(h^{sub}(t), \boldsymbol{a}^{sub}(t)\right)\right\}. \quad (22)$$

Remark 3. *We define the state of a buffer only by the buffer occupancy and the residence time of the first part in the buffer. Following the approximate method detailed in Ju* et al. *(2017), the optimization problem can be treated as an MDP with an exact stochastic model, and standard methods, such as the value iteration and the policy iteration, can be used to solve the problem.*

Let $\pi^{sub}: \mathcal{H}^{sub} \to \mathcal{A}^{sub}$ be a mapping from state to action under control policy π^{sub}. As control policy π^{sub} is implemented, the subsystem reaches steady state. Let $\mu: \mathcal{H}^{sub} \to [0, 1]$ be a mapping from state to its steady-state probability under control policy π^{sub}. $\widehat{PR}^{sub}$, $\widehat{SR}^{sub}$, $\widehat{ST}^{sub}$ and $\widehat{BL}^{sub}$ denote the estimated long-term performance measures of the subsystem under control policy π^{sub}, and they are defined and derived as follows.

- *Estimated production rate* $\widehat{PR}^{sub}$: the expected number of parts produced by the last machine of the subsystem in a cycle, and specifically,

$$\widehat{PR}^{sub} = \sum_{h^{sub} \in \mathcal{H}_{PR}^{sub}} \mu(h^{sub}) p_3^{sub}(1-p^b), \quad (23)$$

where

$$\mathcal{H}_{PR}^{sub} = \left\{h^{sub} \in \mathcal{H}^{sub} | \tau_2^{sub} \geq T_{2,min}^{sub}\right\}. \quad (24)$$

The subset of state space, $\mathcal{H}_{PR}^{sub}$, represents all states where the residence time of the first part in buffer B_2^{sub} is equal to or larger than $T_{2,min}^{sub}$. It is suggested by Equation (23) that one part can be produced for a subsystem in a state in $\mathcal{H}_{PR}^{sub}$ if machine m_3^{sub} is up and there is no blockage to the machine.

- *Estimated scrap rate* $\widehat{SR}^{sub}$: the expected number of scrapped parts from the subsystem in a cycle, and specifically,

$$\begin{aligned}\widehat{SR}^{sub} = & \sum_{h^{sub} \in \mathcal{H}_{SR,1}^{sub}} \mu(h^{sub})\left(1 - \begin{bmatrix} 0 & p_2^{sub} \end{bmatrix} \pi^{sub}(h^{sub})\right) \\ & + \sum_{h^{sub} \in \mathcal{H}_{SR,2}^{sub}} \mu(h^{sub}) \begin{bmatrix} 0 & p_2^{sub} \end{bmatrix} \pi^{sub}(h^{sub})\left(1 - p_3^{sub}(1-p^b)\right) \\ & + \sum_{h^{sub} \in \mathcal{H}_{SR,3}^{sub}} \mu(h^{sub})\left(1 - p_3^{sub}(1-p^b)\right),\end{aligned} \quad (25)$$

where

$$\mathcal{H}^{sub}_{SR,1} = \left\{ h^{sub} \in \mathcal{H}^{sub} | \tau_1^{sub} = T^{sub}_{1,max} - 1 \right\}, \tag{26}$$

$$\mathcal{H}^{sub}_{SR,2} = \left\{ h^{sub} \in \mathcal{H}^{sub} | \tau_1^{sub} = T^{sub}_{1,max} - 1, n_2^{sub} = N_2^{sub}, \tau_2^{sub} < T^{sub}_{2,max} - 1 \right\}. \tag{27}$$

$$\mathcal{H}^{sub}_{SR,3} = \left\{ h^{sub} \in \mathcal{H}^{sub} | \tau_2^{sub} = T^{sub}_{2,max} - 1 \right\}. \tag{28}$$

Equation (25) is the summation of three terms. The first term represents the case that a part is scrapped from buffer B_1^{sub} due to failure of machine m_2^{sub} or an action that turns machine m_2^{sub} down. In the second term, machine m_2^{sub} is capable of working, but a part is scrapped from buffer B_1^{sub} due to blockage of buffer B_2^{sub}. The third term represents a scrap from buffer B_2^{sub} caused by machine m_3^{sub}.

- *Estimated starvation probability* $\widehat{ST}^{sub}$: the probability that the last machine of the subsystem is not able to produce due to starvation, and specifically,

$$\widehat{ST}^{sub} = \sum_{h^{sub} \in \mathcal{H}^{sub}_{ST}} \mu(h^{sub}), \tag{29}$$

where

$$\mathcal{H}^{sub}_{ST} = \left\{ h^{sub} \in \mathcal{H}^{sub} | \tau_2^{sub} < T^{sub}_{2,min} \right\}. \tag{30}$$

The estimated starvation probability $\widehat{ST}^{sub}$ is the probability that buffer B_2^{sub} has no part with residence time larger than or equal to $T^{sub}_{2,min}$.

- *Estimated blockage probability* $\widehat{BL}^{sub}$: the probability that the first machine of the subsystem is not able to produce due to blockage or control policy, and specifically,

$$\begin{aligned} \widehat{BL}^{sub} = & \sum_{h^{sub} \in \mathcal{H}^{sub}} \mu(h^{sub}) \left(1 - [1 \quad 0] \pi^{sub}(h^{sub})\right) \\ & + \sum_{h^{sub} \in \mathcal{H}^{sub}_{BL,1}} \mu(h^{sub}) [1 \quad 0] \pi^{sub}(h^{sub}) \\ & \left(1 - \begin{bmatrix} 0 & p_2^{sub} \end{bmatrix} \pi^{sub}(h^{sub})\right) \\ & + \sum_{h^{sub} \in \mathcal{H}^{sub}_{BL,2}} \mu(h^{sub}) [1 \quad 0] \pi^{sub}(h^{sub}) \\ & \begin{bmatrix} 0 & p_2^{sub} \end{bmatrix} \pi^{sub}(h^{sub}) \left(\left(1 - p_3^{sub}\right) + p_3^{sub} p^b \right), \end{aligned} \tag{31}$$

where

$$\mathcal{H}^{sub}_{BL,1} = \left\{ h^{sub} \in \mathcal{H}^{sub} | n_1^{sub} = N_1^{sub}, \tau_1^{sub} < T^{sub}_{1,max} - 1 \right\}, \tag{32}$$

$$\begin{aligned} \mathcal{H}^{sub}_{BL,2} = \{ & h^{sub} \in \mathcal{H}^{sub} | n_1^{sub} = N_1^{sub}, \tau_1^{sub} < T^{sub}_{1,max} - 1, n_2^{sub} \\ & = N_2^{sub}, \tau_2^{sub} < T^{sub}_{2,max} - 1 \}. \end{aligned} \tag{33}$$

The first term of Equation (31) is the probability that machine m_1^{sub} is blocked by the control policy that directly turns machine m_1^{sub} down. The second term represents the case when buffer B_1^{sub} is full and machine m_2^{sub} cannot produce a part from buffer B_1^{sub} due to the control policy or failure on machine m_2^{sub}. The third term gives the situation where both buffer B_1^{sub} and buffer B_2^{sub} are full and machine m_3^{sub} cannot produce a part due to the control policy or failure.

In a similar way, the MDP model of a two-machine-one-buffer subsystem can be built, and the performance measures of a two-machine-one-buffer subsystem can be derived.

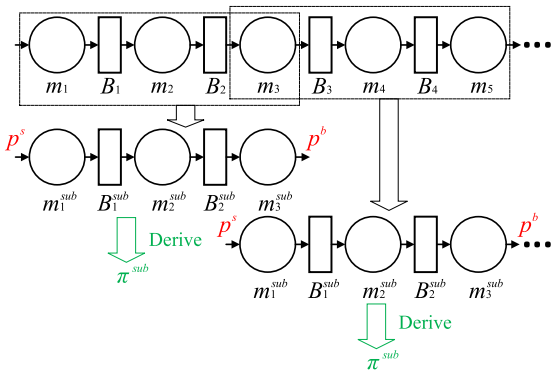
4.6. Aggregation procedure

The structural relationship between neighboring subsystems is defined by the starvation probability p^s and blockage probability p^b. If p^s and p^b are accurate, the behavior of a subsystem will be similar to its corresponding part in the multi-stage line. The control policy of each subsystem is derived from its MDP model as p^s and p^b are assumed to be known. However, as the control policy of each subsystem is implemented, it changes the relationship between neighboring subsystems. Thus, it requires the relationship to be updated. The update of the relationship further requires each subsystem to derive an updated control policy. Thus, an iterative method, the aggregation procedure, is proposed to update the relationship between neighboring subsystems and the control policy of each subsystem.

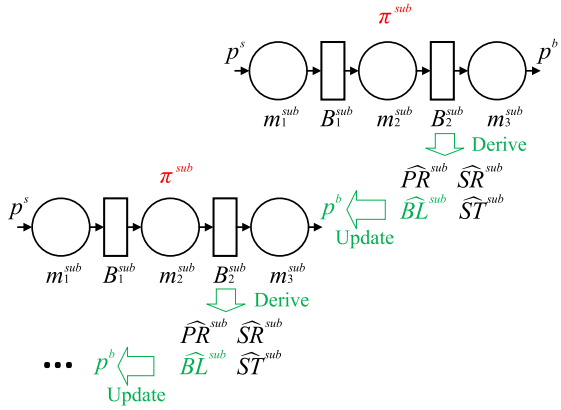
The aggregation procedure, shown in Figure 5, includes the backward aggregation and the forward aggregation. Figure 5(a) shows that a multi-stage line is decomposed into several subsystems, and a control policy π^{sub} is derived for each subsystem as the starvation probability p^s and the blockage probability p^b of each subsystem are assumed to be known and fixed. Figure 5(b) and Figure 5(c) illustrate the backward aggregation and the forward aggregation, respectively. In this process, the control policy π^{sub} is fixed, and p^b and p^s are updated through the backward aggregation and forward aggregation, respectively. In addition, performance measures, including $\widehat{PR}^{sub}$, $\widehat{SR}^{sub}$, are derived.

The backward aggregation, shown in Figure 5(b), starts with the last subsystem and moves backward. The blockage probability $\widehat{BL}^{sub}$, derived by Equation (31) from the a subsystem, is used to update p^b of its upstream neighboring subsystem, and this process continues until p^b of the first subsystem is updated. The forward aggregation, shown in Figure 5(c), is similar to the backward aggregation but starts with the first subsystem. The starvation probability $\widehat{ST}^{sub}$, derived by Equation (29) from a subsystem, is used to update p^s of its downstream neighboring subsystem. This forward aggregation continues until p^s of the last subsystem is updated.

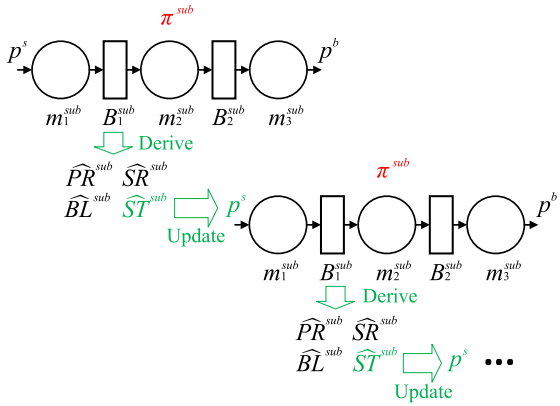
Figure 6 provides the pseudocode of the decomposition-based control approach. Line 1 is to decompose the multi-stage line into subsystems. Line 2 initializes the control policy for each subsystem, and the initial control policy never turns machines down. The decomposition-based control consists of several iterations to finally derive the control



(a) Fix p^s and p^b, and derive control policy π^{sub}



(b) The backward aggregation. Fix control policy π^{sub} and update p^b



(c) The forward aggregation. Fix control policy π^{sub} and update p^s

Figure 5. The aggregation procedure.

policy for each subsystem, and the iterations are presented from line 3 to line 18. It is a loop from line 3 to line 18. Inside the loop, the steps from line 6 to line 9 represent the backward aggregation, and the the steps from line 10 to line 13 represent the forward aggregation. As p^s and p^b of each subsystem are updated, the new control policy for each subsystem is derived, shown from line 15 to line 17. A stop criterion is set for the loop, and it can be a certain number of iterations or the indication of convergence of p^s and p^b.

4.7. Convergence

A parameter setting is selected as follows to numerically study the convergence of the aggregation procedure:

$$\begin{aligned} &D = 7, \\ &p_1 = 0.9, p_2 = 0.87, p_3 = 0.85, p_4 = 0.83, \\ &p_5 = 0.8, p_6 = 0.77, p_7 = 0.75, \\ &N_i = 6, T_{i,max} = 8, T_{i,min} = 2, \quad \text{for } i = 1, \ldots, D-1, \\ &\omega = 1.3. \end{aligned} \tag{34}$$

The discount, λ, is set to be 0.95. A set of control policies for subsystems are obtained in each iteration, and we compare the steady-state performance measures under those control policies through simulation. The simulation repeats 1000 times, and the steady-state performance measures are shown in Figure 7. The horizontal axis represents the iterations, and the vertical axis represents the performance measures. Iteration 0 shows the performance measures where the initial control policy is implemented. The result suggests that the decomposition-based control can soon improve the performance in a small number of iterations. The performance measures oscillate with in a small zone, primarily due to the random error from the simulation. The oscillation of production rate looks more obvious, because the control policy does not significantly change the production rate.

To numerically study the convergence in a more general sense, we introduce vectors $\boldsymbol{p^s}_i$ and $\boldsymbol{p^b}_i$, for $i = 0, 1, \ldots$. Let $\boldsymbol{p^s}_i$ and $\boldsymbol{p^b}_i$, for $i = 0, 1, \ldots$, be a vector of the starvation probabilities and a vector of blockage probabilities from all subsystems under the control policy obtained from the ith iteration, respectively. Specifically,

$$\boldsymbol{p^s}_i = \begin{bmatrix} p_1^s & p_2^s & \cdots \end{bmatrix}^T, \tag{35}$$

$$\boldsymbol{p^b}_i = \begin{bmatrix} p_1^b & p_2^b & \cdots \end{bmatrix}^T, \tag{36}$$

where p_j^s and p_j^b, for $j = 1, 2, \ldots$, are the starvation probability p^s and blockage probably p^b of subsystem SS_j, respectively. The distance of $\boldsymbol{p^s}_i$ and $\boldsymbol{p^s}_{i-1}$ and the distance of $\boldsymbol{p^b}_i$

```
Decompose multi-stage line into K subsystems
Initialize control policy for each subsystem
while Stop criteria is not satisfied do
    Initialize p^s and p^b to 0
    while Stop criteria is not satisfied do
        for k = 1, K − 1 do
            Derive performance of subsystem SS_{K−k+1}
            Update p^b of subsystem SS_{K−k}
        end for
        for k = 1, K − 1 do
            Derive performance of subsystem SS_k
            Update p^s of subsystem SS_{k+1}
        end for
    end while
    for k = 1, K − 1 do
        Obtain control policy for subsystem SS_k
    end for
end while
```

Figure 6. The iterative procedure for decomposition-based control.

and $\boldsymbol{p}^{\boldsymbol{b}}{}_{i-1}$ are denoted by d_i^s and d_i^b for $i = 1, 2, \ldots$, respectively, and defined as follows:

$$d_i^s = (\boldsymbol{p}^{\boldsymbol{s}}{}_i - \boldsymbol{p}^{\boldsymbol{s}}{}_{i-1})^T(\boldsymbol{p}^{\boldsymbol{s}}{}_i - \boldsymbol{p}^{\boldsymbol{s}}{}_{i-1}), \tag{37}$$

$$d_i^b = (\boldsymbol{p}^{\boldsymbol{b}}{}_i - \boldsymbol{p}^{\boldsymbol{b}}{}_{i-1})^T(\boldsymbol{p}^{\boldsymbol{b}}{}_i - \boldsymbol{p}^{\boldsymbol{b}}{}_{i-1}). \tag{38}$$

The convergence can be observed, if d_i^s and d_i^b are getting close to zero as i increases.

To numerically show the convergence of the aggregation procedure of the decomposition-based control, 2000 parameter settings are randomly generated from the range of parameter settings as follows:

$$\begin{aligned}
&p_1 \in [0.85, 0.99],\\
&p_i \in [0.65, 0.99] \text{ for } i = 2, \ldots, D,\\
&N_i \in \{5, 6, 7\} \text{ for } i = 1, \ldots, D-1,\\
&T_{i,max} \in \{N_i + 1, N_i + 2, N_i + 3\} \text{ for } i = 1, \ldots, D-1,\\
&T_{i,min} \in \{1, 2\} \text{ for } i = 1, \ldots, D-1,\\
&\omega \in [0.7, 1.7].
\end{aligned} \tag{39}$$

Parameters are selected with equal probability from the range. We let the number of machines be nine. The number of iterations is set to be eight. Both d_8^s and d_8^b at the end of the iteration are obtained for each parameter setting. The experiment result shows that 99.95% of all cases have d_8^s smaller than 10^{-3} and 100.00% of the cases result in d_8^b smaller than 10^{-3}. This indicates that the performance measures converge within a small interval after a certain number of iterations.

5. Numerical experiments and performance comparison

5.1. RL control for comparison

The decomposition-based control is compared with a feature-based reinforcement learning control (RL control). In the RL control, a feature-based architecture is used to handle the large state space.

Let $r(H(t-1), \boldsymbol{a}(t-1))$ be the reward function of the multi-stage line at time $(t-1)$. Specifically,

$$r\big(H(t-1), \boldsymbol{a}(t-1)\big) = \widetilde{PR_D}(t) - \omega\widetilde{SR}(t). \tag{40}$$

Given the initial system state $H(0)$, the optimal expected total discounted reward is expressed as follows:

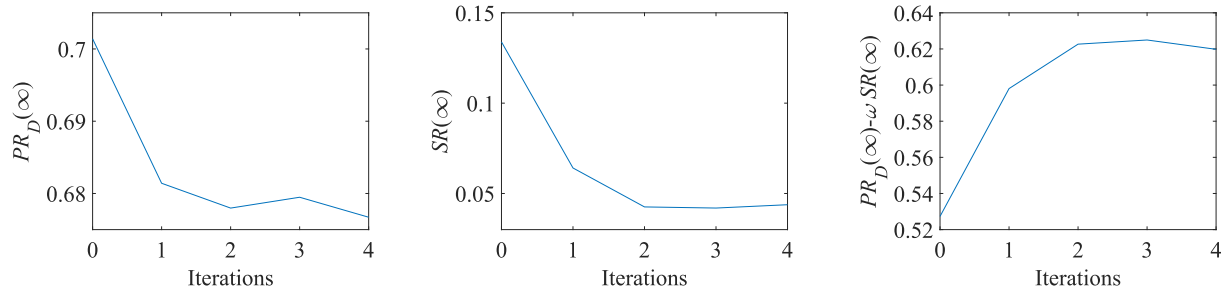
$$v^*(H(0)) = \max_{\pi} E^{\pi}\left\{\sum_{i=0}^{\infty} \lambda^i r(H(i), \boldsymbol{a}(i))\right\}, \tag{41}$$

which, however, is impossible to obtain, due to the large state space of the problem. An approximate lookahead function $\hat{v}(\phi(H(t)), \boldsymbol{\beta})$ with parameters $\boldsymbol{\beta}$ is introduced to replace $v^*(H(t))$. Function $\phi(H(t))$ maps system state $H(t)$ to the feature, and $\hat{v}(\phi(H(t)), \boldsymbol{\beta})$ can be obtained through training. The buffer occupancy of each buffer and the residence time of the first part in each buffer are important measures to capture system dynamics, and thus we take them as candidates of features. To further explore features, a preliminary analysis of features is performed with parameters given as follows.

$$\begin{aligned}
&D = 4,\\
&p_1 = 0.9, p_2 = 0.83, p_3 = 0.75, p_4 = 0.7,\\
&N_i = 6, \text{ for } i = 1, 2, 3\\
&T_{i,max} = 8 \text{ for } i = 1, 2, 3,\\
&T_{i,min} = 0 \text{ for } i = 1, 2, 3,\\
&\omega = 0.9, \lambda = 0.95.
\end{aligned} \tag{42}$$

Let the initial buffer occupancy of each buffer be two and residence time of the first part in each buffer be six. The effect of the initial buffer occupancy is studied. We change the initial buffer occupancy from one to six for each buffer each time with all other parameters fixed. For each initial buffer occupancy, 4000 initial system states are randomly generated, and a simulation is run for 50 cycles starting with each initial system state. The average total discounted rewards, $\sum_{t=1}^{50} \lambda^{t-1}(PR(t) - \omega SR(t))$, starting with different initial buffer occupancy are compared. The result is shown in Figure 8. This suggests that, to have a large average total discounted reward, the buffer occupancy should not be either too small or too large. A small buffer occupancy results in a high probability of starvation for the downstream machines, and it reduces the production rate. In contrast, a large buffer occupancy requires a long time to have all the parts in the buffer processed, and the risk of scrap increases.

Following the same way with parameters given in Equation (42), we study the effect of initial residence time of the first part in the buffer. The initial buffer occupancy is set to be four for each buffer, and the initial residence time of the head part in each buffer is set to be three. We change the residence time from three to seven and plot the average total discounted reward in Figure. 9. A trend can be seen that the average total discounted reward decreases as the initial residence time of the head part in the buffer increases. A large residence time results in a high risk of scrap, and thus a small residence time is always preferred.



(a) Production rate (b) Scrap rate (c) Reward

Figure 7. Steady-state performance measures with control policies obtained in each iteration.

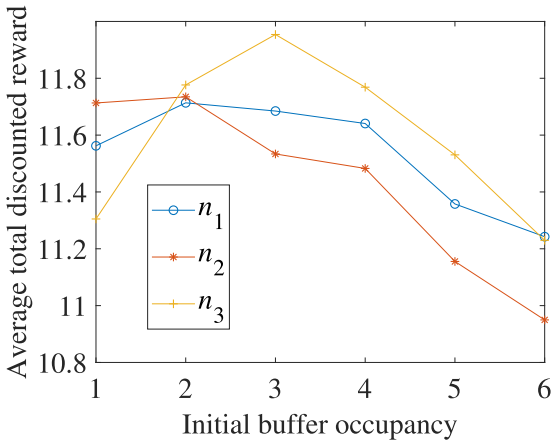


Figure 8. The average total discounted reward with different initial buffer occupancy.

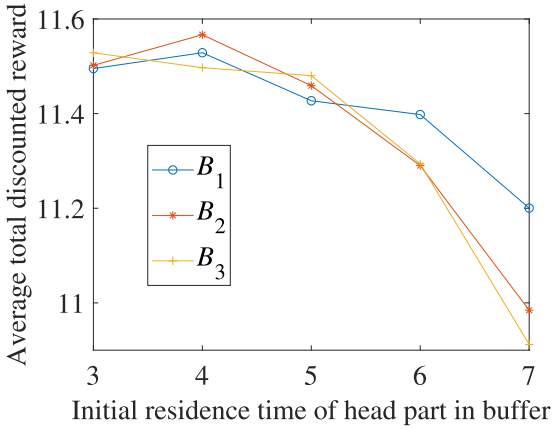


Figure 9. The average total discounted reward with different initial residence time of head part in the buffer.

According to the simulation study, three features are adopted for each buffer, and they are the buffer occupancy n_i, the square of the buffer occupancy n_i^2, and residence time of the first part in the buffer $\tau_{i,1}$. Thus, the features for the multi-stage line are provided by

$$\phi(H(t)) = [\phi_1 \quad \phi_2 \quad \cdots \quad \phi_{3D-2}]^T, \tag{43}$$

where ϕ_1 is a constant term, and ϕ_{3i-1}, ϕ_{3i} and ϕ_{3i+1} are features of buffer B_i, for $i = 1, 2, \ldots, D-1$. Specifically,

$$\begin{aligned} \phi_{3i-1} &= n_i, \\ \phi_{3i} &= n_i^2, \\ \phi_{3i+1} &= \begin{cases} \tau_{i,1}, & \text{if } n_i \neq 0 \\ 0, & \text{if } n_i = 0 \end{cases}, i = 1, 2, \ldots, D-1. \end{aligned} \tag{44}$$

Then, the lookahead function, following a linear feature-based architecture, is expressed as follows:

$$\hat{v}(\phi(H(t)), \boldsymbol{\beta}) = \boldsymbol{\beta}^T \phi(H(t)). \tag{45}$$

Parameter $\boldsymbol{\beta}$ in Equation (45) can be estimated in training through simulation. The optimal action can be expressed as

$$\boldsymbol{a}^*(t-1) \in \arg\max_{\boldsymbol{a}(t-1)\in\mathcal{A}} E\{r(H(t-1), \boldsymbol{a}(t-1)) + \lambda \hat{v}(\phi(H(t)), \boldsymbol{\beta})\}. \tag{46}$$

5.2. Simulation experiment with a single case

To show how the decomposition-based control improves the multi-stage line, we use the parameter setting in Equation (34). The simulation runs 200 cycles with all buffers empty initially and repeats 1000 times. The multi-stage line is decomposed into three-machine-two-buffer subsystems.

The result of the simulation experiment is shown in Figure 10. In each one of the three plots in Figure 10, the horizontal axis represents the time from cycle 0 to cycle 200, and the vertical axis represents the performance measures. There are three plots representing three performance measures, and they are production rate $PR_D(t)$, scrap rate $SR(t)$ and reward $(PR_D(t) - \omega SR(t))$. The average performance measures and 95% confidence intervals without control are plotted by blue lines and blue shaded areas, respectively. Similarly, the green color and red color are used for the RL control and the decomposition-based control, respectively.

Production rates with two control methods and without control are plotted in Figure 10(a), and it shows no significant difference in production rates among the three methods. The two control methods slightly reduce the production rate. Among the two control methods, the decomposition-based control maintains a higher production rate. The two control methods show a significant improvement in the scrap rate, shown in Figure 10(b), and in this case, RL control reduces the scrap rate to a greater extent. This result suggests that both control methods can significantly reduce the scrap rate without sacrificing too much in the production rate. Figure 10(c) shows the rewards of the three methods. The rewards under RL control and decomposition-based control are higher than

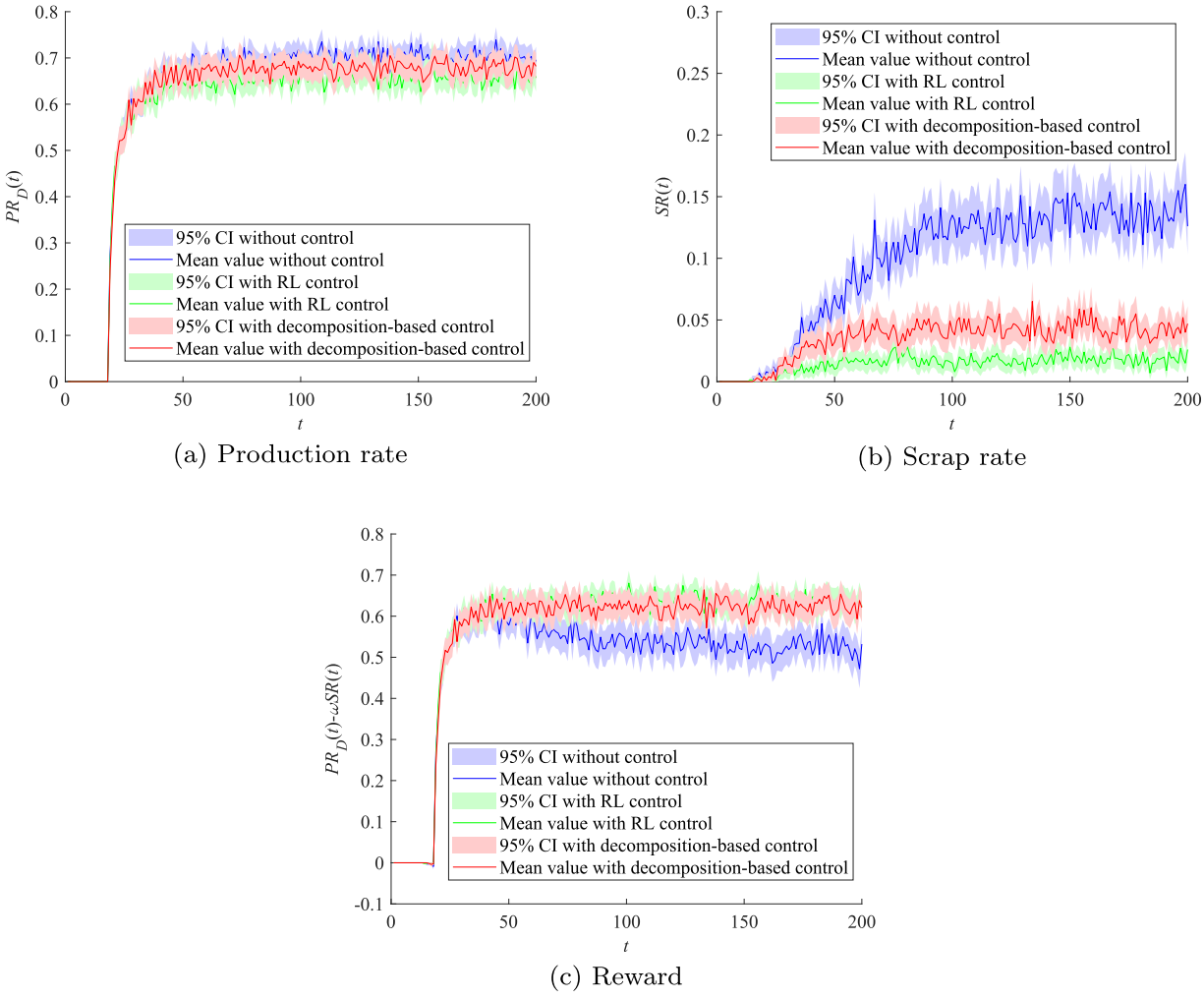


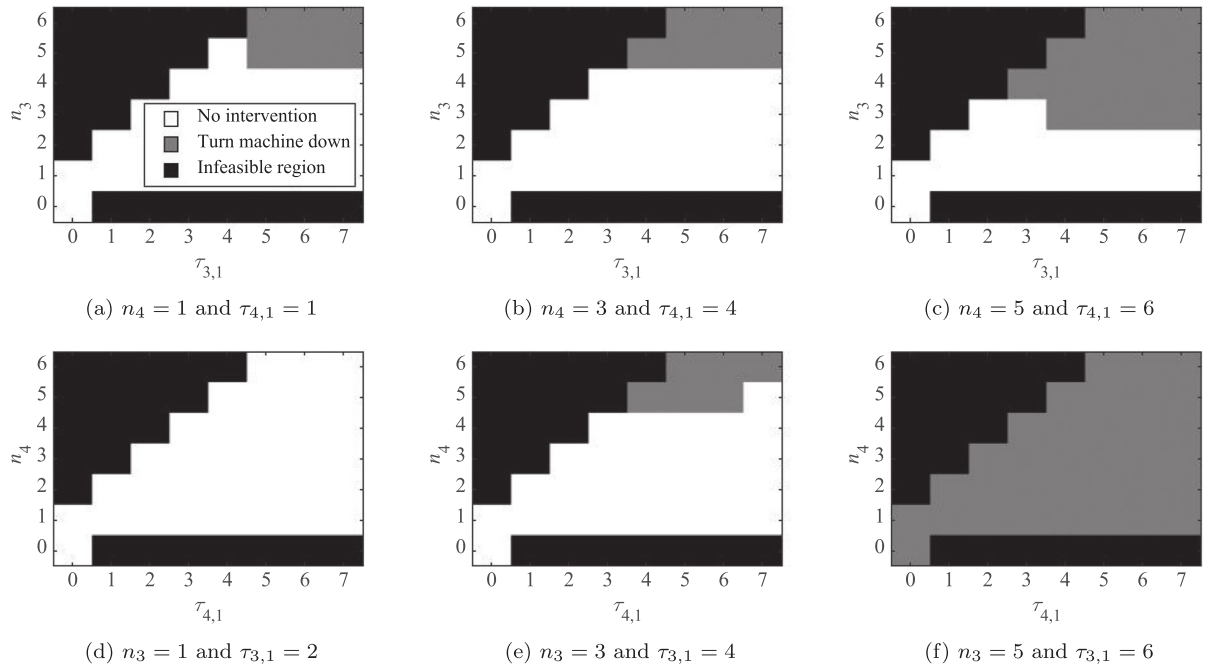
Figure 10. Comparison of performance measures.

the reward without control. The rewards of RL control and decomposition-based control are almost overlapped, and RL control results in a slightly higher reward in this case. In terms of computing time, the decomposition-based control is much more computationally efficient than the RL control. The experiment runs on a server with Intel(R) Core(TM) i7-5930K CPU, sufficiently large RAM and Linux operating system. In this single experiment, the decomposition-based control takes 19 seconds to generate the control policy, whereas the RL control needs as much as 10 222 seconds for training.

After the last iteration of the aggregation procedure, each subsystem has its control policy. We partially present the control policies for machine m_3 and machine m_4 in Figure 11 and Figure 12, respectively. Machine m_3 and machine m_4 are assigned to the second subsystem, which consists of machine m_3, machine m_4, machine m_5, buffer B_3 and buffer B_4. Both machine m_3 and machine m_4 take actions by observing the states of buffer B_3 and buffer B_4.

Figure 11 presents the control policy for machine m_3. We first fix the state of buffer B_4. The relationship between the action that machine m_3 takes and the state of buffer B_3 is illustrated in Figure 11(a), Figure 11(b) and Figure 11(c). The horizontal axis represents the residence time of the first part in buffer B_3, and the vertical axis represents the buffer occupancy of buffer B_3. Given that the state of buffer B_4 is fixed, a state for the subsystem is represented by a block in the figure. The black blocks are the infeasible regions that the subsystem never visits. In the feasible regions, a block is colored to be white or gray, indicating two actions. The white color means that machine m_3 will be unchanged, whereas the gray color indicates that machine m_3 will be turned down manually. Figure 11(a) shows the case when buffer B_4 has a low buffer occupancy and a small residence time of the first part. It can be observed that machine m_3 is turned down only when there is a high buffer occupancy in buffer B_3. Figure 11(b) shows a control policy where buffer B_4 has a median buffer occupancy and a median residence time of the first part, and the control policy is similar to the control policy shown in Figure 11(a). When buffer B_4 reaches a high occupancy and has a large residence time of the first part, machine m_3 is more likely to be turned down to maintain a lower buffer occupancy for buffer B_3, shown in Figure 11(c).

Then, we fix the state of buffer B_3 and present the control policy with respect to the state of buffer B_4. The result is shown in Figure 11(d), Figure 11(e) and Figure 11(f). Figure 11(d) indicates that machine m_3 always keeps unchanged whatever state buffer B_4 is when buffer B_3 has a



(a) $n_4 = 1$ and $\tau_{4,1} = 1$ (b) $n_4 = 3$ and $\tau_{4,1} = 4$ (c) $n_4 = 5$ and $\tau_{4,1} = 6$

(d) $n_3 = 1$ and $\tau_{3,1} = 2$ (e) $n_3 = 3$ and $\tau_{3,1} = 4$ (f) $n_3 = 5$ and $\tau_{3,1} = 6$

Figure 11. Control policy obtained from the decomposition-based control for machine m_3.

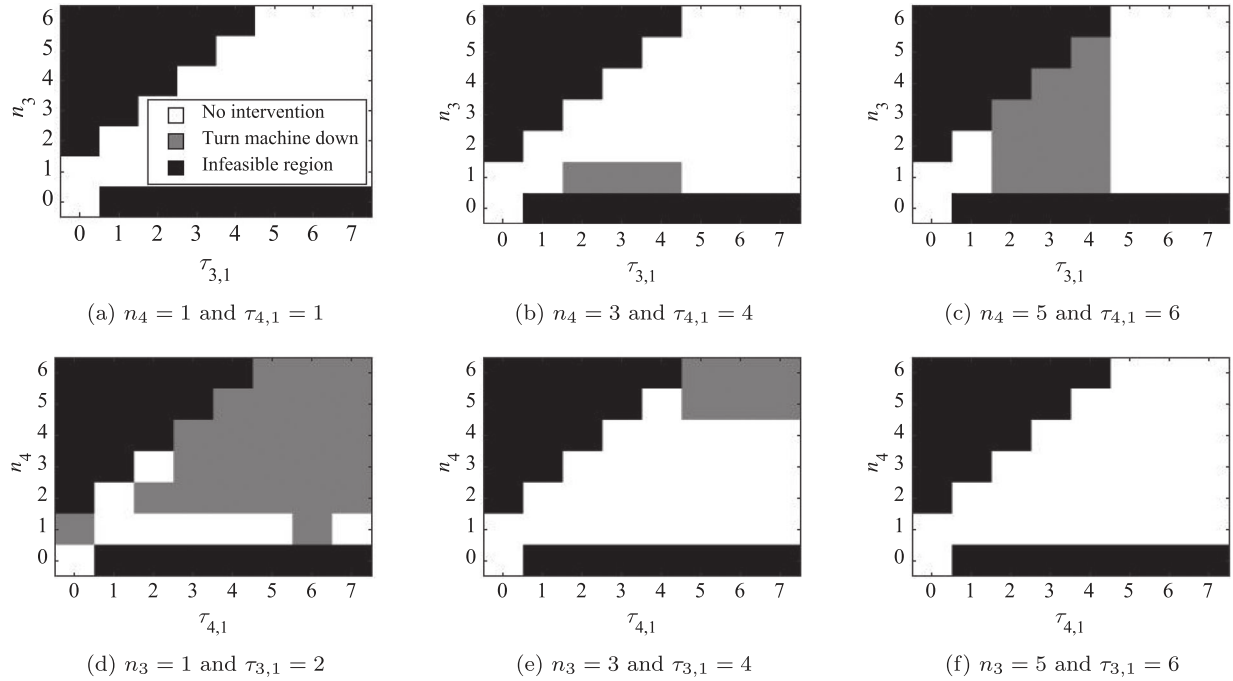
low buffer occupancy and a small residence time of the first part. In contrast, machine m_3 is always turned down whatever state buffer B_4 is when buffer B_3 has a high buffer occupancy and a large residence time of the first part, which is shown in Figure 11(f). Figure 11(e) indicates that when buffer B_3 has a median buffer occupancy and a median residence time of the first part, machine m_3 is turned down only when buffer B_4 reaches a high buffer occupancy.

From Figure 11, three main features related to the decision making of machine m_3 can be observed. First, buffer occupancy of buffer B_3 and buffer B_4 plays an important role in machine m_3's decision making. Machine m_3 is more likely to be turned down when buffer B_3 and/or buffer B_4 have/has a high buffer occupancy. Such actions prevent the subsystem from a potential scrap by turning machine m_3 down and stopping new parts from entering the subsystem. Since the buffer occupancy is high, the action that turns machine m_3 down will not cause too much loss of production. Second, buffer B_3 has a larger influence on machine m_3's decision making than buffer B_4. It can be observed that the lookup tables shown in Figure 11(a), Figure 11(b) and Figure 11(c) does not change too much mutually, while the lookup tables in Figure 11(d), Figure 11(e) and Figure 11(f) shows a large difference. Machine m_3 is closer to buffer B_3 than buffer B_4, and it explains why buffer B_3 has a larger influence on machine m_3's decision making. Finally, the control policy is not sensitive to the residence time of the first part in either buffer B_3 or buffer B_4, and the boundary that separates the white region and gray region does not show the property of monotonicity.

Figure 12 presents the control policy for machine m_4. In each plot, the black blocks represent the infeasible regions. Within the feasible regions, the white blocks indicate the action that no intervention is given, whereas the gray block indicates the action to turn machine m_4 down. We first fix the state of buffer B_4. When buffer B_4 has a low buffer occupancy and a small residence time of the first part, machine m_4 is always kept unchanged. In such a situation, there is no risk of scrap from buffer B_4, and letting machine m_4 work can potentially increase the production rate. When buffer B_4 has a median buffer occupancy and a median residence time of the first part, machine m_4 is turned down when buffer B_3 has a small buffer occupancy and a small residence time for the first part. This action can decrease the risk of scrap from buffer B_4 without increasing the risk of scrap from buffer B_3. It can be observed that the feasible region with $\tau_{3,1}$ smaller than two is white, and the actions in those states in fact do not make any difference. The reason for this behavior is that machine m_4 cannot produce a part from buffer B_3 when the residence time of the first part in buffer B_3 is smaller than $T_{3,min}$. When buffer B_4 has a high buffer occupancy and a large residence time of the first part, machine m_4 produces when the residence time of the first part in buffer B_3 is large. In this case, machine m_4 has to do a trade-off by considering scrap from both buffer B_3 and buffer B_4.

Then, we fix the state of buffer B_3. Figure 12(d) suggests that machine m_4 is more likely to be turned down when B_3 has a low buffer occupancy and a small residence time of the first part. Figure 12(e) indicates that, in the cases that B_3 has a median buffer occupancy and a median residence time of the first part, machine m_4 is turned down only when buffer occupancy of buffer B_4 is high. Machine m_4 does so due to the trade-off of scrap in buffer B_3 and buffer B_4. Figure 12(f) suggests that machine m_4 is unchanged whatever state buffer B_4 is when buffer B_3 has a high buffer occupancy and a large residence time of the first part.

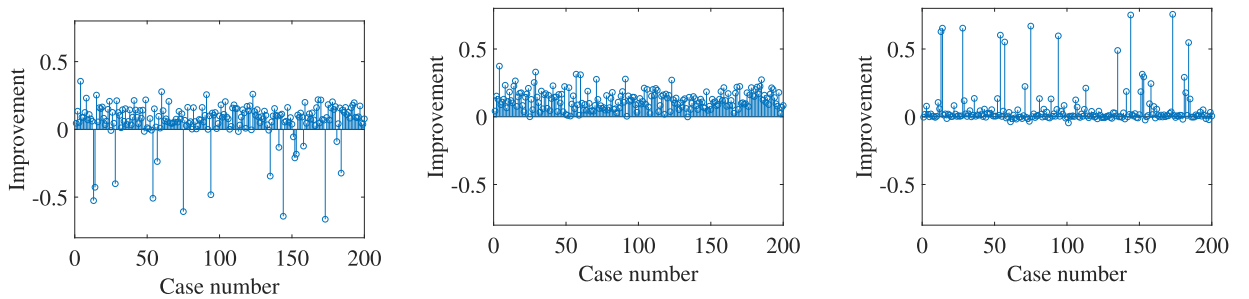
When we compare Figure 11 with Figure 12, we can observe that the action on machine m_3 and the action on machine m_4 play different roles in improving the systems.



(a) $n_4 = 1$ and $\tau_{4,1} = 1$ (b) $n_4 = 3$ and $\tau_{4,1} = 4$ (c) $n_4 = 5$ and $\tau_{4,1} = 6$

(d) $n_3 = 1$ and $\tau_{3,1} = 2$ (e) $n_3 = 3$ and $\tau_{3,1} = 4$ (f) $n_3 = 5$ and $\tau_{3,1} = 6$

Figure 12. Control policy obtained from the decomposition-based control for machine m_4.



(a) Improvement of RL control (b) Improvement of decomposition-based control (c) Improvement of decomposition-based control minus improvement of RL control

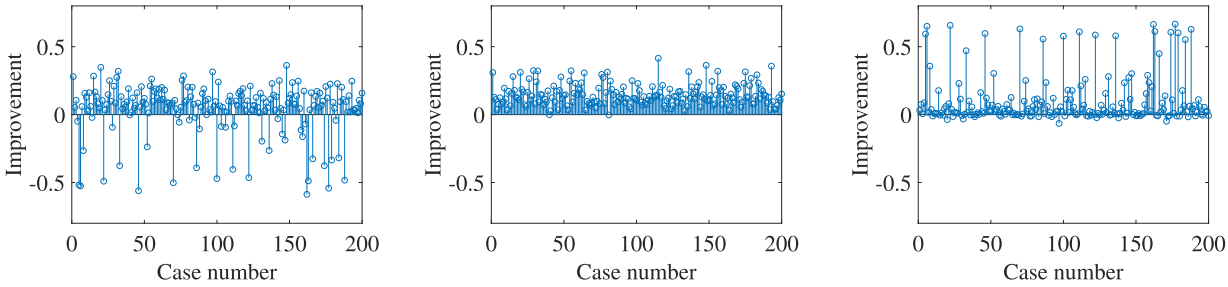
Figure 13. Improvement of average reward for multi-stage lines with five machines. The average reward without control is 0.525.

Machine m_3 is the first machine of its subsystem, and it decides to allow a part to enter the subsystem or prevent a part from entering the subsystem. Machine m_4 is in the middle of buffer B_3 and buffer B_4. Its responsibility is to balance the risk of scrap from buffer B_3 and buffer B_4.

5.3. Simulation experiment with randomly selected parameter settings

To evaluate the performance of the decomposition-based control in a more general sense, we randomly select parameter settings of a multi-stage line from a predefined range and compare the performance measures of a system without control, with RL control and with decomposition-based control. The range of parameter settings is given in Equation (39). Parameters are selected with equal probability from the range. A total of 200 parameter settings are randomly selected for multi-stage lines with $D = 5$ machines and $D = 7$ machines, respectively. In each parameter setting, the average steady-state performance measures of a system without control, with RL control and with decomposition-based control are obtained through simulation and compared mutually. The simulation starts with empty buffers. The average reward of each cycle from cycle 201 to cycle 400 among 100 repeats, which is the mean value of 20 000 observations, is calculated and compared. In the decomposition-based control, multi-stage line is decomposed into three-machine-two-buffers subsystems.

The results of the simulation experiment for the multi-stage lines with five machines and seven machines are shown in Figure 13 and in Figure 14, respectively. Figure 13(a) and Figure 13(b) show the improvement of reward by RL control and the decomposition-based control for the multi-stage line with five machines, respectively. In most cases among 200 random parameter settings, the RL control can improve the system, but it could happen in some cases that the RL control makes the performance worse. In contrast, the decomposition-based control is more robust, and all 200 cases can be improved. Figure 13(c) shows a



(a) Improvement of RL control

(b) Improvement of decomposition-based control

(c) Improvement of decomposition-based control minus improvement of RL control

Figure 14. Improvement of average reward for multi-stage lines with seven machines. The average reward without control is 0.464.

Table 1. Average reward of different methods.

D	*Average reward without control*	*Average reward with decomposition-based control*	*Relative improvement of decomposition-based control(%)*	*Average reward with RL control*	*Relative improvement of RL control(%)*
5	0.525	0.648	23.4	0.588	12
7	0.464	0.608	31.0	0.506	9.1
9	0.403	0.563	39.7	–	–
11	0.385	0.543	41.0	–	–

pairwise comparison where the improvement of decomposition-based control minus the improvement of RL control for each case is presented, and the result suggests that the decomposition-based control outperforms the RL control. Considering the average reward without control is 0.525, such an improvement is significant. The same comparison is performed for the multi-stage line with seven machines as well. Figure 14(a), compared with Figure 13(a), shows more negative improvement. It suggests that as the number of machines increases the RL control is more likely to fail to work. In contrast, Figure 14(b) suggests that the decomposition-based control can still maintain a good performance. Figure 14(c), compared with Figure 13(c), shows that the strength of decomposition-based control over the RL control is more significant as the number of machines increases. The average reward without control is 0.464, and it shows a significant improvement of the decomposition-based control.

The control methods are developed with MATLAB and run on a server with Intel(R) Core(TM) i7-5930K CPU, sufficiently large RAM and Linux operating system. It takes time to perform training for RL control and perform the aggregation procedure of decomposition-based control. When there are five machines, the average computing time is 1037.8 seconds for RL control and 34.6 seconds for decomposition-based control. As the total number of machines increases to seven, the average computing time is 14 506.3 seconds for RL control and 70.8 seconds for decomposition-based control. The result suggests that the decomposition-based control is much more computationally efficient than the RL control. When the number of machines increases, the computing time of the RL control increases much faster than the decomposition-based control.

Transfer lines with more machines are tested, and the result is summarized in Tables 1 and 2. Table 1 provides the reward of each method under each setting. The decomposition-based control shows a good performance and also outperforms the RL control and the case under no control. The computing time is presented in Table 2. The computing time of the aggregation procedure of the decomposition-based control is much smaller than the training time of RL control and also much less sensitive to the number of machines than the RL control.

Table 2. Computing time of different methods (seconds).

D	*Aggregation procedure of decomposition-based control*	*Training of RL control*
5	34.6	1037.8
7	70.8	14 506.3
9	170.7	–
11	211.6	–

6. Conclusions and future work

In this article, a multi-stage Bernoulli transfer line with residence time constraints is formulated. Due to a large state space of the production line, it is difficult to perform real-time control according to system state. The decomposition-based control is proposed to address the problem. The simulation experiment suggests that the proposed method can improve system performance. Compared with a general-purpose reinforcement learning-based control method, the decomposition-based control can achieve a better system performance improvement and a significant reduction in computing time. It thus provides production engineers with an effective and quantitative tool to perform real-time control of production lines with residence time constraints.

In the future, work can be directed to investigating transfer lines with different structures, such as distributed system and assembly systems. In addition, it is worth studying

decomposition-based control in a manufacturing environment with more general machine reliability models.

Funding

This work is supported by the U.S. National Science Foundation under Grant CMMI-1922739.

Notes on contributors

Feifan Wang received a bachelor's degree from the Department of Industrial Engineering, Zhejiang University of Technology, Hangzhou, China, in 2013, and a master's degree from the Department of Industrial and Systems Engineering, Zhejiang University, Hangzhou, China, in 2016. He is currently pursuing a PhD degree with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. His research interests include the modeling, analysis, and control of production systems. He won the best student paper award in IEEE CASE 2019.

Feng Ju is an assistant professor with the School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. He received a BS degree from Shanghai Jiao Tong University, Shanghai, China, in 2010, and an MS degree in electrical and computer engineering and PhD degree in industrial and systems engineering from the University of Wisconsin, Madison, WI, USA, in 2011 and 2015, respectively. His current research interests include modeling, analysis, continuous improvement, and optimization of manufacturing systems. Dr. Ju is also a member of the Institute for Operations Research and the Management Sciences, Institute of Industrial and Systems Engineers, and Institute of Electrical and Electronics Engineers. He was a recipient of multiple awards, including the best paper award in IFAC MIM, best student paper award in IEEE CASE, and best student paper finalist in IEEE CASE and IFAC INCOM.

ORCID

Feng Ju http://orcid.org/0000-0003-3452-0795

References

Amorim, P., Meyr, H., Almeder, C. and Almada-Lobo, B. (2013) Managing perishability in production-distribution planning: A discussion and review. *Flexible Services and Manufacturing Journal*, **25**, 389–413.

Angius, A., Colledani, M., Horváth, A. and Gershwin, S.B. (2016) Analysis of the lead time distribution in closed loop manufacturing systems. *IFAC-PapersOnLine*, **49**, 307–312.

Barbosa, J., Leitão, P., Adam, E. and Trentesaux, D. (2015) Dynamic self-organization in holonic multi-agent manufacturing systems: The adacor evolution. *Computers in Industry*, **66**, 99–111.

Bertsekas, D.P. (2018) Feature-based aggregation and deep reinforcement learning: A survey and some new implementations. *IEEE/CAA Journal of Automatica Sinica*, **6**, 1–31.

Bertsekas, D.P. (2019) *Reinforcement Learning and Optimal Control*, Athena Scientific, Belmont, MA.

Cao, P. and Xie, J. (2015) Optimal control of an inventory system with joint production and pricing decisions. *IEEE Transactions on Automatic Control*, **61**, 4235–4240.

Gershwin, S.B. (1994) *Manufacturing Systems Engineering*, Prentice Hall, Englewood Cliffs, NJ.

Giret, A., Garcia, E. and Botti, V. (2016) An engineering framework for service-oriented intelligent manufacturing systems. *Computers in Industry*, **81**, 116–127.

Han, B. and Kim, D.S. (2017) Moisture ingress, behavior, and prediction inside semiconductor packaging: A review. *Journal of Electronic Packaging*, **139**, 010802–1–010802–11.

Jaegler, Y., Jaegler, A., Burlat, P., Lamouri, S. and Trentesaux, D. (2018) The CONWIP production control system: A systematic review and classification. *International Journal of Production Research*, **56**, 5736–5757.

Jia, Z., Zhang, L., Arinez, J. and Xiao, G. (2016) Performance analysis for serial production lines with Bernoulli machines and real-time WIP-based machine switch-on/off control. *International Journal of Production Research*, **54**, 6285–6301.

Ju, F., Li, J. and Deng, W. (2016) Selective assembly system with unreliable Bernoulli machines and finite buffers. *IEEE Transactions on Automation Science and Engineering*, **14**, 171–184.

Ju, F., Li, J. and Horst, J.A. (2017) Transient analysis of serial production lines with perishable products: Bernoulli reliability model. *IEEE Transactions on Automatic Control*, **62**, 694–707.

Ju, F., Li, J., and Horst, J.A. (2015) Transient analysis of Bernoulli serial line with perishable products. *IFAC-PapersOnLine*, **48**, 1670–1675.

Kang, N., Ju, F. and Zheng, L. (2016) Transient analysis of geometric serial lines with perishable intermediate products. *IEEE Robotics and Automation Letters*, **2**, 149–156.

Lee, J.H., Li, J. and Horst, J.A. (2017) Serial production lines with waiting time limits: Bernoulli reliability model. *IEEE Transactions on Engineering Management*, **65**, 316–329.

Lee, J.H., Zhao, C., Li, J. and Papadopoulos, C.T. (2018) Analysis, design, and control of Bernoulli production lines with waiting time constraints. *Journal of Manufacturing Systems*, **46**, 208–220.

Leitão, P. (2009) Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, **22**, 979–991.

Li, J. and Meerkov, S.M. (2009) *Production Systems Engineering*, Springer Science & Business Media, New York, NY.

Liberopoulos, G. and Tsarouhas, P. (2005) Reliability analysis of an automated pizza production line. *Journal of Food Engineering*, **69**, 79–96.

Lu, Y. and Ju, F. (2017) Smart manufacturing systems based on cyber-physical manufacturing services (cpms). *IFAC-PapersOnLine*, **50**, 15883–15889.

Lu, Y., Riddick, F. and Ivezic, N. (2016) The paradigm shift in smart manufacturing system architecture, in *IFIP International Conference on Advances in Production Management Systems*, Springer, Cham, Switzerland. pp. 767–776.

Naebulharam, R. and Zhang, L. (2014) Bernoulli serial lines with deteriorating product quality: performance evaluation and system-theoretic properties. *International Journal of Production Research*, **52**, 1479–1494.

Nahmias, S. (1982) Perishable inventory theory: A review. *Operations Research*, **30**, 680–708.

Papadopoulos, C.T., Li, J. and O'Kelly, M.E. (2019) A classification and review of timed Markov models of manufacturing systems. *Computers & Industrial Engineering*, **128**, 219–244.

Raafat, F. (1991) Survey of literature on continuously deteriorating inventory models. *Journal of the Operational Research Society*, **42**, 27–37.

Shi, C. and Gershwin, S.B. (2012) Part waiting time distribution in a two-machine line. *IFAC Proceedings Volumes*, **45**, 457–462.

Shi, C. and Gershwin, S.B. (2015) Lead time distribution of three-machine two-buffer lines with unreliable machines and finite buffers, in *Conference on Stochastic Models of Manufacturing and Service Operations*, pp. 211–220, University of Thessaly Press, Volos, Greece.

Shi, C. and Gershwin, S.B. (2016) Part sojourn time distribution in a two-machine line. *European Journal of Operational Research*, **248**, 146–158.

Stricker, N., Kuhnle, A., Sturm, R. and Friess, S. (2018) Reinforcement learning for adaptive order dispatching in the semiconductor industry. *CIRP Annals*, **67**, 511–514.

Thürer, M., Fernandes, N.O., Stevenson, M., Qu, T. and Li, C.D. (2019) Centralised vs. decentralised control decision in card-based control systems: Comparing kanban systems and cobacabana. *International Journal of Production Research*, **57**, 322–337.

Wang, F. and Ju, F. (2020) Transient and steady-state analysis of multi-stage production lines with residence time limits. *IEEE Transactions on Automation Science and Engineering*. doi:10.1109/TASE.2020.2979179

Wang, F., Ju, F. and Kang, N. (2019) Transient analysis and real-time control of geometric serial lines with residence time constraints. *IISE Transactions*, **51**, 709–728.

Wang, F., Ju, F. and Lu, Y. (2017) A study on performance evaluation and status-based decision for cyber-physical production systems, in *13th IEEE Conference on Automation Science and Engineering (CASE)*, IEEE Press, Piscataway, NJ, pp. 1000–1005.

Wang, F., Lu, Y. and Ju, F. (2018) Condition-based real-time production control for smart manufacturing systems, in *IEEE 14th International Conference on Automation Science and Engineering*, IEEE Press, Piscataway, NJ, pp. 1052–1057.

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A. and Kyek, A. (2018) Deep reinforcement learning for semiconductor production scheduling, in: *2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, IEEE Press, Piscataway, NJ, pp. 301–306.

Yang, H., Kumara, S., Bukkapatnam, S.T. and Tsung, F. (2019) The internet of things for smart manufacturing: A review. *IISE Transactions*, **51**, 1190–1216.

Zhang, L., Wang, C., Arinez, J. and Biller, S. (2013) Transient analysis of Bernoulli serial lines: Performance evaluation and system-theoretic properties. *IIE Transactions*, **45**, 528–543.