Stochastic Cyclic Scheduling Problem in Synchronous Assembly and Production Lines

Selçuk Karabatı and Barış Tan

Koç University
College of Administrative Sciences and Economics
Çayır Cad. No. 5, İstinye, İstanbul, 80860, Turkey

September 1997; Revised March 1998

Abstract

In this paper we address the stochastic cyclic scheduling problem in synchronous assembly and production lines. Synchronous lines are widely used in the production and assembly of various goods such as automobiles and household appliances. We consider cycle time minimization (or throughput rate maximization) as the objective of the scheduling problem with the assumption that the processing times are independent random variables. We first discuss the two-station case and present a lower bounding scheme and an approximate solution procedure for the scheduling problem. For the general case of the problem, two heuristic solution procedures are presented. An extension of the two-station lower bound to the general case of the problem is also discussed. The performance of the proposed heuristics on randomly generated problems is documented, and the impact of scheduling decisions in problems with different levels of variability in processing times is analyzed. We also analyze the problem of sequence determination when the available information is limited to the expected values of individual processing times.

Keywords: Cyclic Scheduling, Stochastic Scheduling, Synchronous Flow Lines

Introduction

Synchronous lines are designed mostly to meet the assembly or production requirements of a single product, which is usually in the maturity stage of its life cycle, and efficiency is always considered as the primary objective in the design process. With the single-product focus, efficiency can be achieved by appropriately assigning operations of the product to stations of the line. Changes experienced in the nature of competition in various markets have decreased the relative importance of the efficiency criterion, and increased the weights of criteria such as product range flexibility, volume flexibility, and delivery speed. As a result of these changes, the line concept has shifted its focus from a single product to a mix of products that are in different stages of their life cycles, and therefore simultaneous production of a mix of products

has become an important issue in the management of assembly and production lines. The shift to mixed-model production has also changed the practice, and operational decisions, such as scheduling, have gained more importance. For example, in the mixed-model setting, the performance of the line now also depends on the scheduling decisions.

One of the recent approaches to the management of mixed-model environments is to employ a cyclic scheduling policy at the operational level. Cyclic scheduling policies have proved to be very effective in repetitive manufacturing environments within the JIT context (Schonberger¹). In this approach the minimal product set (*i.e.*, the smallest product set having the same product mix ratio with the forecasted demand for various items), or a multiple of it, is repetitively produced. Therefore, for an effective implementation of the cyclic scheduling approach, we require that the assembly line can be operated with zero (or negligible) setup times.

The synchronous transfer of products can be observed in different production environments. The most common example is the final assembly operation of automobile manufacturers. Fuxman² provides detailed information on an American automobile manufacturer's final assembly operation where product, *i.e.*, *automobile*, transfers are synchronously performed. In a synchronous assembly line, the product transfers are coordinated, *i.e.*, they are simultaneously performed. The duration of each cycle is determined by the operation with the longest processing time among the operations performed in that cycle, and upon the completion of this longest operation each product moves to the next station.

The deterministic cyclic scheduling problem in *asynchronous* flow lines, where product transfers are not coordinated and processing times are deterministic, has been extensively studied in the literature (McCormick *et al.*³, Karabati and Kouvelis⁴, Abadi *et al.*⁵). Hall and Sriskandarajah⁶ present a review of scheduling problems in blocking and no-wait flow shops. The cyclic scheduling problem in a *synchronous* assembly line has been first discussed in Fuxman². Fuxman² presents a model to study the effects of lot sizing decisions on the throughput rate of the assembly line. Kouvelis and Karabati⁷ present exact and approximate solution procedures for the deterministic cyclic scheduling problem in synchronous flow lines.

In this paper we study the cyclic scheduling problem in synchronous assembly/production lines with stochastic processing times. One of the most commonly used measures to evaluate the performance of a production line is the throughput rate. The throughput rate is defined as the expected number of batches assembled or manufactured per unit time in the long run. In this study, the cycle time of a batch of products is defined as the reciprocal of the throughput rate. The objective of the models and solution procedures discussed in this study is to evaluate performances of given sequences, and to find sequences which maximize the throughput rate

of the line. The models which are simultaneously assembled or manufactured on the line are released to the line in identical batches. Therefore composition of each batch reflects the production requirements of different models over the planning horizon. For example, in an automobile assembly environment, if the production requirements of models A, B, and C are respectively 2000, 3000, and 5000 units, a batch of 10 automobiles, which contains 2 units of Model A, 3 units of Model B, and 5 units of Model C, can be released to the line in a repetitive manner to fulfill the production requirements. The objective of the sequencing problem is then to find a release sequence of these 10 automobiles which minimizes the average time to produce the batch of 10 automobiles, or maximizes the average number of batches produced per unit time in the long run.

In an attempt to better represent the problem environment, the models presented in this paper assume that the processing times are subject to variability. In an assembly line environment randomness in processing times can be attributed to factors such as inherent variability in manual assembly operations, non-uniformities which may be present in the assembled parts, variations in the speed of tools used in the assembly operations, unreliable stations etc. From a managerial point of view, inclusion of variability in the model is desired to the extent that the resulting model is tractable. An extensive discussion of scheduling problems with stochastic processing times is presented in Pinedo⁸. Rao and Jackson⁹ address the stochastic cyclic scheduling problem in an asynchronous job shop, and present an approach to estimate the two-moment behavior of system throughput and job flow time performance measures for a given sequence of operations. Bowman and Muckstadt¹⁰ model the same problem as a continuous time Markov chain, and present an approach to determine the expected value and variance of the cycle time performance measure.

This paper is organized as follows. We first present a general framework for the stochastic cyclic scheduling problem in synchronous assembly and production lines. We then analyze the 2-station problem, and develop a lower bounding scheme and an approximate solution procedure for the problem. Subsequently, we address the general case of the problem, present two approximate solution procedures. We then present numerical examples and report our computational experience with these procedures. We also discuss certain properties of the sequencing problem when the available information is limited to the expected values of individual processing times. We conclude the paper with a summary of our results and a discussion of directions for future research.

The Model

Consider a serial assembly line with m stations $ST_1, ST_2, \dots, ST_m$, where the product (or, interchangeably, job) transfers between stations are performed synchronously. Unlike the asynchronous transfer mode, the synchronous transfer mode requires coordination of all job transfers. In the asynchronous transfer mode, jobs can flow downstream in the line upon completion of their operations if their downstream stations, or buffer locations, are idle. In the synchronous transfer mode, on the other hand, job transfers are performed simultaneously, therefore jobs can be transferred to their downstream stations at certain fixed points in time. These transfer instances are determined by the longest processing time of operations that are being simultaneously processed. The n jobs, $J_1, J_2, \dots, J_n$, are scheduled cyclically, *i.e.*, the batch of n jobs is repetitively produced using the same order of jobs in every batch. We also note that some of these jobs may be the same product type. The processing time of job J_i on station ST_j is given by independent random variable $P_{i,j}, i = 1, 2, \dots, n; j = 1, 2, \dots, m$. We also let $\mu_{i,j} = E[P_{i,j}], i = 1, 2, \dots, n; j = 1, 2, \dots, m$, *i.e.*, $\mu_{i,j}$ is the expected value of job J_i 's processing time on station ST_j . The objective is to find a sequence of jobs that optimizes the performance of the line with respect to a certain criterion, such as minimizing the expected value of batch cycle time, or variance of batch cycle time. We will present a more detailed discussion of the performance criteria in the subsequent sections.

Let $S = (J_{[1]}, J_{[2]}, \dots, J_{[n]})$ denote a cyclic sequence of jobs. The jobs are released to the assembly line according to the $J_{[1]}J_{[2]} \dots J_{[n]}J_{[1]}J_{[2]} \dots J_{[n]} \dots J_{[1]}J_{[2]} \dots J_{[n]} \dots$ sequence in a repetitive manner, where $[i]$ is the index of job that is the in the i -th position in schedule S . The cycle time performance criterion reflects the performance of the assembly line when there are no idle stations in the line. In the synchronous cyclic scheduling problem, every station becomes occupied by one job when the job that has been released to the system in the first position reaches the last station. For sequence S we have the following processing times to determine the subcycles times when all stations are occupied:

Subcycle (i)	Stations					Subcycle Time $C_i(S)$
	1	2	3	...	m	
1	$P_{[1],1}$	$P_{[n],2}$	$P_{[n-1],3}$	...	$P_{[(1+1-m) \bmod n],m}$	$\max_{1 \leq j \leq m} P_{[(1+1-j) \bmod n],j}$
2	$P_{[2],1}$	$P_{[1],2}$	$P_{[n],3}$	...	$P_{[(2+1-m) \bmod n],m}$	$\max_{1 \leq j \leq m} P_{[(2+1-j) \bmod n],j}$
3	$P_{[3],1}$	$P_{[2],2}$	$P_{[1],3}$	...	$P_{[(3+1-m) \bmod n],m}$	$\max_{1 \leq j \leq m} P_{[(3+1-j) \bmod n],j}$
...	...	...	...	...	...	...
$n-1$	$P_{[n-1],1}$	$P_{[n-2],2}$	$P_{[n-3],3}$	...	$P_{[((n-1)+1-m) \bmod n],m}$	$\max_{1 \leq j \leq m} P_{[((n-1)+1-j) \bmod n],j}$
n	$P_{[n],1}$	$P_{[n-1],2}$	$P_{[n-2],3}$	...	$P_{[(n+1-m) \bmod n],m}$	$\max_{1 \leq j \leq m} P_{[(n+1-j) \bmod n],j}$
						Total Cycle Time = $\sum_{i=1}^n C_i(S)$

Due to the cyclic nature of the problem, the convention $0 \bmod n = n$ is used in determining the

processing times of schedule S . In the above example, in subcycle 1 we process the first job of sequence S in the first station ($P_{[1],1}$), the last job of S in the second station ($P_{[n],2}$) and so on. Note that these two jobs belong to different batches. As product batches are released to the line, one batch of jobs is completed in n subcycles. At the end of each subcycle one completed product comes off the line. Let $C_i(S), i = 1, 2, \dots, n$, be the random variable that denotes the duration of i -th subcycle, *i.e.*, the subcycle in which the first operation of job $J_{[i]}$ in schedule S is processed. $C_i(S)$ is given by

$$C_i(S) = \max_{1 \leq j \leq m} P_{[(i+1-j) \bmod n], j}.$$

The batch cycle time of schedule S , $CT(S)$, is then equal to

$$CT(S) = \sum_{i=1}^n C_i(S). \quad (1)$$

Therefore,

$$E[CT(S)] = E\left[\sum_{i=1}^n C_i(S)\right] = \sum_{i=1}^n E[C_i(S)].$$

Let Π be the set of permutation sequences of n jobs. The expected batch cycle time minimization problem can be formally presented as follows:

$$\min_{S \in \Pi} \sum_{i=1}^n E[C_i(S)].$$

The random variables $C_i(S), i = 1, 2, \dots, n$, are statistically independent, because each processing time appears only in one subcycle, and processing times are independent random variables. Since $C_i(S), i = 1, 2, \dots, n$, are independent random variables,

$$V(CT(S)) = V\left(\sum_{i=1}^n C_i(S)\right) = \sum_{i=1}^n V(C_i(S)) = \sum_{i=1}^n (E[C_i^2(S)] - E[C_i(S)]^2).$$

Let $\rho_{i,j}(\cdot)$ ($\theta_{i,j}(\cdot)$) be the probability density (cumulative distribution) function of random variable $P_{i,j}$. Let $F_i(t)$ be the cumulative distribution function of random variable $C_i(S)$. $F_i(t)$ is given by

$$\begin{aligned} F_i(t) = \Pr\{C_i(S) \leq t\} &= \prod_{j=1}^m \theta_{[(i+1-j) \bmod n], j}(t) \\ &= \prod_{j=1}^m \int_0^t \rho_{[(i+1-j) \bmod n], j}(z) dz. \end{aligned}$$

In the subsequent computational analysis, processing times are assumed to follow a Weibull distribution with mean $E[P_{i,j}] = \mu_{i,j}$ and shape parameter α . It is also assumed that the shape

parameter α is common for all processing time distributions. Note that a variety of distributions can be approximated by using a Weibull distribution. Therefore, real data collected from a process can be used to fit a two-parameter Weibull distribution. For a discussion of maximum-likelihood estimators of the Weibull distribution the reader is referred to Law and Kelton¹¹, Chapter 5, and the references therein. Assuming a common shape parameter implies that the shape of the density functions are similar for the processing time distributions. The squared coefficient of variation, *s.c.v.*, of a random variable with a Weibull distribution with shape parameter α is

$$s.c.v. = \frac{2\alpha\Gamma(\frac{2}{\alpha})}{[\Gamma(\frac{1}{\alpha})]^2} - 1,$$

where $\Gamma(\cdot)$ is the gamma function defined by $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ for any real number $z > 0$ (see Law and Kelton¹¹ for a detailed discussion of the Weibull distribution). Note that, if α is equal to 1, then *s.c.v.* is equal to 1 which corresponds to zero memory (Weibull distribution with $\alpha = 1$ and exponential distribution are identical). If α is less than 1 then *s.c.v.* is greater than 1 which implies negative memory. Similarly if α is greater than 1 then *s.c.v.* is less than 1, thus the processing time distribution has positive memory. In the limiting cases, as α approaches infinity *s.c.v.* approaches 0, which is basically the deterministic case, and as α approaches 0, *s.c.v.* approaches infinity, which is the pure random case. In a manufacturing system with reliable stations, the expected remaining time after some time has passed is less than the expected service completion time, *i.e.*, we expect to observe the positive memory property. Since Weibull distribution allows us to evaluate systems with both positive and negative memory, unreliable manufacturing systems can be analyzed in the same manner.

When the processing times are Weibull random variables, the expected value of subcycle i ($C_i(S)$) is given by

$$\begin{aligned} E[C_i(S)] &= E \left[\max_{1 \leq j \leq m} \{P_{[(i+1-j \bmod n), j]}\} \right] \\ &= \sum_{j=1}^m \mu_{[(i+1-j \bmod n), j]} - \sum_{j=1}^{m-1} \sum_{k=j+1}^m \left(\mu_{[(i+1-j \bmod n), j]}^{-\alpha} + \mu_{[(i+1-k \bmod n), k]}^{-\alpha} \right)^{-\frac{1}{\alpha}} \\ &\quad + \sum_{j=1}^{m-2} \sum_{k=j+1}^{m-1} \sum_{l=1}^m \left(\mu_{[(i+1-j \bmod n), j]}^{-\alpha} + \mu_{[(i+1-k \bmod n), k]}^{-\alpha} + \mu_{[(i+1-l \bmod n), l]}^{-\alpha} \right)^{-\frac{1}{\alpha}} \\ &\quad + \dots \\ &\quad + (-1)^m \left(\mu_{[(i+1-1 \bmod n), 1]}^{-\alpha} + \mu_{[(i+1-2 \bmod n), 2]}^{-\alpha} + \dots + \mu_{[(i+1-m \bmod n), m]}^{-\alpha} \right)^{-\frac{1}{\alpha}}. \quad (2) \end{aligned}$$

Similarly, the expected value of $(C_i(S))^2$ is given by:

$$\begin{aligned}
E[(C_i(S))^2] &= \frac{2\alpha\Gamma(\frac{2}{\alpha})}{[\Gamma(\frac{1}{\alpha})]^2} \left(\sum_{j=1}^m \mu_{[(i+1-j \bmod n),j]}^2 - \sum_{j=1}^{m-1} \sum_{k=j+1}^m \left(\mu_{[(i+1-j \bmod n),j]}^{-\alpha} + \mu_{[(i+1-k \bmod n),k]}^{-\alpha} \right)^{-\frac{2}{\alpha}} \right. \\
&\quad + \sum_{j=1}^{m-2} \sum_{k=j+1}^{m-1} \sum_{l=1}^m \left(\mu_{[(i+1-j \bmod n),j]}^{-\alpha} + \mu_{[(i+1-k \bmod n),k]}^{-\alpha} + \mu_{[(i+1-l \bmod n),l]}^{-\alpha} \right)^{-\frac{2}{\alpha}} \\
&\quad + \dots \\
&\quad \left. + (-1)^m \left(\mu_{[(i+1-1 \bmod n),1]}^{-\alpha} + \mu_{[(i+1-2 \bmod n),2]}^{-\alpha} + \dots + \mu_{[(i+1-m \bmod n),m]}^{-\alpha} \right)^{-\frac{2}{\alpha}} \right). \quad (3)
\end{aligned}$$

Proofs of Equations (??) and (??) are given in the Appendix.

The Two-Station Scheduling Problem

In this section the 2-station case of the cyclic scheduling problem in synchronous assembly lines is discussed. We will first demonstrate that the problem can be formulated as a Traveling Salesman Problem (TSP), and then present a lower bounding scheme and an approximation procedure based on certain properties of the TSP formulation.

Suppose job J_i is the immediate predecessor of job J_j in a given sequence S , *i.e.*, jobs J_i and J_j are two contiguous jobs in sequence S , and job J_i is released to the line before job J_j . With this sequencing assumption, the first operation of job J_j is processed in the same subcycle with the second operation of job J_i . Let $c_{i,j}$ be the expected cycle time of this subcycle. $c_{i,j}$ is then given by $c_{i,j} = E[\max\{P_{j,1}, P_{i,2}\}]$. The two-station expected batch cycle time minimization problem can be modeled as a TSP with n cities and distance matrix $C = [c_{i,j}]$. Without any loss of generality, we will assume that

$$\mu_{1,2} \leq \mu_{2,2} \leq \dots \leq \mu_{n,2}.$$

Also let ϕ be a permutation of the n jobs with the following property:

$$\mu_{\phi(1),1} \leq \mu_{\phi(2),1} \leq \dots \leq \mu_{\phi(n),1}.$$

We now present a definition of a special distance matrix for TSP.

Definition 1 (Gilmore *et al.*¹²) *Let $A = [a_{i,j}]$ be an $n \times n$ matrix. A is a permuted distribution matrix if there exists a permutation δ such that*

$$a_{i,\delta(j)} + a_{i+1,\delta(j-1)} - a_{i,\delta(j-1)} - a_{i+1,\delta(j)} \geq 0, \quad 1 \leq i \leq n-1; 2 \leq j \leq n.$$

Theorem 1 (Gilmore *et al.*¹²) *If A is a permuted distribution matrix for permutation δ , then δ is an optimal assignment for A .*

Theorem 2 C is a distribution matrix for permutation ϕ , when $\mu_{i,j} \leq \mu_{k,l}$ implies $\theta_{i,j}(t) \geq \theta_{k,l}(t)$, $\forall t$, i.e., when ordering in expectation implies stochastic ordering.

Proof: A proof of Theorem 2 is presented in the Appendix.

Theorem 2 implies that permutation ϕ is an optimal assignment for the distance matrix C and $\sum_{i=1}^n c_{i,\phi(i)}$ constitutes a lower bound on the minimum expected batch cycle time value. In the TSP context, $\phi(i), i = 1, 2, \dots, n$, is the city that follows city i in the tour. An example illustrating the application of Theorem 2 is provided in the numerical examples section.

Certain properties of permuted distribution matrices are discussed in Gilmore *et al.*¹² These properties can be used to develop efficient solution procedures for some problem classes. The Gilmore-Gomory¹³ TSP is one such example. In the Gilmore-Gomory¹³ TSP the cost matrix is defined as follows:

$$c_{i,j} = \begin{cases} \int_{b_i}^{a_j} f(x)dx & \text{if } b_i \leq a_j, \\ \int_{a_j}^{b_i} g(x)dx & \text{if } b_i \geq a_j, \end{cases}$$

where $f(x) + g(x) \geq 0$. Gilmore and Gomory¹³ have shown that TSP with the above distance matrix can be solved in polynomial time. Matsuo¹⁴ has shown that 2-station *deterministic asynchronous* cyclic scheduling problem with no buffers between the two stations can be formulated as a Gilmore-Gomory¹³ TSP with $f(x) = 0$ and $g(x) = 1$ and where $a_i(b_i), i = 1, 2, \dots, n$, is the processing time of job J_i on the first(second) station. A detailed presentation of the Gilmore-Gomory algorithm, and an example of its application are given in Matsuo¹⁴. In the 2-station and no buffers case, asynchronous and synchronous transfer modes are equivalent for the cycle time criterion, therefore the *deterministic synchronous* cyclic scheduling can be solved in polynomial time using the Gilmore-Gomory¹³ solution procedure. When we map the above cost structure to our problem we get $a_i = \mu_{i,1}$ and $b_i = \mu_{i,2}$. With this mapping, and with functions $f(x) = 0$ and $g(x) = 1$, the Gilmore-Gomory¹³ solution procedure is employed as an approximate solution procedure for the problem, and problem instances where the procedure is not able to generate an optimal solution are observed. The computational experience with the lower bounding scheme and the approximate solution procedure will be presented in the computational results section.

A special case of the 2-station problem is the situation where random variables $P_{i,1}$ and $P_{i,2}$ are identical, i.e., the processing times of job J_i follow the same distribution (possibly with different realizations) on stations ST_1 and ST_2 . In this case, with the $\mu_{1,2} \leq \mu_{2,2} \leq \dots \leq \mu_{n,2}$, indexing of operations, we have $\phi(i) = i, i = 1, 2, \dots, n$. It can now be shown that matrix C of Theorem 2 is a *distribution matrix* (see Definition 1) and for distribution matrices an optimal

solution to TSP can be found in less than $O(n^2)$ time (see Gilmore *et al.*¹² for a discussion of this solution procedure). Therefore, it can be concluded that this special case of the 2-station problem can be solved in polynomial time.

The m -Station Scheduling Problem

In this section an analysis of the cyclic scheduling problem in a synchronous line with m stations is presented. We first discuss a property of the problem which can be used to reduce the computational complexity of the solution procedure when m is larger than n . Then a lower bounding procedure, and two approximate solution procedures for the expected batch cycle time minimization problem are presented.

In a problem with n jobs there are n subcycles, and therefore, when $m > n$, some of job J_i 's operations are processed in the same subcycles. Consider subcycle i , whose cycle time is equal to $\max_{1 \leq j \leq m} P_{[i+1-j \bmod n],j}$. Let $l = (i + 1 - j) \bmod n$, for a particular value of j smaller than or equal to n , *i.e.*, $j \leq n$. We then have $l = (i + 1 - (j + kn)) \bmod n$, $k = 1, 2, \dots, \lfloor \frac{m-j}{n} \rfloor$. Therefore, job $J_{[l]}$ appears in the i -th subcycle for a total of $(\lfloor \frac{m-j}{n} \rfloor + 1)$ times. Note the above observation holds true for any value of i . Therefore, for a given $n \times m$ problem with $m > n$, an equivalent $n \times n$ problem can be defined with the following random variables:

$$\bar{P}_{i,j} = \max_{k=0,1,\dots,\lfloor \frac{m-j}{n} \rfloor} \{P_{i,j+kn}\}, \quad i, j = 1, 2, \dots, n.$$

With this transformation, we can reduce any $n \times m$ problem with $m > n$, to an equivalent $n \times n$ problem.

The deterministic version of the problem addressed in this paper has been shown to be a combinatorially difficult problem in Kouvelis and Karabati⁷. Since the stochastic version of the problem is a generalization of the deterministic problem, it can be safely concluded that the expected batch cycle time minimization problem is a hard problem. Therefore for its solution we propose the use of approximate solution procedures.

Before proceeding with the discussion of approximate solution procedures, we present an extension of the 2-station lower bounding procedure to the m -station case. Let ST_k and ST_{k+1} two adjacent stations of the line. A TSP for the 2-station problem formed by station ST_k and ST_{k+1} can be defined with the following distance matrix:

$$c_{i,j}^k = E[\max \{P_{j,k}, P_{i,k+1}\}], \quad i, j = 1, 2, \dots, n.$$

The solution of the assignment problem with matrix $C^k = [c_{i,j}^k]$ provides a lower bound for the

expected batch cycle time value. Consider the following problem for a given scalar $\lambda, 0 \leq \lambda \leq 1$:

$$(LB) \quad f(\lambda) = \min \sum_{i=1}^n \sum_{j=1}^n (\lambda c_{i,j}^k + (1-\lambda)c_{i,j}^l) x_{i,j}$$

$$\sum_{i=1}^n x_{i,j} = 1, \quad j = 1, 2, \dots, n, \quad (4)$$

$$\sum_{j=1}^n x_{i,j} = 1, \quad i = 1, 2, \dots, n, \quad (5)$$

$$x_{i,j} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n. \quad (6)$$

In (LB) , C^k is defined for stations ST_k and ST_{k+1} , and C^l is defined for stations ST_l and ST_{l+1} , where $k \neq l$. (LB) is a linear assignment problem with the cost matrix $\lambda C^k + (1-\lambda)C^l$, therefore its solution constitutes a lower bound on the optimal objective function value. The lower bound is motivated by the following observation: Let jobs J_i, J_k , and J_l be three jobs simultaneously processed on stations ST_j, ST_{j+1} , and ST_{j+2} , respectively, in an m -station problem. A lower bound on the expected subcycle time is given by $E\{\max\{P_{i,j}, P_{k,j+1}, P_{l,j+2}\}\}$. It can be readily shown that for $\lambda, 0 \leq \lambda \leq 1$,

$$\begin{aligned} E[\max\{P_{i,j}, P_{k,j+1}, P_{l,j+2}\}] &\geq E[\lambda \max\{P_{i,j}, P_{k,j+1}\} + (1-\lambda) \max\{P_{k,j+1}, P_{l,j+2}\}] \\ &= \lambda c_{k,i}^j + (1-\lambda)c_{l,k}^{j+1}, \end{aligned}$$

therefore $\lambda c_{k,i}^j + (1-\lambda)c_{l,k}^{j+1}$ constitutes a lower bound on the expected value of the corresponding subcycle time. A tighter lower bound can be obtained by finding λ^* , where $f(\lambda^*) = \max_{0 \leq \lambda \leq 1} f(\lambda)$. Since $f(\cdot)$ is piecewise linear and concave in the $[0, 1]$ interval, λ^* can be easily determined by performing a simple line search. We now formally state our lower bounding procedure:

Step 0 Let $k = 2$, $M' = C^1$ and $M'' = C^2$.

Step 1 Find $f(\lambda_k^*) = \max_{0 \leq \lambda \leq 1} \min_{\text{s.t. (??), (??), and (??)}} \sum_{i=1}^n \sum_{j=1}^n (\lambda m'_{i,j} + (1-\lambda)m''_{i,j}) x_{i,j}$.

Step 2 If $k \leq m - 2$ then let $k = k + 1$, $M' = \lambda^* M' + (1 - \lambda^*) M''$, and $M'' = C^k$, and go to Step 1. Otherwise, stop; $f(\lambda_k^*)$ is a lower bound on the minimum expected batch cycle time value.

An example illustrating Step 1 of the lower bounding procedure is provided in the numerical examples section.

Approximate Solution Procedure 1

The first approximate solution procedure is a by-product of the lower bounding procedure. In Step 1 of the lower bounding procedure, as linear assignment problems are solved to find λ^* , assignments which can be transformed to feasible solutions for the sequencing problem are generated. Upper bounds on the optimal value of the objective function (*i.e.*, approximate solutions) can now be obtained by evaluating these sequences for their expected batch cycle time values. An example for this upper bounding procedure is presented in the numerical examples section.

Approximate Solution Procedure 2

The second approximate solution procedure uses a set of expected subcycle time values as input, and tries to find a sequence of jobs that has the highest probability of not violating these given expected subcycle time values. Let $t_i, i = 1, 2, \dots, n$, be a given set of expected subcycle time values. For each job $J_i, i = 1, 2, \dots, n$, and position $j, j = 1, 2, \dots, n$, in the sequence, we define the parameter $a_{i,j}$, which reflects the “cost” of assigning job J_i to position j in the sequence. We can define $a_{i,j}$ in a number of ways:

$$a_{i,j} = \sum_{k=1}^n \Pr \{P_{i,k} \geq t_{j+k-1 \bmod n}\}, \quad (7)$$

or

$$a_{i,j} = \max_{1 \leq k \leq n} \Pr \{P_{i,k} \geq t_{j+k-1 \bmod n}\}. \quad (8)$$

$a_{i,j}$ is defined as the sum or maximum of job J_i 's probability of violating expected subcycle time values (*i.e.*, t_i 's) when it is assigned to position j . The solution of the assignment problem with the cost matrix $A = [a_{i,j}]$ generates a sequence, and we evaluate this sequence for its expected batch cycle time value. We can now define an iterative scheme as follows: In the first step, for a given set of expected subcycle time values, a sequence is obtained by solving the above described assignment problem. Then, in the second step the expected subcycle time values of this new sequence are determined, and the first step is repeated with these new expected subcycle time values. As a stopping criterion, a limit can be imposed on the number of iterations that will be performed. The approximate solution can be formally stated as follows:

Step 0 Choose an initial sequence S_0 and determine its cycle times, $C_i(S_0), i = 1, 2, \dots, n$.
Let $l = 0$.

Step 1 For $i, j = 1, 2, \dots, n$ determine

$$a_{i,j} = \sum_{k=1}^n \Pr \{P_{i,k} \geq C_{j+k-1 \bmod n}(S_l)\},$$

or

$$a_{i,j} = \max_{1 \leq k \leq n} \Pr \{P_{i,k} \geq C_{j+k-1 \bmod n}(S_l)\}.$$

Step 2 Solve the problem $\left(\min \sum_{i=1}^n \sum_{j=1}^n a_{i,j} x_{i,j} \right)$ subject to (??), (??), and (??), and let $X^l = [x_{i,j}^l]$ be an optimal solution.

Step 3 Define sequence $S_{l+1} = (J_{[1]}^l, J_{[2]}^l, \dots, J_{[n]}^l)$ as follows: $J_{[j]}^l = \sum_{i=1}^n i x_{i,j}, j = 1, 2, \dots, n$. Determine $C_i(S_{l+1}), i = 1, 2, \dots, n$, and $CT(S_{l+1})$, and update the current best solution if necessary.

Step 4 Let $l = l + 1$. If $l < L$, where L is the maximum number of iterations allowed, go to Step 1, otherwise stop and report the current best solution.

As the initial sequence S_0 in Step 1 of the procedure, we use the schedule generated by the first approximate solution procedure in the root node of the search tree. An example illustrating Steps 1, 2, and 3 of the approximate solution procedure is provided in the next section.

Numerical Examples

First consider a 4-job 2-station problem with Weibull processing times where $s.c.v. = 0.2$, or equivalently $\alpha = 2.379$. The expected processing times are as follows:

Job	Stations	
	1	2
1	6	3
2	5	4
3	2	9
4	7	9

Let $c_{i,j}$ be the expected duration of a cycle where second operation of job J_i and first operation of job J_j are processed simultaneously. For example, the total expected cycle time of sequence $S = (1, 2, 3, 4)$ is equal to $c_{1,2} + c_{2,3} + c_{3,4} + c_{4,1}$. $c_{1,2}$ is computed as follows:

$$c_{1,2} = \mu_{2,1} + \mu_{1,2} - \frac{1}{\left(\frac{1}{\mu_{2,1}^\alpha} + \frac{1}{\mu_{1,2}^\alpha}\right)^{\frac{1}{\alpha}}} = 5 + 3 - \frac{1}{\left(\frac{1}{5^{2.379}} + \frac{1}{3^{2.379}}\right)^{\frac{1}{2.379}}} = 8 - 2.690 = 5.310.$$

The complete $C = [c_{i,j}, i, j = 1, 2, \dots, n]$ matrix is given as follows:

$$C = \begin{bmatrix} 6.213 & 5.310 & 3.253 & 7.153 \\ 6.507 & 5.706 & 4.142 & 7.375 \\ 9.761 & 9.442 & 9.023 & 10.177 \\ 9.761 & 9.442 & 9.023 & 10.177 \end{bmatrix}.$$

Since, $\mu_{1,2} \leq \mu_{2,2} \leq \mu_{3,2} \leq \mu_{4,2}$, C is a distribution matrix for $\phi = (3, 2, 1, 4)$ according to Theorem 2. Therefore, $\sum_{i=1}^4 c_{i,\phi(i)} = c_{1,3} + c_{2,2} + c_{3,1} + c_{4,4} = 3.253 + 5.706 + 9.761 + 10.177 = 28.897$ is a lower bound on the optimal objective function value. This solution does not constitute a feasible solution for the sequencing problem, because it contains 3 subtours: $1 \rightarrow 3 \rightarrow 1$, $2 \rightarrow 2$, and $4 \rightarrow 4$. A feasible solution for the sequencing problem can be obtained by using the Gilmore-Gomory algorithm with the expected processing times. The Gilmore-Gomory algorithm with $f(x) = 0$ and $g(x) = 1$ results in tour $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1$. This tour corresponds to sequence $S = (1, 3, 4, 2)$ with an expected batch cycle time value of 29.379.

Let us now consider a 4-job 3-station problem with Weibull processing times where $s.c.v. = 0.2$, or equivalently $\alpha = 2.379$. The expected processing times are as follows:

Job	Stations		
	1	2	3
1	6	3	8
2	5	4	8
3	2	9	2
4	7	9	3

For schedule $S = (1, 4, 3, 2)$ the cycle times are computed as follows:

Subcycle (i)	Stations			Subcycle Time $C_i(S)$
	1	2	3	
1	$P_{[1],1}$	$P_{[4],2}$	$P_{[3],3}$	$\max\{P_{1,1}, P_{2,2}, P_{3,3}\} = 6.520$
2	$P_{[2],1}$	$P_{[1],2}$	$P_{[4],3}$	$\max\{P_{4,1}, P_{1,2}, P_{2,3}\} = 9.454$
3	$P_{[3],1}$	$P_{[2],2}$	$P_{[1],3}$	$\max\{P_{3,1}, P_{4,2}, P_{1,3}\} = 10.685$
4	$P_{[4],1}$	$P_{[3],2}$	$P_{[2],3}$	$\max\{P_{2,1}, P_{3,2}, P_{4,3}\} = 9.469$

$C_1(S)$ is given by

$$\begin{aligned} C_1(S) &= \max\{P_{1,1}, P_{2,2}, P_{3,3}\} \\ &= \mu_{1,1} + \mu_{2,2} + \mu_{3,3} - \frac{1}{\left(\frac{1}{\mu_{1,1}^\alpha} + \frac{1}{\mu_{2,2}^\alpha}\right)^{\frac{1}{\alpha}}} - \frac{1}{\left(\frac{1}{\mu_{1,1}^\alpha} + \frac{1}{\mu_{3,3}^\alpha}\right)^{\frac{1}{\alpha}}} - \frac{1}{\left(\frac{1}{\mu_{2,2}^\alpha} + \frac{1}{\mu_{3,3}^\alpha}\right)^{\frac{1}{\alpha}}} \\ &\quad + \frac{1}{\left(\frac{1}{\mu_{1,1}^\alpha} + \frac{1}{\mu_{2,2}^\alpha} + \frac{1}{\mu_{3,3}^\alpha}\right)^{\frac{1}{\alpha}}} \\ &= 6 + 4 + 2 - \frac{1}{\left(\frac{1}{6^{2.379}} + \frac{1}{4^{2.379}}\right)^{\frac{1}{2.379}}} - \frac{1}{\left(\frac{1}{6^{2.379}} + \frac{1}{2^{2.379}}\right)^{\frac{1}{2.379}}} - \frac{1}{\left(\frac{1}{4^{2.379}} + \frac{1}{2^{2.379}}\right)^{\frac{1}{2.379}}} \\ &\quad + \frac{1}{\left(\frac{1}{6^{2.379}} + \frac{1}{4^{2.379}} + \frac{1}{2^{2.379}}\right)^{\frac{1}{2.379}}} \\ &= 6 + 4 + 2 - 3.492 - 1.941 - 1.858 + 1.811 = 6.520. \end{aligned}$$

The lower bound presented for the 2-station example is also valid for this 3-station problem, because the two problems have the same processing times on Stations ST_1 and ST_2 . For stations ST_2 and ST_3 we have the following assignment (or distance) matrix:

$$C^2 = \begin{bmatrix} 8.114 & 8.284 & 10.684 & 10.684 \\ 8.114 & 8.284 & 10.864 & 10.684 \\ 3.253 & 4.142 & 9.023 & 9.023 \\ 3.758 & 4.473 & 9.061 & 9.087 \end{bmatrix}.$$

In Step 1 of the lower bounding procedure, we find $f(\lambda_k^*)$ using a line search procedure. For a given λ value we solve an assignment problem. For example, when $\lambda = 0.4$ we solve an assignment problem with the following matrix:

$$\lambda C + (1 - \lambda)C^2 = \begin{bmatrix} 7.354 & 7.094 & 7.712 & 9.272 \\ 7.471 & 7.253 & 8.067 & 9.360 \\ 5.856 & 6.262 & 9.023 & 9.484 \\ 6.159 & 6.460 & 9.061 & 9.523 \end{bmatrix}.$$

The optimal assignment for $\lambda C + (1 - \lambda)C^2$ is $x_{1,3} = x_{2,4} = x_{3,1} = x_{4,2} = 1$, with an assignment value of 29.388. The expected batch cycle time of schedule $S = (3, 4, 1, 2)$ is equal to 34.801, which would be one of the solutions generated by the first approximate solution procedure.

For the second approximate solution procedure let us use schedule $S = (1, 4, 3, 2)$ as the initial solution. The expected batch cycle time of this schedule is equal to 36.128. In Step 1 of the procedure we need to find $a_{i,j}$ values. For example, according to Equation (??) $a_{1,1}$ would be given by

$$\begin{aligned} a_{1,1} &= Pr\{P_{1,1} \geq 6.520\} + Pr\{P_{1,2} \geq 9.454\} + Pr\{P_{1,3} \geq 10.685\} \\ &= e^{-\left(6.520 \frac{\Gamma(\frac{1}{\alpha})}{\alpha E[P_{1,1}]}\right)} + e^{-\left(9.454 \frac{\Gamma(\frac{1}{\alpha})}{\alpha E[P_{1,2}]}\right)} + e^{-\left(10.685 \frac{\Gamma(\frac{1}{\alpha})}{\alpha E[P_{1,3}]}\right)} = 0.400 + 0.000 + 0.225. \end{aligned}$$

The complete assignment matrix is given by

$$A = \begin{bmatrix} 0.625 & 0.435 & 0.682 & 0.444 \\ 0.960 & 0.538 & 0.565 & 0.920 \\ 0.430 & 0.323 & 0.428 & 0.705 \\ 0.471 & 0.359 & 0.643 & 0.450 \end{bmatrix},$$

and the solution of the assignment problem results in assignment $x_{1,4} = x_{2,3} = x_{3,1} = x_{4,2} = 1$, which is equivalent to sequence $S = (3, 4, 2, 1)$ whose expected batch cycle time is equal to 35.419.

Computational Results

2-Station Case

We first report our computational experience with the lower bounding and approximate solution procedures we presented for the 2-station case. In Table ?? computational experience with small problems for which the optimal sequences can be generated through complete enumeration is presented. Problems with 8 jobs are considered, and expected values of processing times are created using a discrete $U(1, 100)$ distribution. The shape parameter of the Weibull distribution is determined according to different levels of $s.c.v.$'s considered in our computational analysis. For each $s.c.v.$ value 100 problems are randomly created. Table ?? reports the minimum, average, and maximum values of deviations of lower bounds, which are obtained as explained in Theorem 1, from the optimal expected batch cycle time values as a percentage of optimal expected batch cycle time values. In the last three columns of Table ??, the performance of the Gilmore-Gomory¹³ procedure relative to optimal expected batch cycle time values is reported. Table ?? reports the performance of the Gilmore-Gomory¹³ procedure relative to the lower bound values for $n = 25$ and $n = 50$. As the results of Tables ?? and ?? indicate, the quality of the lower bounds, and the performance of the approximate solution procedure are very satisfactory.

m -Station Case

Our computational experience with the lower bounding and heuristic procedures are reported in Tables ?? and ?. We consider 8-job 4-station and 8-job 6-station problems for which we can determine the optimal solutions through complete enumeration for benchmarking purposes. In the first columns of Tables ?? and ?? we tabulate different $s.c.v.$ values considered in our computational analysis. For each $s.c.v.$ value 100 problems are created using processing time generation scheme described earlier (Weibull distributions with a common shape parameter and expected values drawn from a discrete $U(1, 100)$ distribution). For each $s.c.v.$ value two rows of information are presented. The first (second) row tabulates the average value (the minimum and maximum values over the set 100 problems) of the performance measure under consideration. In the second columns of the tables, we present the performance of the lower bounding procedure in terms its deviation from the optimal values, where $f_{\text{Opt.}}$ is the optimal solution value, and $f_{\text{Lower Bound}}$ is the lower bound value. In Column 3 we present the performance of the first approximate solution procedure (Heuristic 1). The performance of the second approximate solution procedure is presented in column 4 when the assignment costs are

computed using Equation (??) (Heuristic 2a), and in Column 5 when they are computed using with Equation (??) (Heuristic 2b). We have started the second approximate solution procedure with a randomly chosen sequence's expected subcycle time values and iterated between the first and second steps of the procedure 10 times.

The results of Tables ?? and ?? indicate that the performance of the 2-station lower bound gets worse with the increasing number of stations, and the amount of variability that is present in processing times. Among the two proposed approximate solution procedures, the second approximate solution procedure with Equation ?? provides a very good level of performance. Our computational results demonstrate that the problem can be approximately solved with relatively small deviations from the optimal values. Note that the computational complexity of the approximate solution procedures is equivalent to that of solving a linear assignment problem.

When the available information about the processing times is limited to their expected values, we can only solve the deterministic sequencing problem by assuming that the processing times would be equal to their expected values. Our analysis of this case starts with the performance evaluation of the sequence obtained by solving the deterministic problem. Problems with 8 jobs and 4 stations whose the optimal solutions can be determined through complete enumeration are considered. The performance of the optimal deterministic sequence is analyzed in the $[0.2, 4.0]$ range of squared coefficient of variation. For each *s.c.v.* value 20 problems are created using the processing time generation scheme described earlier in the computational results section. For each problem first the performance of the sequence which is generated by solving the deterministic version of the problem is evaluated and compared with the optimal expected batch cycle time value that would be obtained in the presence of information on processing time distributions. In Figure ??, the minimum, average, and maximum values (over the set of 20 problems) of the difference between these two values are presented as a percentage of the optimal expected batch cycle time value. In Figure ??, the same analysis is presented for the variances of the deterministic schedule when it is used in the stochastic problem and the sequence that minimizes the variance of the stochastic problem. The information presented in both figures indicate that the performance of the sequence which is obtained by solving the deterministic version of the problem is within a few percent of the optimal expected batch cycle time and variance values. These results clearly indicate that, for the problem sets considered in our analysis, the optimal sequence determination problem can be approximately solved even in the absence of information other than the expected processing times values. The performance of the sequence obtained by solving the deterministic problem is very satisfactory in terms of

its both expected value and variance of batch cycle time.

For performance evaluation, however, we do not have much to do in the absence of information other than the expected processing time values. In Figure ?? we present the ratio of the optimal expected batch cycle time value and the optimal cycle time value of the deterministic problem using the format of Figures ?? and ?. As the results of Figure ?? indicate, it is very difficult to estimate the expected batch cycle time with information limited to expected processing times.

The analysis is concluded with two more figures. In Figure ?? the *s.c.v.* value of the optimal schedule for the batch cycle time performance criterion is presented as a function of the amount variation present in individual processing times. The results of Figure ?? demonstrate that the variation in the batch cycle time remains relatively small even with very high variation in individual processing times. This observation leads to the conclusion that the batch cycle time criterion's *s.c.v.* value is relatively insensitive to errors that we might commit in estimating second and higher moments of the processing times distributions. In Figure ?? we present the same information for the sequence obtained by solving the deterministic problem. As we have shown in Figures ?? and ??, the performance of the deterministic optimal sequence is very close to that of the optimal sequence, and the same performance similarity is observed in the *s.c.v.* values.

Conclusion

In this paper we have addressed the stochastic synchronous assembly line scheduling problem for the cycle time performance criterion. Synchronous assembly lines are widely used in the assembly and production of various items. Our modeling framework incorporates variability in processing times, and thus presents a more realistic representation of the actual problem.

Our scheduling problem can be considered as the first part of a two stage decision making process. In the first stage of our problem we determine a sequence of jobs to control the flow of products in the assembly line. In the second stage of the problem, we estimate performance of the sequence that has been chosen in the first stage. The approximate solution procedures we have presented for the first stage of the problem provide satisfactory results in terms of their relative deviations from optimal solution values. We have also demonstrated that the second stage of the problem is straight forward when the processing time distributions are known. We have presented a simple framework to determine the expected value and variance of the batch cycle performance criterion, and demonstrated that the performance of a given sequence can

be determined in a computationally efficient manner.

When the information on processing times is limited to their expected values, the first stage of the problem can be tackled by solving the deterministic synchronous assembly line scheduling problem with the given expected processing time values. As we have demonstrated with our computational analysis, the performance of this sequence would be very satisfactory when it is implemented. However, since the performance evaluation stage of the problem cannot be realized in advance, one might expect that it would be very difficult to perform planning activities such as due date setting. On the other hand, because of the repetitive nature of the cyclic scheduling problem, the performance evaluation stage may be realized by observing the performance of the sequence in the first few batches and then by employing statistical inference techniques. The issue of performance evaluation when the available information is limited to the expected values of processing times requires further investigation.

Appendices

Proof of the Theorem 2

Let $c_{i,\Phi(j)} = E[\max\{P_{\Phi(j),1}, P_{i,2}\}]$. $c_{i,\Phi(j)}$ is the expected subcycle value if job $J_{\Phi(j)}$ is scheduled next to job J_i . Then $C = [c_{i,j}]$ is the distance matrix for the TSP associated with the two-station problem. Let

$$d_{i,\Phi(j)} = c_{i,\Phi(j)} + c_{i+1,\Phi(j-1)} - c_{i,\Phi(j-1)} - c_{i+1,\Phi(j)}. \quad (9)$$

If the matrix $D = [d_{i,\Phi(j)}]$ is nonnegative, we say that C is a *cumulative* distribution matrix generated by the density matrix D . (Gilmore *et al.*¹²) Substituting $c_{i,\Phi(j)} = E[\max\{P_{\Phi(j),1}, P_{i,2}\}]$ into Equation (??) yields

$$\begin{aligned} d_{i,\Phi(j)} &= E[\max\{P_{\Phi(j),1}, P_{i,2}\}] + E[\max\{P_{\Phi(j-1),1}, P_{i+1,2}\}] \\ &\quad - E[\max\{P_{\Phi(j-1),1}, P_{i,2}\}] - E[\max\{P_{\Phi(j),1}, P_{i+1,2}\}] \\ &= E[P_{\Phi(j),1} + P_{i,2} - \min\{P_{\Phi(j),1}, P_{i,2}\}] + E[P_{\Phi(j-1),1} + P_{i+1,2} - \min\{P_{\Phi(j-1),1}, P_{i+1,2}\}] \\ &\quad - E[P_{\Phi(j-1),1} + P_{i,2} - \min\{P_{\Phi(j-1),1}, P_{i,2}\}] - E[P_{\Phi(j),1} + P_{i+1,2} - \min\{P_{\Phi(j),1}, P_{i+1,2}\}] \\ &= E[\min\{P_{\Phi(j-1),1}, P_{i,2}\}] + E[\min\{P_{\Phi(j),1}, P_{i+1,2}\}] \\ &\quad - E[\min\{P_{\Phi(j),1}, P_{i,2}\}] - E[\min\{P_{\Phi(j-1),1}, P_{i+1,2}\}] \\ &= \int_{t=0}^{\infty} [\theta_{\Phi(j-1),1}(t) - \theta_{\Phi(j),1}(t)][\theta_{i,2}(t) - \theta_{i+1,2}(t)]dt, \end{aligned}$$

where $\theta_{i,j}$ is the cumulative probability function of random variable $P_{i,j}$. From the indexing of operations and stochastic ordering assumption, we have that random variable $P_{\Phi(j-1),1}(P_{i,2})$ is

stochastically larger than random variable $P_{\Phi(j),1}(P_{i+1,2})$, and therefore $d_{i,\Phi(j)} \geq 0$. $\square$

Expected value and Variance of N Weibull Random Numbers

Let $X_1, X_2, \dots, X_N$ be independent Weibull random variables with different means and the same shape parameter α . Let $X_{(1)}$ be the maximum of $X_1, X_2, \dots, X_N$. Let $E[X_i]$ be the mean and $F_i(x)$ be the cumulative distribution function of X_i given as

$$E[X_i] = \mu_i = \frac{\beta_i}{\alpha} \Gamma(1/\alpha), \quad (10)$$

and

$$F_i(x) = P[X_i \leq x] = 1 - e^{-(x/\beta_i)^\alpha}. \quad (11)$$

Then the cumulative distribution function of $X_{(1)}$ is $F_{(1)}(x) = P[X_{(1)} \leq x] = P[X_1 \leq x, X_2 \leq x, \dots, X_N \leq x]$. Since $X_1, X_2, \dots, \text{and } X_N$ are independent

$$F_{(1)}(x) = P[X_1 \leq x, X_2 \leq x, \dots, X_N \leq x] = P[X_1 \leq x] \cdot P[X_2 \leq x] \cdot \dots \cdot P[X_N \leq x] = \prod_{i=1}^N F_i(x). \quad (12)$$

Then the expected value of the maximum of N Weibull random variables can be obtained from the integral

$$E[X_{(1)}] = \int_0^\infty (1 - F_{(1)}(x)) dx = \int_0^\infty \left(1 - \prod_i F_i(x)\right) dx \quad (13)$$

Inserting Equation ?? for $F_i(x)$ into the above equation yields

$$E[X_{(1)}] = \int_0^\infty \left[1 - \prod_i \left(1 - e^{-(x/\beta_i)^\alpha}\right)\right] dx. \quad (14)$$

Expanding the above equation yields

$$E[X_{(1)}] = \int_0^\infty \left[\sum_i e^{-\frac{x^\alpha}{\beta_i^\alpha}} - \sum_{i>j} e^{-x^\alpha \left(\frac{1}{\beta_i^\alpha} + \frac{1}{\beta_j^\alpha}\right)} + \sum_{i>j>k} e^{-x^\alpha \left(\frac{1}{\beta_i^\alpha} + \frac{1}{\beta_j^\alpha} + \frac{1}{\beta_k^\alpha}\right)} - (-1)^N \underbrace{\sum_{i>j>\dots>z} e^{-x^\alpha \left(\frac{1}{\beta_i^\alpha} + \dots + \frac{1}{\beta_k^\alpha}\right)}}_N \right] dx. \quad (15)$$

Now consider the first integral:

$$I_1 = \int_0^\infty \sum_i e^{-\frac{x^\alpha}{\beta_i^\alpha}} dx. \quad (16)$$

Let $t = x^\alpha/\beta_i^\alpha$, and thus $dt = \frac{\alpha x^{\alpha-1}}{\beta_i^\alpha} dx$, and $dx = \frac{\beta_i}{\alpha} t^{1/\alpha-1} dt$. Therefore the above integral is equivalent to $I_1 = \sum_i \int_0^\infty \frac{\beta_i}{\alpha} t^{1/\alpha-1} e^{-t} dt$. Since the gamma function is defined as $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ for any real number $z > 0$, the above equation simplifies to

$$I_1 = \sum_i \frac{\beta_i}{\alpha} \Gamma\left(\frac{1}{\alpha}\right) \quad (17)$$

which is the sum of the means of X_i . That is $I_1 = \sum_i \mu_i$. Now consider the second integral of Equation ??

$$I_2 = \int_0^\infty \sum_{i>j} e^{-x^\alpha \left(\frac{1}{\beta_i^\alpha} + \frac{1}{\beta_j^\alpha} \right)} dx \quad (18)$$

Let $\frac{1}{\beta^\alpha} = \frac{1}{\beta_i^\alpha} + \frac{1}{\beta_j^\alpha}$ or $\beta = \frac{\beta_i \beta_j}{(\beta_i^\alpha + \beta_j^\alpha)^{1/\alpha}}$. Then $I_2 = \int_0^\infty \sum_{i>j} e^{-\frac{x^\alpha}{\beta^\alpha}} dx$ which is in the same form as Equation ?. Therefore, following Equation ?, $I_2 = \sum_i \frac{\beta}{\alpha} \Gamma\left(\frac{1}{\alpha}\right)$. In order to write the above equation in terms of μ_i and μ_j note that $\mu_i = \frac{\beta_i}{\alpha} \Gamma\left(\frac{1}{\alpha}\right)$, and therefore $\beta_i = \frac{\mu_i \alpha}{\Gamma(1/\alpha)}$. Inserting this identity into β gives

$$\beta = \frac{\beta_i \beta_j}{(\beta_i^\alpha + \beta_j^\alpha)^{1/\alpha}} = \frac{\frac{\mu_i \alpha}{\Gamma(1/\alpha)} \frac{\mu_j \alpha}{\Gamma(1/\alpha)}}{(\mu_i^\alpha + \mu_j^\alpha)^{1/\alpha} \frac{\alpha}{\Gamma(1/\alpha)}} = \frac{\mu_i \mu_j}{(\mu_i^\alpha + \mu_j^\alpha)^{1/\alpha}} \frac{\alpha}{\Gamma(1/\alpha)}, \quad (19)$$

and thus

$$I_2 = \sum_{i>j} \frac{\mu_i \mu_j}{(\mu_i^\alpha + \mu_j^\alpha)^{1/\alpha}} = \sum_{i>j} \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha} \right)^{-1/\alpha} \quad (20)$$

Evaluating the other terms of Equation ? finally produces an expression for the expected value of the maximum of N Weibull random variables given as

$$E[X_{(1)}] = \sum_i \mu_i - \sum_{i>j} \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha} \right)^{-1/\alpha} + \sum_{i>j>k} \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha} + \frac{1}{\mu_k^\alpha} \right)^{-1/\alpha} - (-1)^N \underbrace{\sum_{i>j>\dots>z}_N \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha} + \dots + \frac{1}{\mu_z^\alpha} \right)^{-1/\alpha}}. \quad (21)$$

If there are two random variables X_1 and X_2 then the expected value of the maximum of X_1 and X_2 is

$$E[X_{(1)}] = \mu_1 + \mu_2 - \left(\frac{1}{\mu_1^\alpha} + \frac{1}{\mu_2^\alpha} \right)^{-1/\alpha}. \quad (22)$$

Let $Var[X_{(1)}]$ be the variance of the maximum of N Weibull random variables obtained from $Var[X_{(1)}] = E[X_{(1)}^2] - E^2[X_{(1)}]$. The second moment of $X_{(1)}$ can be obtained from the following integral

$$E[X_{(1)}^2] = 2 \int_0^\infty x (1 - F_{(1)}(x)) dx = 2 \int_0^\infty x \left(1 - \prod_i F_i(x) \right) dx. \quad (23)$$

Inserting Equation ? into Equation ? and expanding the product gives

$$E[X_{(1)}^2] = 2 \int_0^\infty \left[x \sum_i e^{-\frac{x^\alpha}{\beta_i^\alpha}} - x \sum_{i>j} e^{-x^\alpha \left(\frac{1}{\beta_i^\alpha} + \frac{1}{\beta_j^\alpha} \right)} + x \sum_{i>j>k} e^{-x^\alpha \left(\frac{1}{\beta_i^\alpha} + \frac{1}{\beta_j^\alpha} + \frac{1}{\beta_k^\alpha} \right)} - (-1)^N x \underbrace{\sum_{i>j>\dots>z}_N e^{-x^\alpha \left(\frac{1}{\beta_i^\alpha} + \dots + \frac{1}{\beta_z^\alpha} \right)}} \right] dx \quad (24)$$

Now consider the first integral in the above term

$$I'_1 = \int_0^\infty x \sum_i e^{-\frac{x^\alpha}{\beta_i^\alpha}} dx.$$

Similar to the previous case after the substituting $t = x^\alpha / \beta_i^\alpha$ and writing the result in terms of the gamma function we obtain $I'_1 = \sum_i \frac{\beta_i^2}{\alpha} \Gamma\left(\frac{2}{\alpha}\right)$, or in terms of μ_i $I'_1 = \sum_i C(\alpha) \mu_i^2$, where $C(\alpha) = \frac{\alpha \Gamma(2/\alpha)}{[\Gamma(2/\alpha)]^2}$. Following the above steps for the second term of Equation ?? gives $I'_2 = C(\alpha) \sum_{i>j} \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha}\right)^{-2/\alpha}$. Finally, we have

$$\begin{aligned} E[X_{(1)}^2] &= 2C(\alpha) \left[\sum_i \mu_i^2 - \sum_{i>j} \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha}\right)^{-2/\alpha} + \sum_{i>j>k} \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha} + \frac{1}{\mu_k^\alpha}\right)^{-2/\alpha} - \dots \right. \\ &\quad \left. - (-1)^N \sum_{\underbrace{i>j>\dots>z}_N} \left(\frac{1}{\mu_i^\alpha} + \frac{1}{\mu_j^\alpha} + \dots + \frac{1}{\mu_z^\alpha}\right)^{-2/\alpha} \right]. \end{aligned} \quad (25)$$

If there are two random variables X_1 and X_2 then the variance of the maximum of X_1 and X_2 is given by

$$Var[X_{(1)}] = 2C(\alpha) \left[\mu_1^2 + \mu_2^2 - \left(\frac{1}{\mu_1^\alpha} + \frac{1}{\mu_2^\alpha}\right)^{-2/\alpha} \right] - \left[\mu_1 + \mu_2 - \left(\frac{1}{\mu_1^\alpha} + \frac{1}{\mu_2^\alpha}\right)^{-1/\alpha} \right]^2 \quad (26)$$

References

- [1] R. J. Schonberger. *Japanese Manufacturing Techniques: Nine Hidden Lessons in Simplicity*. The Free Press, New York, 1982.
- [2] L. Fuxman. Mixed-model assembly lines under cyclic production: Models for productivity enhancement and teamwork. Working paper, Wharton School, University of Pennsylvania, 1994.
- [3] S. T. McCormick, M. L. Pinedo, S. Shenker, and B. Wolf. Sequencing in an assembly line with blocking to minimize cycle time. *Operations Research*, 37(6):925–935, 1989.
- [4] S. Karabati and P. Kouvelis. Cyclic scheduling in flow lines: Modeling observations, effective heuristics and an optimal solution procedure. *Naval Research Logistics*, 43:211–231, 1996.
- [5] I. N. Kamal Abadi, N. G. Hall, and C. Sriskandarajah. Minimizing cycle time in a blocking flowshop. Technical report, University of Toronto, 1995.
- [6] N. G. Hall and C. Sriskandarajah. A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44(3), 1996.

- [7] P. Kouvelis and S. Karabati. Cyclic scheduling in synchronous production lines. Working paper no:1997/16, Koç University, İstanbul, 1997.
- [8] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1995.
- [9] U. S. Rao and P. L. Jackson. Estimating performance measures in repetitive manufacturing environments via stochastic cyclic scheduling. *IIE Transactions*, 28:929–939, 1996.
- [10] R. Bowman and J. Muckstadt. Stochastic analysis of cyclic schedules. *Operations Research*, 41(5):947–958, 1993.
- [11] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, Inc., 1982.
- [12] P. C. Gilmore, E. L. Lawler, and D. B. Shmoys. Well-solved special cases. In E. L. Lawler and D. B. Shmoys J. K. Lenstra, A. H. G. Rinnooy Kan, editors, *The Traveling Salesman Problem*. John Wiley & Sons, 1985.
- [13] P. C. Gilmore and R. E. Gomory. Sequencing a one state-variable machine: a solvable case of the traveling salesman problem. *Operations Research*, 12:655–679, 1964.
- [14] H. Matsuo. Cyclic sequencing problems in the two-machine permutation flowshop: Complexity, worst-case and average-case analysis. *Naval Research Logistics Quarterly*, 37:679–694, 1990.

Table 1: Computational Results for the 2-Station Case ($n = 8$).

Table 2: Computational Results for the 2-Station Case ($n = 25$ and $n = 50$).

Table 3: Computational Results for $n = 8$ and $m = 4$.

Table 4: Computational Results for $n = 8$ and $m = 6$.

Figure 1: Performance of Optimal Deterministic Sequence in the ($\min E[CT]$) Problem.

Figure 2: Performance of Optimal Deterministic Sequence in the ($\min V(CT)$) Problem.

Figure 3: Comparison of Cycle Times.

Figure 4: *s.c.v.* of the Batch Cycle Time with the Optimal Stochastic Sequence.

Figure 5: *s.c.v.* of the Batch Cycle Time with the Optimal Deterministic Sequence.

Table 1: Computational Results for the 2-Station Case ($n = 8$).

<i>s.c.v.</i>	$\frac{f_{\text{Optimal}} - f_{\text{Lower Bound}}}{f_{\text{Optimal}}} \times 100$			$\frac{f_{\text{Heuristic}} - f_{\text{Optimal}}}{f_{\text{Optimal}}} \times 100$		
	min.	ave.	max.	min.	ave.	max.
0.25	0.0000	0.1800	2.3447	0.0000	0.0523	2.5988
0.50	0.0000	0.1751	2.7761	0.0000	0.0114	0.3481
0.75	0.0000	0.1758	1.6964	0.0000	0.0248	1.2156
1.00	0.0000	0.0787	0.8378	0.0000	0.0091	0.2745
1.25	0.0000	0.1136	1.0071	0.0000	0.0235	0.6775
1.50	0.0000	0.0644	0.4244	0.0000	0.0269	1.2245
1.75	0.0000	0.0571	0.7057	0.0000	0.0206	0.6506
2.00	0.0000	0.1096	0.8969	0.0000	0.0130	0.2497

Table 2: Computational Results for the 2-Station Case ($n = 25$ and $n = 50$).

<i>s.c.v.</i>	$n = 25$			$n = 50$		
	$\frac{f_{\text{Heuristic}} - f_{\text{Lower Bound}}}{f_{\text{Lower Bound}}} \times 100$			$\frac{f_{\text{Heuristic}} - f_{\text{Lower Bound}}}{f_{\text{Lower Bound}}} \times 100$		
	min.	ave.	max.	min.	ave.	max.
0.25	0.0000	0.0051	0.0826	0.0000	0.0018	0.0474
0.50	0.0000	0.0046	0.0663	0.0000	0.0004	0.0067
0.75	0.0000	0.0052	0.0661	0.0000	0.0003	0.0037
1.00	0.0000	0.0041	0.0919	0.0000	0.0003	0.0042
1.25	0.0000	0.0031	0.0263	0.0000	0.0002	0.0043
1.50	0.0000	0.0037	0.0633	0.0000	0.0004	0.0207
1.75	0.0000	0.0031	0.0696	0.0000	0.0002	0.0024
2.00	0.0000	0.0020	0.0190	0.0000	0.0002	0.0024

Table 3: Computational Results for $n = 8$ and $m = 4$.

$s.c.v.$	$\frac{f_{Opt.} - f_{Lower\ Bound}}{f_{Opt.}} \times 100$	$\frac{f_{Heuristic(1)} - f_{Opt.}}{f_{Opt.}} \times 100$	$\frac{f_{Heuristic(2a)} - f_{Opt.}}{f_{Opt.}} \times 100$	$\frac{f_{Heuristic(2b)} - f_{Opt.}}{f_{Opt.}} \times 100$
0.25	13.8635 [04.1684, 20.2986]	04.7926 [00.2673, 14.9378]	02.9133 [00.0000, 10.4060]	03.8617 [00.0136, 09.8851]
0.50	17.7066 [10.6104, 25.7110]	03.8705 [00.0000, 11.2932]	02.2428 [00.0000, 09.1682]	03.0204 [00.0000, 07.9692]
0.75	20.5723 [11.2392, 26.4452]	03.0011 [00.0000, 06.8900]	01.7239 [00.0000, 06.2453]	02.6786 [00.0960, 08.8354]
1.00	22.3378 [15.7273, 27.8375]	02.8977 [00.1478, 07.2806]	01.6675 [00.0000, 06.1156]	02.5175 [00.0000, 07.5684]
1.25	24.1325 [11.6538, 30.7625]	02.4497 [00.0000, 06.2459]	01.3512 [00.0000, 04.7573]	02.2048 [00.1068, 05.1999]
1.50	25.5285 [14.4945, 31.5867]	02.3849 [00.0000, 05.2081]	01.2203 [00.0000, 03.7968]	02.0916 [00.2375, 05.4841]
1.75	26.2162 [16.6448, 33.7589]	02.2993 [00.0000, 05.5804]	01.0843 [00.0000, 03.7553]	01.8232 [00.0000, 05.0473]
2.00	27.2399 [17.9197, 34.6020]	01.9751 [00.0000, 06.7351]	01.1021 [00.0000, 05.2294]	01.8506 [00.0596, 05.3606]

Table 4: Computational Results for $n = 8$ and $m = 6$.

$s.c.v.$	$\frac{f_{Opt.} - f_{Lower\ Bound}}{f_{Opt.}} \times 100$	$\frac{f_{Heuristic(1)} - f_{Opt.}}{f_{Opt.}} \times 100$	$\frac{f_{Heuristic(2a)} - f_{Opt.}}{f_{Opt.}} \times 100$	$\frac{f_{Heuristic(2b)} - f_{Opt.}}{f_{Opt.}} \times 100$
0.25	21.7380 [12.1045, 28.2790]	04.0780 [00.0000, 08.0760]	02.1304 [00.0000, 08.9543]	03.1340 [00.0000, 07.1591]
0.50	26.9498 [19.1999, 33.4736]	03.5305 [00.7357, 09.1705]	01.8190 [00.0000, 08.2716]	02.9749 [00.0000, 08.8846]
0.75	30.4491 [22.4456, 36.6803]	02.7318 [00.0000, 05.5938]	01.5827 [00.0000, 05.4322]	02.7288 [00.0000, 06.7037]
1.00	33.9279 [28.2698, 42.0292]	02.6281 [00.1172, 07.1233]	01.3779 [00.0000, 04.6545]	02.1931 [00.0000, 07.0264]
1.25	35.2271 [27.2899, 41.1964]	02.1855 [00.0000, 04.8012]	01.2319 [00.0000, 03.5321]	02.0787 [00.0000, 05.6750]
1.50	37.0098 [29.0632, 43.1780]	02.1178 [00.0000, 05.6955]	01.1181 [00.0000, 03.6747]	01.8716 [00.0000, 03.9247]
1.75	38.0638 [29.4946, 45.5657]	01.8645 [00.3970, 04.8050]	01.0194 [00.0000, 03.4572]	01.7123 [00.0000, 03.8655]
2.00	39.3231 [30.5347, 45.0369]	01.6617 [00.1017, 04.5125]	00.8571 [00.0000, 02.3782]	01.6761 [00.0000, 03.9784]

Figure 1: Performance of Optimal Deterministic Sequence in the $(\min E[CT])$ Problem.

Figure 2: Performance of Optimal Deterministic Sequence in the $(\min V(CT))$ Problem.

Figure 3: Comparison of Cycle Times.

Figure 4: *s.c.v.* of the Batch Cycle Time with the Optimal Stochastic Sequence.

Figure 5: *s.c.v.* of the Batch Cycle Time with the Optimal Deterministic Sequence.