Due-date assignment for multi-server multi-stage assembly systems

Saeed Yaghoubi

Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Published online: 09 Jul 2013.


To link to this article: http://dx.doi.org/10.1080/00207721.2013.815826

Please scroll down for article
Due-date assignment for multi-server multi-stage assembly systems

Saeed Yaghoubi*

Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

(Received 11 January 2013; final version received 7 June 2013)

In this paper, we attempt to present a constant due-date assignment policy in a multi-server multi-stage assembly system. This system is modelled as a queuing network, where new product orders are entered into the system according to a Poisson process. It is assumed that only one type of product is produced by the production system and multi-servers can be settled in each service station. Each operation of every work is operated at a devoted service station with only one of the servers located at a node of the network based on first come, first served (FCFS) discipline, while the processing times are independent random variables with exponential distributions. It is also assumed that the transport times between each pair of service stations are independent random variables with generalised Erlang distributions. Each product's end result has a penalty cost that is some linear function of its due date and its actual lead time. The due date is calculated by adding a constant to the time that the order enters into the system. Indeed, this constant value is decided at the beginning of the time horizon and is the constant lead time that a product might expect between the time of placing the order and the time of delivery. For computing the due date, we first convert the queuing network into a stochastic network with exponentially distributed arc lengths. Then, by constructing an appropriate finite-state continuous-time Markov model, a system of differential equations is created to find the manufacturing lead-time distribution for any particular product, analytically. Finally, the constant due date for delivery time is obtained by using a linear function of its due date and minimising the expected aggregate cost per product.

Keywords: assembly system; due-date assignment; Markov processes; queuing

1. Introduction

Some producers of goods such as large-scale machine tools (Drex1 and Kolisch 1996), airplanes (Chao and Graves 1998), trains and ships (Lee, Lee, Park, Hong, and Lee 1997), employ a make-to-order (MTO), engineer-to-order (ETO) or assembly-to-order (ATO) system in production planning. In such organisation, not only the processing times of service stations are stochastic, in addition the product orders arrive at the system dynamically over the time horizon. The make-to-order system is often a complex assembly system due to having several stages of manufacture and assembly and in such situations, in addition to a competitive price, obtaining a competitive delivery lead time is typically important for overcoming the customer orders. The multi-stage assembly system can be modelled as an open queuing network, where each service station, located at a node of the queuing network, represents a manufacturing or assembly operation.

In this paper, it is assumed that only one type of product is produced by the production system and the new product orders, including all their operations, are entered into the system according to a Poisson process. Each work (product) includes a number of separate raw parts, which should be processed and assembled together. Each separate part of the product arrives at the first service station and continues its routing sequence of manufacturing operations. The final product leaves the production system in its finished form after assembling the individual parts and passing some other manufacturing and/or assembly operations. Each operation of a work is operated at a devoted service station with only one of the servers located at the node of the network based on a first come, first served (FCFS) discipline, while the processing times are independent random variables with exponential distributions.

An implicit hypothesis in the literature is that the transit time in buffers is null, i.e., the part that leaves a service station is assumed to be instantaneously available for the next service station. The time needed to pass the intersection buffers may be much greater than the service time and there is no reason to claim that its impact on the system will be negligible (see Azaron and Kianfar 2006). Consequently, in this paper, it is assumed that the transport times between each pair of service stations are independent random variables with generalised Erlang distributions. On the other hand, it is also assumed that each product's end result has a penalty cost that is some linear function of its due date and its actual lead time. The due date is calculated by adding a constant to the time that the order enters into the system. Indeed, this constant value is decided at the beginning of the time horizon and is the constant lead time that a product might expect between the time of placing the order and the time of delivery.

*Email: yaghoubi@iust.ac.ir

© 2013 Taylor & Francis
In a multi-stage assembly system, the parts after passing their routing sequences of manufacturing operations have to wait not only for receiving service but also for the other parts to arrive before the service station can begin processing. Therefore, the manufacturing lead time in a multi-stage assembly system will be equal to the length of the longest path in the queuing network. Givoli and Fu (1999) explained that the assembly network models are intractable by exact analytical models.

The queuing theory is extensively applied in the manufacturing system, see Papadopoulos and Heavey (1996) and Givoli and Fu (1999). Nof, Wilhelm, and Warnecke (1997) also showed the use of queuing models in assembly systems.

Manufacturing systems, based on the physical layout of the manufacturing resource and also the type of material flow in the systems, are categorised as follows: job shop, flow line, flexible manufacturing systems (FMS) and assembly systems (Givoli and Fu 1999). In this article, we briefly review the literature of assembly systems.

Harrison (1973) assumed an assembly system with \( k(\geq 2) \) input stream and infinite buffers. In these conditions, he proved that the length of the queue in front of any assembly station does not obtain a stable steady state. Latouche and Neuts (1980) revealed then that the waiting times in service stations are stable, if the arrival of works is Poisson and the processing times at the assembly service stations are exponential. Also, Baybars (1986) and Ghosh and Gannon (1989) extensively analysed the mass production assembly systems.

Bhat (1986) presented a model of the assembly station with two input streams under Markovian assumptions and a finite capacity. Lipper and Sengupta (1986) presented an approximate analysis of the assembly-like queue, where customers from different classes arrive to a single server queuing station according to independent Poisson processes with equal arrival rates. Processing times at the station are exponentially distributed and each customer class has a different buffer, while the size of all buffers is equal and finite. A similar work was also done by Hopp and Simon (1989).

Liu and Buzacott (1990) proposed an approximate model for assembly systems with finite inventory banks. Di Mascolo, David, and Dallery (1991) proposed an assembly system with finite capacity, in which processing times are fixed and machines are unreliable. Baker and Powell (1995) predicted the throughput of balanced three station assembly systems. Liu and Yuan (2001) investigated an unreliable assembly system with finite buffer, in which different types of components were executed by two separate work centres before merging into an assembly station with random breakdown. It was further extended by Yuan and Liu (2005).

Hemachandra and Eedupuganti (2003) presented a finite capacity fork–join queuing model for open assembly systems with arrival and departure synchronisations by using and enumerating the state space and obtaining the steady state probabilities under exponential assumptions. The methods for throughput analysis and bottleneck identification in assembly systems with non-exponential machines and finite buffer were proposed by Ching, Meerkov, and Zhang (2008). Moreover, Manitz (2008) proposed an approximation procedure for determining the throughput in assembly line with a synchronisation constraint, generally distributed processing times. Recently, the assembly system has been analysed by Louly, Dolgui, and Al-Ahmari (2012), Chang, Su, Yang, and Weng (2012) and Ho, Pan, and Hsiao (2012).

In an assembly-to-order (ATO) system, managers are repeatedly met with the task of determining a new product’s due date, which must compete with other products already in progress or expected (forecasted) to start in the future. In the literature, two approaches exist for due-date assignment. One approach identifies the most desirable due-date assignment method, while the second approach deals with the determination of optimal due dates. The first pioneering researchers in studying due-date assignment for production systems are Elvers (1973), Jones (1973), Rochette and Sadowski (1976) and Weeks (1979), who analysed this problem by using a simulation study. Eilon and Chowdhury (1976) and Weeks and Fryer (1977) revealed that the due-date assignment procedure incorporating expected job flow times execute better than rules based only on job content.

Seidman and Smith (1981) presented an algorithm to assign due date for a dynamic job shop, where production times are randomly distributed and each job has a penalty cost that is some non-linear function of its due date and its actual completion time. Furthermore, Seidmann, Panwalkar, and Smith (1981) and Panwalkar, Smith, and Seidmann (1982) studied a single machine with penalties for earliness and tardiness and penalties associated with the assignment of due dates. Dumond and Mabert (1988) addressed the problem of establishing due dates for projects that require limited resources, in an environment where new projects arrive continuously and randomly over time. A good survey of the state-of-the-art of due-date assignment and scheduling research was also presented by Cheng and Gupta (1989) and Gordon, Proth, and Chu (2002).

In addition, Ahmed and Fisher (1992) investigated the effects and interactions of due-date assignment, job order release and sequencing with the simulation model of a dynamic five-machine job shop in which early shipments are prohibited. The problem of assigning a common due date to a set of jobs and scheduling them on a single machine was also analysed by Biskup and Jahneke (2001). Cheng, Kang, and Ng (2004) determined an optimal combination of the due date and schedule so as to minimise the sum of due date, earliness and tardiness penalties. Xia, Chen, and Yue (2008) considered the due-date assignment and sequencing for multiple jobs in a single machine shop.
Moreover, Gordon and Strusevich (2009) analysed single-machine scheduling and due-date assignment problems in which the processing time of a job depends on its position in a processing sequence, while Mosheiov and Sarig (2009) researched the due-date assignment on uniform machines. Recently, Steiner and Zhang (2011), Vinod and Sridharan (2011) and, T’kindt and Croce (2012) also analysed due-date assignment for production systems.

On the other hand, Azaron and Kianfar (2009) proposed the constant due-date assignment policy in a multi-stage assembly system, in which processing times are independent random variables with exponential distributions, product orders are generated according to a Poisson process, the transport times between the service stations are as random variables with generalised Erlang distributions. Moreover, they assumed that either one or infinite servers are located in each service station, while this assumption is a restricting assumption. In this paper, we relax this restricted assumption and assume that multi-servers can be settled in each service station, which is the main difference between this paper under consideration and the previous paper, proposed by Azaron and Kianfar (2009). Reviewing the above-mentioned investigations makes it clear that no analytical method has already been developed to study due-date assignment in multi-server multi-stage assembly networks.

For modelling a multi-server multi-stage assembly system, we first transform the corresponding network of queues into an appropriate stochastic network with exponentially distributed arc lengths. Then, by constructing an appropriate finite-state continuous-time Markov model, a system of differential equations is created to get the approximate manufacturing lead-time distribution for any particular product, analytically. Finally, the constant due date for delivery time is obtained by using a linear function of its due date and minimising the expected aggregate cost per product.

In the problem under consideration, the objective is to find a constant due date for products in multi-server multi-stage assembly system in order to obtain a given criterion based on the due date and the completion times of jobs.

It is worth mentioning that Azaron, Katagiri, Kato, and Sakawa (2006) suggested a multi-objective model to optimally control the service rates of the manufacturing and the assembly operation in a dynamic multi-stage assembly system, in which processing times are independent random variables with exponential distributions, product orders are generated according to a Poisson process, and in each service station, one server is located. Also, they considered the transport times between the service stations with exponential distributions. Then, Azaron and Kianfar (2006) extended this problem by considering the transport times between the service stations with generalised Erlang distributions, one or infinite number of servers (machines) in service stations, and using an interactive method for solving it. Perkgoz, Azaron, Katagiri, Kato, and Sakawa (2007) also extended this problem by applying genetic algorithms for solving it. Recently, Yaghoubi, Noori, and Pourdadashi-komachali (2012) presented two multi-objective models to optimally control the lead time in multi-server multi-stage assembly system by considering the server allocation problem and also service rate control problem.

This paper is composed of five sections. The remainder of the paper is organised as follows: in Section 2, we obtain an approximation of the manufacturing lead time in multi-server multi-stage assembly system by employing a finite-state continuous-time Markov process. In Section 3, we assign constant due date to each product by minimising the expected aggregate cost. We also solve two numerical examples in Section 4, and the conclusion is given in Section 5.

2. Dynamic multi-server multi-stage assembly system

In this section, the multi-server multi-stage assembly system is modelled as a queuing network using a continuous Markov process and an analytical method to compute the approximate distribution function of lead time is also presented. For this purpose, we use the method presented by Kulkarni and Adlakha (1986), because this method is an analytical approach, simple, and an easy approach to implement through a computer, and is computationally stable. It is assumed that this system is represented as a queuing network and new product orders, including all their operations, arrive at the system according to a Poisson process at the rate of \( \lambda \). Furthermore, each operation of work is executed at a devoted service station settled in the node of a network based on the FCFS discipline, where the processing times are exponentially distributed.

Moreover, the arrival stream of the works at each service station is followed according to a Poisson process at the rate of \( \lambda \). It is supposed that the number of servers in node \( a \) is \( m_a \) and service-processing times in service station \( a \) are exponentially distributed with the rate of \( \mu_a \). So, the node \( a \) treats as an \( M/M/m_a \) model. The flow chart of our proposed method for modelling the multi-server multi-stage assembly system is shown in Figure 1.

In continuation, the stages of our proposed method for multi-server multi-stage assembly system as a continuous-time Markov process are extensively explained:

**Step 1.** For each service station, compute the density function of the sojourn time (waiting time plus processing time).

**Step 1.1.** If \( m_a = 1 \), then the queuing system would be an \( M/M/1 \) queue, and the density function of time spent at the service station \( a \) (\( w_a(t) \)) would be exponential with parameter \( \mu_a - \lambda \), therefore, \( w_a(t) \) is calculated as follows:

\[
 w_a(t) = (\mu_a - \lambda) e^{-(\mu_a - \lambda) t} t > 0, \quad \text{if } m_a = 1. \quad (1)
\]
Step 1.2. If $m_a = \infty$, then the queuing system is $M/M/\infty$, and the density function of the time spent in the service station $a$ would be exponential with parameter $\mu_a$, therefore, $w_a(t)$ is obtained as follows:

$$w_a(t) = \mu_a e^{-\mu_a t}, \quad \text{if } m_a = \infty. \quad (2)$$

Step 1.3. If $1 < m_a < \infty$, then the queuing system is $M/M/m_a$, and the density function of time spent in the service station $a$ approximately would be two series of exponential arcs with parameters $\frac{m_a \mu_a - \lambda}{\rho_a}$ and $\frac{m_a \mu_a}{m_a - 1}$, where $\rho_a = \frac{\lambda}{m_a \mu_a}$. Therefore, $w_a(t)$ approximately is calculated as follows: (for more details see Yaghoubi et al. 2012)

$$w_a(t) \approx \left( \frac{m_a \mu_a - \lambda}{\rho_a} - \frac{m_a \mu_a}{m_a - 1} \right) \cdot e^{-\frac{m_a \mu_a - \lambda}{\rho_a}} \cdot e^{-\frac{m_a \mu_a}{m_a - 1} t \cdot t > 0. \quad (3)$$
Step 2. Transform the queuing network into a substitute stochastic network with exponentially distributed arc lengths.

Step 2.1. Considering the queuing network, substitute each node with a stochastic arc whose length is equal to the sojourn time in the corresponding service station.

For this purpose, node $a$ in the queuing network should be replaced with a stochastic arc. Assume $b_1, b_2, \ldots, b_n$ are the incoming arcs to node $a$ and $d_1, d_2, \ldots, d_m$ are the outgoing arcs from it in the queuing network. Then, node $a$ is substituted by arc $(v, w)$, whose length is equal to the sojourn time in the service station $a$. Furthermore, all arcs $b_1, b_2, \ldots, b_n$ terminate at node $v$, while all arcs $d_1, d_2, \ldots, d_m$ begin from node $w$ (for more details, see Azaron and Modarres 2005).

Step 2.2. Replace each arc including the sojourn time of the corresponding service station with proper exponentially distributed arcs, if a change is required.

If one or infinite servers be in the service station $a$, then the length of arc would be exponential with parameters $\mu_a - \lambda_a$ and $\mu_a$, respectively, and the corresponding arc would not be changed. But, if several servers $(1 < m_a < \infty)$ be in the service station $a$, then the corresponding arc would be substituted with two series of exponential arcs with the parameters $\frac{m_a \mu_a - \lambda_a}{\mu_a}$ and $\frac{m_a \mu_a}{\mu_a - 1}$.

Step 2.3. Replace each transport time with proper exponentially distributed arc lengths.

Assume that the transport time of service station $a$ is distributed according to a generalised Erlang distribution of order $n_a$, with the infinitesimal generator matrix, $G_a$, obtained as follows:

$$G_a = 
\begin{bmatrix}
-\lambda_{a_1} & \lambda_{a_1} & 0 & \cdots & 0 & 0 \\
0 & -\lambda_{a_2} & \lambda_{a_2} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -\lambda_{a_k} & \lambda_{a_k} \\
0 & 0 & 0 & \cdots & 0 & 0 
\end{bmatrix}. \quad (4)
$$

This generalised Erlang arc due to decomposing into $n_a$ exponential serial arcs with the parameters $\lambda_{a_1}, \lambda_{a_2}, \ldots, \lambda_{a_k}$, can be decomposed into $n_a$ series of exponential arcs with the parameters $\lambda_{a_1}, \lambda_{a_2}, \ldots, \lambda_{a_k}$. After replacing all such arcs with the proper exponential serial arcs, a new stochastic network with exponentially distributed arc lengths is obtained.

Step 3. Obtain the distribution function of manufacturing the lead time.

Step 3.1. Determine the state space of the system.

For this purpose, let $G = (V, A)$ be the new stochastic network, obtained in Step 2.3, in which $V$ and $A$ represent, respectively, the set of nodes and the set of arcs of the stochastic network $G$. Note that the words operation and arc will be applied equivalently along this paper under consideration. Let $s$ and $t$ be the source and sink nodes in stochastic network $G$, respectively, and length of arc $a \in A$ be a random variable that is exponentially distributed with parameter $\gamma_a$. For $a \in A$, the starting node and the ending node of arc $a$, are denoted as $\alpha(a)$ and $\beta(a)$, respectively. Henceforth, in this section, we analyse the stochastic network $G$, obtained in Step 2.3, to determine a continuous-time Markov process with finite state space.

Definition 1: Let $I(v)$ be the set of arcs ending at node $v$ and $O(v)$ be the set of arcs starting at node $v$ in a stochastic network $G$, which are defined as follows (see Kulkarni and Adlakha 1986):

$$I(v) = \{a \in A : \beta(a) = v\} \ (v \in V),$$

$$O(v) = \{a \in A : \alpha(a) = v\} \ (v \in V). \quad (5)$$

Definition 2: For $X \subset V$ such that $s \in X$ and $t \in \bar{X} = V - X$, an $(s, t)$ cut is defined as follows:

$$(X, \bar{X}) = \{a \in A : \alpha(a) \in X, \beta(a) \in \bar{X}\}. \quad (6)$$

An $(s, t)$ cut $(X, \bar{X})$ is denominated a uniformly directed cut (UDC), if $(X, \bar{X}) = \phi$, i.e. there are not two arcs in the cut belonging to the same path in the stochastic network. Each UDC is clearly a set of arcs, in which the starting node of each arc belongs to $X$ and the ending node of each arc belongs to $\bar{X}$.

Example 1: Consider the network of queues with exponential sojourn times shown in Figure 2(a) taken from

![Figure 2](image_url)
Perkgoz et al. (2007). According to the definition, the UDCs of this network are (1,2,3), (1,3,4), (1,2,5), (1,4,5), (1,6), (7) and (8), by considering the stochastic network depicted in Figure 2(b).

**Definition 3:** An \((E,F)\), subsets of \(A\), is defined as an admissible two-partition of a UDC \(D\) if \(D = E \cup F\) and \(E \cap F = \phi\), and also \(I(\beta(a)) \not\subset F\) for any \(a \in F\).

Consider Example 1 again. As mentioned, \((1, 4, 5)\) is a UDC. For example, this cut can be decomposed into \(E = \{1, 4\}\) and \(F = \{5\}\). In this case, the cut is an admissible two-partition, because \(I(\beta(3)) = \{4, 5\} \not\subset F\). However, if \(E = \{1\}\) and \(F = \{4, 5\}\), then the cut is not an admissible two-partition, because \(I(\beta(3)) = \{4, 5\} \subset F = \{4, 5\}\).

**Definition 4:** Each operation (arc) at time \(t\) can be in one and only one of the active, dormant or idle states, which are defined as follows:

1. **Active:** an operation \(a\) is active at time \(t\), if it is being performed at time \(t\).
2. **Dormant:** an operation \(a\) is called dormant at time \(t\), if it has been completed but there is at least one unfinished operation in \(I(\beta(a))\) at time \(t\).
3. **Idle:** an operation \(a\) is denominated idle at time \(t\), if it is neither active nor dormant at time \(t\).

**Definition 5:** The state of system at time \(t\) is \(X(t) = (Y(t), Z(t))\), where \(Y(t)\) and \(Z(t)\) are denoted as follows:

\[
Y(t) = \{a \in A : a \text{ is active at time } t\},
\]

\[
Z(t) = \{a \in A : a \text{ is dormant at time } t\},
\]

The set of all admissible two-partition cuts for the network is defined as \(S\) and also \(\bar{S} = S \cup \{(\phi, \phi)\}\). Note that \(X(t) = (\phi, \phi)\) presents that all operations are idle at time \(t\) and therefore the work is finished by time \(t\). It is demonstrated that \(\{X(t), t \geq 0\}\) is a finite-state absorbing continuous-time Markov process (for more details, see Kulkarni and Adlakha 1986).

**Step 3.2.** Determine a continuous-time Markov process with finite states.

A UDC is divided into \(E\) and \(F\) that contain active and dormant operations, respectively. If operation \(a\) terminates at the rate \(\gamma_a\), and \(I(\beta(a)) \not\subset F \cup \{a\}\), namely there is at least one unfinished operation in \(I(\beta(a))\), then \(E = E - \{a\}, F = F \cup \{a\}\). Furthermore, if by completing operation \(a\), all operations in \(I(\beta(a))\) become idle \((I(\beta(a)) \subset F \cup \{a\})\), then \(E' = (E - \{a\}) \cup O(\beta(a)), F' = F - I(\beta(a))\). Namely, all operations in \(I(\beta(a))\) will become idle, the successor operation of this operation, \(O(\beta(a))\), will become active. Therefore, the components of the infinitesimal generator matrix \(Q = [q(E,F),(E',F')]\), where \((E,F)\) and \((E',F')\) \(\in \bar{S}\), are obtained as follows:

\[
q((E,F),(E',F')) = \begin{cases} 
\gamma_a & \text{if } a \in E, I(\beta(a)) \not\subset F \cup \{a\}, \\ E' = E - \{a\}, F' = F \cup \{a\} \\
\gamma_a & \text{if } a \in E, I(\beta(a)) \subset F \cup \{a\}, \\ E' = (E - \{a\}) \cup O(\beta(a)), \\ F' = F - I(\beta(a)) \\
- \sum_{a \in E} \gamma_a & \text{if } E' = E, F' = F \\
0 & \text{otherwise}
\end{cases}
\]

\(\{X(t), t \geq 0\}\) is a continuous-time Markov process with finite state space \(\bar{S}\) and since \(q((\phi, \phi), (\phi, \phi)) = 0\), the work is completed. In this Markov process, all of the states, except \(X(t) = (\phi, \phi)\) that is an absorbing state, are transient. Furthermore, the states in \(\bar{S}\) should be numbered such that this \(Q\) matrix is an upper triangular one. It is assumed that the states are numbered as \(1, 2, \ldots, N = |\bar{S}|\) such that \(X(t) = (O(s), \phi)\) and \(X(t) = (\phi, \phi)\) are State 1 (initial state) and State \(N\) (absorbing state), respectively.

**Step 3.3.** Obtain a system of differential equations and solve it.

Let \(T\) be the length of the longest path or the manufacturing lead time in the new stochastic network, obtained in step 2.3. Obviously, \(T = \min\{t \geq 0 : X(t) = N|X(0) = 1\}\).

Chapman–Kolmogorov backward equations can be used to calculate \(F(t) = P(T \leq t)\). If we define

\[
P_i(t) = P(X(t) = N |X(0) = i) = i = 1, 2, \ldots, N,
\]

then \(F(t) = P_1(t)\).

The system of linear differential equations for the vector \(P(t) = [P_1(t), P_2(t), \ldots, P_N(t)]^T\) is presented as follows:

\[
P'(t) = \frac{dP(t)}{dt} = Q.P(t)
\]

where \(P'(t)\) and \(Q\) represent the derivation of the state vector \(P(t)\) and the infinitesimal generator matrix of the stochastic process \(\{X(t), t \geq 0\}\), respectively.

Let \(M\) be the modal matrix of \(Q\), an \(N \times N\) matrix whose \(N\) columns are eigenvectors of \(Q\). Let \(\alpha_1, \alpha_2, \ldots, \alpha_N\) be the eigenvalues of \(Q\), which are the diagonal elements of \(Q\) owing to its upper triangular nature. We can compute \(P(t)\) as follows:

\[
P(t) = M.e^{\alpha M}.M^{-1}.P(0),
\]
where \( e^{\Lambda t} \) is a diagonal matrix (see Luenberger 1979, for details) as follows:

\[
e^{\Lambda t} = \begin{bmatrix} e^{\alpha_1 t} & 0 & \ldots & 0 \\
0 & e^{\alpha_2 t} & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & e^{\alpha_N t} \end{bmatrix}.
\] (12)

3. Due-date assignment model

As mentioned before, Seidmann and Smith (1981) presented an algorithm to assign the due date for a dynamic job shop, where production times are randomly distributed and each job has a penalty cost that is some non-linear function of its due date and its actual completion time. Then, Azaron and Kianfar (2009) studied the constant due-date assignment policy in a multi-stage assembly system. In this paper, we attempt to compute the constant lead time for each product’s end result in this stochastic network, the manufacturing lead time or the length of the longest path later. Let \( t_d \) be the due date, \( K_1 \) be the due date assignment cost coefficient, \( K_2 \) be the tardiness cost coefficient, and \( K_3 \) be the earliness penalty cost coefficient.

\[ d = p + t_d \] (13)

where \( p \) is the time epoch at which the product arrives at the system and \( t_d \) is the constant lead time.

In this paper, we attempt to compute the constant lead time by minimising the expected sum of the due-date cost, tardiness cost and earliness cost, which will be explained later. Let \( B \) be the maximum un-penalised lead time, \( T \) be the manufacturing lead time or the length of the longest path in the transformed stochastic network, \( K_1 \) be the due date assignment cost coefficient, \( K_2 \) be the tardiness cost coefficient, and \( K_3 \) be the earliness penalty cost coefficient.

1. Due-date cost: This is a cost that represents the potential loss of sales associated with quoting long delivery dates and is given by:

\[ C_1(t_d) = \begin{cases} 0 & \text{if } t_d \leq B \\ K_1(t_d - B) & \text{if } t_d > B \end{cases}. \] (14)

2. Tardiness cost: This is a cost experienced each time an order is delivered after its due date and is given by:

\[ C_2(t, t_d) = \begin{cases} 0 & \text{if } t \leq t_d \\ K_2(t - t_d) & \text{if } t > t_d \end{cases}. \] (15)

3. Earliness cost: This cost is associated with finished goods inventory carrying costs when products cannot be shipped out at their moment of completion and is given by:

\[ C_3(t, t_d) = \begin{cases} 0 & \text{if } t \geq t_d \\ K_3(t_d - t) & \text{if } t < t_d \end{cases}. \] (16)

The expected cost of assigning a lead time \( t_d \) is then given by:

\[ E[C(t_d)] = K_1(t_d - B)^+ + K_2 E[(T - t_d)^+] + K_3 E[(t_d - T)^+]. \] (17)

where \((a)^+ = \max(a, 0)\). The corresponding algorithm for computing the constant lead time \( t_d^* \) is as follows (see Seidmann and Smith 1981 for details):

**Step 1.** Solve the system of differential Equation (10) according to relation (11) and obtain \( F(t) \).

**Step 2.** Check if \( K_3 = 0 \). If yes, then go to Step 4. Otherwise, go to Step 3.

**Step 3.** Check if \( F(B) \geq (K_2 / (K_2 + K_3)) \). If yes, then set \( t_d^* = \bar{t}_d \), where \( F(\bar{t}_d) = (K_2 / (K_2 + K_3)) \). Otherwise, go to Step 4.

**Step 4.** Check if \( F(B) < (K_2 / (K_2 + K_3)) \). If yes, then set \( t_d^* = \bar{t}_d \), where \( F(\bar{t}_d) = (K_2 / (K_2 + K_3)) \). Otherwise, set \( t_d^* = B \).

Note that in this section we are assuming linear costs. Results for more general cost functions are also available in Seidmann and Smith (1981).

4. Numerical examples

To illustrate the analytical proposed method, we solve two typical small and medium-sized cases with different configurations for due-date assignment in multi-server multi-stage assembly system, which are presented as the networks of queue.

4.1. Case I

The first case, which is depicted in Figure 3, has been taken from Azaron et al. (2006). In Case 1, it is assumed that we have a system with five service stations depicted as a queuing network in Figure 3. We want to find the constant lead time for each particular product’s end result in this

![Figure 3. The queuing network of Case 1.](image-url)
example. The assumptions of this example are:

- The new product orders, including all their operations, arrive at the system according to a Poisson process at the rate $\lambda = 5$ per week.
- $B = 1.5$, and the cost coefficients are: $K_1 = 10$, $K_2 = 35$, $K_3 = 12$.
- The processing times are independent random variables with exponential distributions, while service rates are: $\mu_1 = 3$, $\mu_2 = 1$, $\mu_3 = 4$, $\mu_4 = 2.5$, $\mu_5 = 3$.
- The allocated servers are: $m_1 = 4$, $m_2 = \infty$, $m_3 = 3$, $m_4 = 5$, $m_5 = 4$.
- The transport time between the service stations settled in nodes 1 and 4 is independent exponentially distributed random variables with the parameters $\lambda_{1,4} = 7$.

Now, we substitute all the nodes in Figure 3 with two series of exponential nodes and also consider the transport times by adding the node $t_{14}$ between the nodes 1 and 4 (see Figure 4). All admissible two-partition cuts of network of Figure 3, i.e. the system states, are presented in Table 1. A superscript star is applied to denote a dormant operation and all others are active. Then, we determine the transition rates, depicted in Figure 5, where $\gamma_a = \frac{(m_a\mu_a - \lambda)}{\lambda}$, $\gamma_a' = \frac{m_a\mu_a}{m_a - 1}$ for $a = 1, 3, 4, 5$ and also $\gamma_2 = \mu_2$.

We organise the infinitesimal generator matrix $Q(\mu)$ according to (8). Table 2 presents the infinitesimal generator matrix $Q(\mu)$ (diagonal components are equal to minus sum of the other components at the same row).

After forming the differential equations and using the relation (11), the distribution function for lead time, depicted in Figure 6, is obtained as follows:

$$ F(t) = 5.465e^{-29t} - 18.534e^{-5t} - 0.95e^{-40t} - 2.559e^{-8t} + 0.973e^{-25t} + 1.877e^{-34t} - 2.632e^{-t} - 2.218e^{-28t} + 2.267e^{-10t} - 0.007e^{-21t} + 0.2e^{-13t} + 48.592e^{-4t} + 0.922e^{-6t} + 3.134e^{-7t} - 0.003e^{-37.5t} - 17.063e^{-3.125t} + 1. $$

Following the presented algorithm in Section 3 for assignment due date, since $K_3 \neq 0$ and $F(1.5) = 0.366 < \frac{K_2}{(K_2 + K_1)} = 0.745$, we go to Step 4. Therefore, $t_{14}^* = 1.79$, where $F(1.5) = 0.366 < \frac{(K_2 - K_1)}{(K_2 + K_1)} = 0.532$, namely $F(1.79) = 0.532$.

---

Table 1. All admissible two-partition cuts of the product in Case I.

| 1. (1, 2) | 4. (2, 3) | 7. (2, $t_{14}^*$) | 10. (1, 3) | 13. (1', 3') | 16. (4) | 19. (5') |
| 2. (1', 2) | 5. (1', 3) | 8. (3, 4) | 11. ($t_{14}^*$, 4) | 14. (1', 3') | 17. (4') | 20. (\(\phi, \phi\)) |
| 3. (1, 3) | 6. (1, 3') | 9. (1', 3') | 12. ($t_{14}', 3'$) | 15. ($t_{14}', 3'$') | 18. (5) |

---

Figure 4. The substituted network of Case I.

Figure 5. Rate diagram for the continuous-time Markov chain in Case I.
Table 2. Matrix $Q(\mu)$.

<table>
<thead>
<tr>
<th>State</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\gamma_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>$\gamma_1'$</td>
<td>$\gamma_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_1$</td>
<td>$\gamma_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\lambda_{14}$</td>
<td>$\gamma_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_1'$</td>
<td>$\gamma_3'$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_1$</td>
<td>$\gamma_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_1'$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_3'$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\lambda_{14}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\gamma_1'$</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. $F(t)$ versus $t$ in Case I.

4.2. Case II

We again consider Example 1 as Case II, which is a medium-scale case (taken from Perkgoz et al. 2007). In this case, it is assumed that the new product orders arrive at the system according to a Poisson process at the rate of $\lambda = 4.5$ per week, the cost coefficients are $K_1 = 10, K_2 = 50, K_3 = 0$ and $B = 5$. The processing times are also independent random variables with exponential distributions, while service rates are $\mu_1 = 0.5, \mu_2 = 1, \mu_3 = 1.5, \mu_4 = 10, \mu_5 = 7, \mu_6 = 3, \mu_7 = 2.5, \mu_8 = 4$ and the allocated servers are $m_1 = m_2 = m_3 = \infty, m_4 = m_5 = 1, m_6 = 4, m_7 = 5, m_8 = 3$. The transport time between the service stations settled in nodes 7 and 8 has a generalised Erlang distribution of order 2 with the parameters $(\lambda_{7,8}, \lambda_{7,8}) = (3, 5)$. The transport times between the other service stations are equal to zero.

Similar to Case I, we transform the queuing network of Case II into a substitute stochastic network with exponentially distributed arc lengths. The corresponding stochastic process $\{X(t), t \geq 0\}$ has 28 states. After forming the differential equations, the distribution function for lead time, depicted in Figure 7, is also obtained as follows:

$$F(t) = 3264e^{-3t} - 5.4753e^{-0.5t} + 6.2064e^{-4t} + 29.2478e^{-5t} + 16.1415e^{-1.5t} + 3.8784e^{-6t} - 1.0413 \times 10^{14}e^{-8t} + 1.9210 \times 10^{-13}e^{-4.5t} + 5.8933 \times 10^{-9}e^{-3.0t} - 3.9580 \times 10^{3}e^{-3.125t}$$
Following the presented algorithm in Section 3 for assignment due date, since $K_3 = 0$, we go to Step 4. Therefore, $t_d^* = 6.533$, where $F(5) = 0.5974 < \frac{(K_2 - K_1)}{(K_2 + K_3)} = 0.8$, namely $F(6.533) = 0.8007$.

5. Conclusion
In this article, we modelled a multi-server multi-stage assembly system as a queuing network, where the new product orders, including all their operations, are entered into the system according to a Poisson process. It was assumed that the multi-servers can be settled in each service station and each operation of any work is operated at a devoted service station with only one of the servers located at a node of the network, while the transport times between each pair of service stations are independent random variables with generalised Erlang distributions. Moreover, each product’s end result has a penalty cost that is some linear function of its due-date and its actual lead time.

For computing the due date for multi-server multi-stage assembly system, first the queuing network was converted into a stochastic network with exponentially distributed arc lengths and then by constructing a proper finite-state continuous-time Markov model, a system of differential equations was created to solve and find the manufacturing lead time distribution for any particular product, analytically. Note that the number of system states grows combinatorially with the number of UDCs. Next, the constant due date for delivery time was obtained by using a linear function of its due date and minimising the expected aggregate cost per product. Finally, to illustrate the analytical proposed method, we solved two numerical examples for due date assignment in multi-server multi-stage assembly systems.

Notes on contributor
Saeed Yaghoubi is an assistant professor of industrial engineering at Iran University of Science and Technology, Tehran, Iran. He obtained his PhD degree in 2012 from the Iran University of Science and Technology in industrial engineering. His current research interests include stochastic processes and their applications, supply chains and project scheduling. He has contributed articles to different international journals such as *European Journal of Operational Research, Iranian Journal of Science & Technology, International Journal of Advanced Manufacturing Technology, Journal of Industrial Engineering International*, etc.

References


