Setting planned job release times in stochastic assembly systems with resource constraints

DONGPING SONG†*, C. HICKS† and C. F. EARL‡

This paper considers a stochastic assembly system operating on a make-to-order basis with complex product structure and resource constraints. The problem is to find the optimal planned job release times by minimizing the expected sum of the work-in-progress holding cost, product earliness cost and product tardiness cost. A perturbation analysis algorithm is developed to derive the gradient estimate of the cost function with respect to the job release times. This gradient estimate is shown to be unbiased and may lead to the optimal solution by using a stochastic approximation method. Moreover, a procedure is presented to adjust planned job release times to meet service level constraint for each individual job. Numerical examples, which use manufacturing and assembly data from a capital goods company, are given to demonstrate the results.

1. Introduction

Determining planned job release times is an important problem in manufacturing systems. The effect of job release times on the system performance is complicated, especially for assembly systems with complex product structure and resource constraints. For example, if a job is released too early it will result in unnecessary holding cost due to waiting for the required resource to become available or waiting for other jobs, which go into the same assembly activity. On the other hand, if a job is released too late then it may also result in extra cost. First, some jobs that go into the same assembly activity have to wait. Second, some jobs that use the same resource have to wait for the resource to become available. Third, successive jobs specified in the product structure are delayed. This problem is relatively easy in deterministic circumstances if the job sequence at each resource is given. However, it becomes difficult in stochastic situations since jobs cannot be finished exactly at the expected times. In addition, uncertainties in processing times accumulate and interact due to the complex product structure and resource constraints.

Yano (1987a) considered the optimal planned lead-time design problem in a stochastic serial production system by minimizing the expected sum of inventory holding costs and tardiness cost. In her formulation, the start lead-time is actually the same as job release time. She proposed an analytical method to solve the problem and further extended the method to a two-level assembly system and a two-level distribution type network (Yano 1987b, c). Gong *et al.* (1994) resolved the optimal lead-time-planning problem in a serial production system by using an equivalent

Received January 2000.

† Department of Mechanical, Materials and Manufacturing Engineering, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU, UK.

‡ Department of Design and Innovation, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK.

* To whom correspondence should be addressed. e-mail: d.p.song@ncl.ac.uk

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2001 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540010022377

serial inventory model. However, their analytical methods are very difficult to extend to complex systems. In fact, stochastic models of manufacturing systems have proven to be analytically intractable except for some simple systems (Buzzacott and Shanthikumar 1993).

Another possible way to solve this problem is to use the Stochastic Approximation (SA) method to optimize the planned job release times (termed control parameters). The key step in using stochastic approximation is to find the effective gradient information of the cost function with respect to the control parameters. Recently, the Perturbation Analysis (PA) technique has been successfully applied to obtain the gradient estimate in various stochastic manufacturing situations (Donohue and Spearman 1993, Yan *et al.* 1994, Bremaud *et al.* 1997, Song and Sun 1998, Hasan and Spearman 1999, Song *et al.* 1999, Tang and Boukas 1999). Hasan and Spearman (1999) considered a make-to-order environment and applied perturbation analysis and stochastic approximation to optimize the material release times for a fixed sequence of work. However, their system, which consists of a series of G/G/1 queues, is relatively simple. They did not consider product structure, assembly activities or work-in-progress holding cost. For multistage assembly systems, the present authors have used PA and SA to design optimal planned lead-time start times based upon an infinite capacity model. Although the infinite capacity assumption is often made in Material Requirements Planning (MRP) and scheduling using Critical Path Methods (CPM) or the Programme Evaluation and Review Technique (PERT), it is more realistic to take into account finite capacity or resource constraints in manufacturing planning.

This paper extends the PA technique to a general stochastic assembly system with complex product structure and resource constraints. In such systems a plan is characterized by the job sequence on each resource and the timing of each job (in terms of job starting time and due date). Owing to the uncertainty in the production process, the job arrival sequence on a resource may be different from the scheduled one. Two approaches could be taken to deal with this situation. One is to keep the original plan without changing the job sequence. The other is to reschedule the jobs using some kind of priority rule. This paper assumes the first situation. That means, the sequence of jobs on each resource is predetermined and fixed. The reasons for this assumption are: (1) rescheduling the job sequence may be good in the local sense but may be not in the global sense; (2) rescheduling causes further deviation from the original plan; and (3) the interacting effects of job sequence and job timing is very complicated in our system. Here the focus is on the job-timing problem. The first aim is to find the optimal planned release times (or planned processing start times) for each job on each resource by minimizing the expected sum of work-in-progress (WIP) holding cost, product earliness cost and product tardiness cost.

The second aim is to adjust further the planned job release times to meet service level constraints. Here the service level is defined as the probability of completing a job before its due date. For make-to-order systems, service level is a very important performance measure. A higher service level in the production process means that a production plan is more reliable and will more probably be executed smoothly. In particular, an appropriate service level in the final product is a critical factor to promote customer satisfaction and generate further business (Hendry and Kingsman 1989). In the present system, a job is related to a part and each job is assumed to have one operation. For simplicity of narrative, part, job and operation are used interchangeably. The product structure and process routing determine the job precedence. It is assumed that the due date of a job is equal to the planned release time of the job that immediately succeeds it. A procedure is proposed to adjust the planned job release times to meet specified service level constraints. This is a trade-off between reducing cost and achieving service level.

The rest of the paper is organized as follows. In Section 2 the problem is formulated and notations are given. In Section 3 a PA algorithm is presented and the unbiasedness of the PA gradient estimate is proven. Section 4 simply describes the SA method. Section 5 considers the service level constraint problem. Numerical examples are given in Section 6 and conclusions are drawn in Section 7.

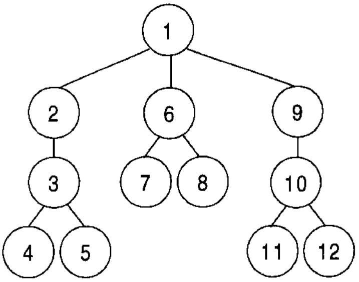
2. Problem formulation

Consider an assembly manufacturing system operating on a make-to-order basis. A product (or an order) involves many different jobs and many different resources. Some of these jobs are assembly activities. The precedence relationship between jobs is described by a tree-type product structure, which is determined by technical requirements. As an example, a real product structure from a capital goods company is shown in figure 1. Each node represents a job and the job at the highest level (root node) is sometimes termed a product. In the example the product structure is that of a subassembly used in the final product.

The product structure not only gives rise to the job precedence constraints, but also the job assembly coordination requirements. That means, all subassembly jobs that go to the same assembly activity should be scheduled to arrive as closely as possible. However, it is difficult in reality due to the uncertain processing times and resource constraints.

Planned job release times are introduced to control the production process. A job is not allowed to be released (or start processing) before the specified planned release time. However, if a job's preceding activities are finished later than its planned release time or the required resource is not available at its planned release time, then this job should start processing as soon as possible. There are several reasons to introduce the planned release times for all jobs in the product structure. First, it can coordinate the completion of assembly jobs to avoid holding cost. Second, it specifies a detailed schedule for resource allocation, that is, at what time each resource is allocated to each job. Third, it can reduce the variability of the process.

In this paper, it is assumed that the job sequence on each resource is predetermined and does not conflict with the job precedence constraints given in the product structure. Therefore, the planned job release times determines a production plan. It



Resource code	Job sequence
1001:	6, 1
1002:	10, 2
1003:	3, 9
1004:	4, 11
1005:	5, 12
1006:	8, 7

Figure 1. Product structure.

specifies the timings of all jobs and the timings of all resources. The objective is to find the optimal planned job release times that minimize the expected sum of holding costs and tardiness cost.

The notation used is defined as follows.

Γ job set,
L product set, i.e. the set of jobs that have no successive job,
M resource set,
$\mathrm{B}(i)$ set of jobs that immediately precedes job i.
$\sigma(i)$ job that immediately follows job i in the product structure,
$r(i)$ resource on which job i is processed,
$\varphi(i)$ job that immediately precedes job i on resource $r(i)$,
(i) job that immediately follows job i on resource $r(i)$,
s_i planned release time (i.e. planned processing start time) of job i
x_i processing time of job i, which is a known independent continuous random variable,
a_i actual processing start time of job i,
c_i processing completion time of job i,
d_i due date of job i.

Clearly, the production process can be described by $\{a_i\}$ and $\{c_i\}$:

$$a_i = \max\left(s_i, c_{\varphi(i)}, \{c_j : j \in \mathrm{B}(i)\}\right) \tag{1}$$

$$c_i = a_i + x_i. \tag{2}$$

On the right-hand side (RHS) of equation (1), the first term represents that job i cannot start processing before the planned job release time s_i; the second term describes that job i cannot start processing on the resource $r(i)$ until job $\varphi(i)$ is completed; and the third term represents that job i cannot start processing until all its subassembly jobs are finished. If $\varphi(i) \in \mathrm{B}(i)$ then the second term is not necessary. Let $\mathbf{s}$ be a vector composed of all planned job release times. The cost function is given by:

$$J(\mathbf{s}) = \mathrm{E}\left[\sum_{i\in\Gamma\backslash L} h_i(a_{\sigma(i)} - c_i) + \sum_{i\in\mathrm{L}} h_i \max(d_i - c_i, 0) + \sum_{i\in\mathrm{L}} h_i^- \max(c_i - d_i, 0)\right], \tag{3}$$

where $\Gamma\backslash\mathrm{L}$ is the difference set between Γ and L, h_i is the holding cost coefficient for job i, and h_i^- is the corresponding tardiness cost coefficient.

The problem is to find the optimal $\mathbf{s}$ by minimizing the above cost function. The key step to solve this optimization problem is to obtain the effective gradient information. In the next section PA is applied to derive an unbiased gradient estimate by observing a single sample process.

3. Gradient estimate and its unbiasedness

The PA technique is well addressed by Ho and Cao (1991), Glasserman (1991) and Cassandras (1993). The idea can be simply stated as follows. Consider a stochastic system. Let ω represent a realisation of the set of random sequences involved in the system. In our systems the random sequences consist of the random variables of operation times and ω is a particular set of values of these random variables. Let θ be a vector composed of n real parameters. Let $V(\theta, \omega)$ be the value of some system performance measure corresponding to (θ, ω), which is called a sample performance function. Define the system performance measure by $J(\theta) = \mathrm{E}[V(\theta, \omega)]$. The aim of PA is to obtain the gradient of $V(\theta, \omega)$ with respect to θ through a single sample realisation. That means, from one simulation run all n partial derivatives of $V(\theta, \omega)$ with respect to θ's elements can be obtained simultaneously (it is well known that if the pure simulation method is used, at least $n + 1$ simulations should be performed to get all n partial derivatives). Then this gradient information $\partial V(\theta, \omega)/\partial\theta$ is used as an estimate of $\partial J(\theta)/\partial\theta$. This may lead to an optimal or suboptimal θ if the gradient estimate is unbiased, i.e. $\partial J(\theta)/\partial\theta = \mathrm{E}[\partial V(\theta, \omega)/\partial\theta]$.

The advantages of the PA technique are: (1) it can save a lot of computational effort and simulation time compared with the pure simulation method; (2) SA based on a PA gradient estimate has faster convergence rates than SA based on finite difference estimator (Fu and Hu 1997); (3) PA does not require strict assumptions on the distributions of processing times and can also include other possible stochastic factors, e.g. machine failures and repairs (Song and Sun 1998); and (4) it can deal with complex manufacturing systems with deep and wide product structure and resource constraints.

Consider a sample process under a set of predetermined release times $\{s_i\}$. Let ω be the realisation of the set of random sequences in the system. Let Ω be the whole sample space. A sample cost function is defined by:

$$V(\mathbf{s}, \omega) = \sum_{i \in \Gamma \backslash \mathrm{L}} h_i(a_{\sigma(i)} - c_i) + \sum_{i \in \mathrm{L}} h_i \max(d_i - c_i, 0) + \sum_{i \in \mathrm{L}} h_i^- \max(c_i - d_i, 0). \quad (4)$$

For given $\mathbf{s}$ and ω, the sample process is deterministic and is conventionally termed a nominal path (NP). To simplify the notation, $\{a_i\}$ and $\{c_i\}$ are also used to represent the starting and completion times in the nominal path. Let planned release time s_j be perturbed to be $s_j + \Delta$, where Δ is a small positive number. The sample process for $\{s_j + \Delta, s_i, i \neq j, i \in \Gamma\}$ with the same stochastic condition ω is termed a perturbed path (PP). The perturbed path can be constructed by analysing the nominal path without running the sample realisation again. Let $\{a_i'\}$ and $\{c_j'\}$ denote the above sequences of the actual processing start times and completion times in the perturbed path.

From (1) and (2), a small positive perturbation Δ on s_j has no influence on the whole production process if $a_j > s_j$, but will generate a perturbation gain Δ on a_j' and c_j' if $a_j = s_j$. This perturbation gain may be propagated to other jobs along the perturbed path due to job precedence constraints and resource constraints. For example, in equation (1) if $a_i = c_{\varphi(i)}$, then the perturbation gain on $c'_{\varphi(i)}$ is propagated to a_i' and c_i'. Similarly, if $a_i = c_k (k \in \mathrm{B}(i))$, then the perturbation gain on c_k' is propagated to a_i' and c_i'. However, if $a_i = s_i (i \neq j)$ in (1), then no perturbation gain will be propagated to a_i' and c_i'. These perturbation generation and propagation rules are summarized as follows.

Proposition 1: For any given $\omega \in \Omega$, if $s_j (j \in \Gamma)$ is perturbed to be $s_j + \Delta$ and Δ is a sufficiently small positive number, then

(1) if $a_j > s_j$, then $a_i' \equiv a_i$ and $c_i' \equiv c_i'$, for $i \in \Gamma$;
(2) if $a_j = s_j$, then $a_j' = a_j + \Delta$ and $c_j' = c_j + \Delta$;
(3) if $a_i = s_i (i \neq j)$, then $a_i' = a_i$ and $c_i' = c_i$;
(4) if $a_i = c_{\varphi(i)} (i \neq j)$, then $a_i' = a_i + (c'_{\varphi(i)} - c_{\varphi(i)})$ and $c_i' = c_i + (c'_{\varphi(i)} - c_{(i)})$; and

(5) if $a_i = c_k(i \neq j, k \in \mathrm{B}(i))$, then $a_i' = a_i + (c_k' - c_k)$ and $c_\mathrm{i}' = c_i + (c_k' - c_k)$.

The first two rules describe perturbation generation and the last three rules describe perturbation propagation. In Proposition 1, since ω is given and Δ is sufficiently small, the whole perturbation gain Δ will be propagated along the perturbed path. That means, the perturbation gain on PP is only Δ or 0. Define

- $I(i) := 1\{a_i' \neq a_i\}$, where $1\{.\}$ is an indicate function, which takes 1 if $\{.\}$ is true, otherwise takes 0.

Note that a_i' and c_i' have the same perturbation gain by (2). The sequence $\{I(i), i \in \Gamma\}$ determines the difference between PP and NP. To implement perturbation propagation rules and determine $\{I(i), i \in \Gamma\}$ from the nominal path, a recursive procedure is developed. Initially, set $I(i) = 0, i \in \Gamma$. In the following procedure the parameter k represents a job code. If $a_k = s_k$, then call *Perturbation_propagation* (k), which is given by

```
Perturbation_propagation $(k)$ {
Step 1: set $I(k) = 1$;
Step 2: if $a_{\sigma(k)} = c_k$ and $\sigma(k) \notin \mathrm{L}$, then call Perturbation_propagation $(\sigma(k))$;
if $a_{\sigma(k)} = c_k$ and $\sigma(k) \in \mathrm{L}$, then set $I(\sigma(k)) = 1$.
Step 3: if  $(k)$ exists and $a_{\ (k)} = c_k$, then call Perturbation_propagation ( $(k)$);
Step 4: return.
}
```

The partial derivative of the sample cost function with respect to planned job release time is given in the following theorem.

Theorem 1—Gradient estimator: For any $\omega \in \Omega$ and $j \in \Gamma$, we have:

$$\partial V(\mathrm{s}, \omega)/\partial s_j = \sum_{i \in \Gamma \backslash \mathrm{L}} h_i \cdot (I(\sigma(i)) - I(i)) + \sum_{i \in \mathrm{L}} h_i \cdot I(i) \cdot 1\{d_i > c_i\} + \sum_{i \in \mathrm{L}} h_i^- \cdot I(i) \cdot 1\{d_i \leqslant c_i\}. \tag{5}$$

Proof: Let Δ be a sufficiently small positive number. With a slight abuse of the notation, let $V(s_j, \omega)$ and $V(s_j + \Delta, \omega)$ denote the sample cost functions for NP and PP respectively. From (4):

$$\begin{aligned} V(s_j + \Delta, \omega) - V(s_j, \omega) = & \sum_{i \in \Gamma \backslash \mathrm{L}} h_i (a_{\sigma(i)}' - c_\mathrm{i}') + \sum_{i \in \mathrm{L}} h_i \max(d_i - c_i', 0) \\ & + \sum_{i \in \mathrm{L}} h_i^- \max(c_i' - d_i, 0) \\ & - \left[\sum_{i \in \Gamma \backslash \mathrm{L}} h_i (a_{\sigma(i)} - c_i) + \sum_{i \in \mathrm{L}} h_i \max(d_i - c_i, 0) \right. \\ & \left. + \sum_{i \in \mathrm{L}} h_i^- \max(c_i - d_i, 0) \right] \end{aligned}$$

$$= \Delta \cdot \left[\sum_{i \in \Gamma \backslash \mathrm{L}} h_i \cdot (I(\sigma(i)) - I(i)) + \sum_{i \in \mathrm{L}} h_i \cdot I(i) \cdot 1\{d_i > c_i\} \right.$$

$$\left. + \sum_{i \in L} h_i^- \cdot I(i) \cdot 1\{d_i \leqslant c_i\} \right].$$

The last equation comes from $a_i' - a_i = \Delta \cdot I(i)$ and $c_i' - c_i = \Delta \cdot I(i)$. Divide by Δ on both sides of the last equation and let $\Delta \to 0$, the assertion is proved. □

From the proof of Theorem 1, $(V(s_j + \Delta, \omega) - V(s_j, \omega))/\Delta = \partial V(\mathbf{s}, \omega)/\partial s_j$ holds for any sufficiently small positive Δ. In the remainder of this section, the aim is to show that the gradient estimator given in Theorem 1 is unbiased (Song and Sun 1998, Song *et al.* 1999). The following two Lemmas are established first.

Lemma 1: Let Ω_i be a set of sample processes in which there exists at least two elements on the RHS of equation (1) that take the same values, i.e.

$$\begin{aligned} \Omega_i = \{\omega \mid c_{\varphi(i)} = s_i\} \cup \{\omega \mid \exists k \in \mathrm{B}(i) \text{ s.t. } c_k = s_i\} \cup \\ \{\omega \mid \exists k \in \mathrm{B}(i) \,\text{and}\, k \neq \varphi(i) \text{ s.t. } c_{\varphi(i)} = c_k\} \cup \\ \{\omega \mid \exists k, l \in \mathrm{B}(i) \,\text{and}\, k \neq l \,\text{s.t. } c_k = c_l\} \end{aligned} \qquad (6)$$

Let Ω_0 be the union of all $\Omega_i, i \in \Gamma$ such as $\Omega_0 = \cup_{i \in \Gamma} \Omega_i$ Then, $\mathrm{Prob}(\Omega_0) = 0$.

Proof: Since $x_i (i \in \Gamma)$ is an independent continuous random variable (r.v.) and $s_i (i \in \Gamma)$ is a constant, it is easy to show that $a_i (i \in \Gamma)$ is a truncated continuous r.v. and $c_i (i \in \Gamma)$ is a continuous random variable. Therefore, $\mathrm{Prob}\{\omega \mid c_{\varphi(i)} = s_i\} = 0$ and $\mathrm{Prob}\{\omega \mid \exists k \in \mathrm{B}(i) \text{ s.t. } c_k = s_i\} = 0$. The last term of RHS of (6) can be rewritten as:

$$\{\omega \mid \exists k, l \in \mathrm{B}(i) \text{ and } k \neq l \text{ s.t. } c_k = c_l\} = \{\omega \mid \exists k, l \in \mathrm{B}(i) \text{ and } k \neq l \text{ s.t. } (a_k - a_l) + (x_k - x_l) = 0\}.$$

Note that $(a_k - a_l)$ and $(x_k - x_l)$ are independent, $(a_k - a_l)$ is a truncated continuous r.v. and $(x_k - x_l)$ is a continuous r.v., it follows that $(a_k - a_l) + (x_k - x_l)$ is another continuous r.v. Hence, $\mathrm{Prob}\{\omega \mid \exists k, l \in \mathrm{B}(i)$ and $k \neq l$ s.t. $c_k = c_l\} = 0$. Similarly, $\mathrm{Prob}\{\omega \mid \exists k \in \mathrm{B}(i)$ and $k \neq \varphi(i)$ s.t. $c_{\varphi(i)} = c_k\} = 0$. Thus, $\mathrm{Prob}(\Omega_i) = 0$ and $\mathrm{Prob}(\Omega_0) \leqslant \Sigma_{i \in \Gamma} \mathrm{Prob}(\Omega_i) = 0$. □

Lemma 2: Let Δ be a positive real number and define:

$$\begin{aligned} \Omega_i(\Delta) = \{\omega \mid |c_{\varphi(i)} - s_i| \leqslant \Delta\} \cup \{\omega \mid \exists k \in \mathrm{B}(i) \text{ s.t. } |c_k - s_i| \leqslant \Delta\} \cup \\ \{\omega \mid \exists k \in \mathrm{B}(i) \text{ and } k \neq \varphi(i) \text{ s.t. } |c_{\varphi(t)} - c_k| \leqslant \Delta\} \cup \\ \{\omega \mid \exists\, k, l \in \mathrm{B}(i) \text{ and } k \neq l \text{ s.t. } |c_k - c_l| \leqslant \Delta\}. \end{aligned} \qquad (7)$$

Let $(\Omega_\Delta = \cup_{i \in \Gamma} \Omega_i(\Delta)$. Then, $\lim_{\Delta \to 0} \mathrm{Prob}(\Omega_\Delta) = 0$.

Proof: The proof follows directly as a result of Lemma 1. □

Lemma 3: Let Δ be a positive real number and Ω_Δ be defined in Lemma 2. If $\omega \notin \Omega_\Delta$, then

$$(V(s_j+\Delta,\omega)-V(s_j,\omega))/\Delta=\partial V(\mathbf{s},\omega)/\partial s_j$$

holds for $j\in\Gamma$, where the RHS of above equation is given in Theorem 1.

Proof: Lemma 1 states that if $\omega\notin\Omega_0$, then for any $i\in\Gamma$ there exists only one element on the RHS of (1), which is equal to a_i. From Lemma 2, if $\omega\notin\Omega_\Delta$ then a perturbation Δ on any $s_i(i\in\Gamma)$ does not change the order in which the events occur in the whole path. That means, NP and PP have the exactly same event order. The assertion is true. □

Now the unbiasedness of the gradient estimator can be established.

Theorem 2—Unbiasedness of gradient estimator: Let $\partial V(\mathbf{s},\omega)/\partial s_j$ be given in Theorem 1, then

$$\partial J(\mathbf{s})/\partial s_j=\mathrm{E}\partial V(\mathbf{s},\omega)/\partial s_j \text{ and } \mathrm{E}|\partial V(\mathbf{s},\omega)/\partial s_j|<\infty,\text{for any } j\in\Gamma.$$

Proof: As above, let $V(s_j,\omega)$ and $V(s_j+\Delta,\omega)$ denote the sample cost functions for NP and PP respectively. Consider

$$\begin{aligned}(\mathrm{E}V(s_j+\Delta,\omega)-\mathrm{E}V(s_j,\omega))/\Delta&=\mathrm{E}(V(s_j+\Delta,\omega)-V(s_j,\omega))/\Delta\\&=\mathrm{E}\{(V(s_j+\Delta,\omega)-V(s_j,\omega))/\Delta\,|\,\Omega\backslash\Omega_\Delta\}\\&\quad+\mathrm{E}\{(V(s_j+\Delta,\omega)-V(s_j,\omega)/\Delta\,|\,\Omega_\Delta\}\\&=\mathrm{E}\{\partial V(\mathbf{s},\omega)\partial s_j\,|\,\Omega\backslash\Omega_\Delta\}\\&\quad+\mathrm{E}\{(V(s_j+\Delta,\omega)-V(s_j,\omega))/\Delta\,|\,\Omega_\Delta\}.\end{aligned}\tag{8}$$

The last equation is from Lemma 3. By the definition of sample cost function, it is easy to show that for any $\omega\in\Omega_\Delta$

$$|(V(s_j+\Delta,\omega)-V(s_j,\omega))/\Delta|\leqslant C\cdot\sum_{i\in\Gamma}\bar{h},\tag{9}$$

where C is a constant and $\bar{h}=\max\{h_i,h_j^-,i\in\Gamma,j\in \mathrm{L}\}$. From (9) and Lemma 2, it follows

$$\lim_{\Delta\to0}\mathrm{E}\{(V(s_j+\Delta,\omega)-V(s_j,\omega))/\Delta\,|\,\omega\in\Omega_\Delta\}=0.\tag{10}$$

Let Δ tend to zero on both sides of (8), from (10) and Lemma 2:

$$\partial J(\mathbf{s})/\partial s_j=\mathrm{E}\partial V(\mathbf{s},\omega)/\partial s_j.$$

The second part of Theorem 2 can be directly derived from Theorem 1. □

4. Stochastic approximation algorithm

This section describes the stochastic approximation algorithm to find the optimal planned job release times by minimizing the cost function J(s). The general form of SA is as follows:

$$\theta_{n+1}=\theta_n+\gamma_n\cdot\nabla J_n,\tag{11}$$

where θ_n is the parameter vector at the beginning of iteration n, ΔJ_n is an estimator of $\nabla J(\theta_n)$ (where $\nabla J(\theta_n):=(\partial J(\theta_n)/\partial s_1,\ldots,\partial J(\theta_n)/\partial s_{||\Gamma||})^{\mathrm{T}}$), γ_n is a positive sequence of step sizes which decreases to zero. When ∇J_n is an unbiased estimator of $\nabla J(\theta_n)$, (11) is called a Robbins–Monro (RM) algorithm and when a finite difference estimator is used, it is called a Kiefer-Wolfowitz (KW) algorithm. RM algorithm has faster convergence rates than the KW algorithm.

Since Theorems 1 and 2 provide an unbiased gradient estimator, equation (11) yields a RM algorithm. The details of the stochastic approximation algorithm can be found in Fu and Hu (1997) and Rubinstein (1986). Owing to the complexity of our systems, it is difficult to prove that our optimization algorithm converges to the optimal solution. However, a multitude of numerical examples show that it does lead to an optimal or near-optimal solution.

It should be pointed out that if s_j is too small then $\partial J(\mathbf{s})/\partial s_j$ may be 0 (because it is possible that $a_j > s_j$ holds for all sample processes). Therefore, at each iteration, for all those s_j s.t. $\partial J(\mathbf{s})/\partial s_j = 0$ it is necessary to make sure $s_j \geqslant \max\{s_{\varphi(j)}, s_k, k \in \mathrm{B}(j)\}$.

5. Service level constraint problem

This section addresses another important objective in make-to-order systems, that is, meeting the service level constraints. The service level is defined as the probability of completing a job before its due date. These constraints may be assigned by customer or planner. It is important to meet these constraints when creating a plan. In this way estimates of production progress and delivery performance will be accurate. This is also the main concern of risk analysis.

Assume that all service level constraints are given and the optimal planned job release times are found based on PA and SA. The aim now is to adjust further these planned job release times to meet the service level constraints. Let p_i denote the service level constraint for job i. The following procedure is developed for this purpose.

5.1. *Procedure to adjust planned job release times to meet service level constraints*

Step 1. Set $\mathrm{A}_0 = \varnothing$(empty set), $\mathrm{A}_1 = \Gamma$ and $\mathbf{s} =$ the vector of planned job release times.

Step 2. Run multiple simulation experiments for current $\mathbf{s}$.

Step 3. For any $i \in \mathrm{A}_1$, if $\varphi(i) \in \mathrm{A}_0$ and $\mathrm{B}(i) \subseteq \mathrm{A}_0$, then:

(i) If $i \notin \mathrm{L}$, adjust $s_{\sigma(i)}$ such that the proportion of runs in which job i is completed before $s_{\sigma(i)}$ is not less than p_i and minimize the cost function.

(ii) If $i \in \mathrm{L}$, shift all $\{s_i, i \in \Gamma\}$ such that the proportion of runs in which job i is completed before d_i is not less than p_i and minimize the cost function.

(iii) Set $\mathrm{A}_1 = \mathrm{A}_1 - \{i\}$ and $\mathrm{A}_0 = \mathrm{A}_0 \cup \{i\}$.

Step 4. If $||\mathrm{A}_0|| < ||\Gamma||$, upgrade $\mathbf{s}$ and go to Step 2.

Step 5. Stop.

Here $||.||$ is the length of a set. In Step 3(i), the proportion of runs completing job i before $s_{\sigma(i)}$ means the number of sample processes in which job i is finished before $s_{\sigma(i)}$) divided by the total number of sample processes. Note that in each loop only one service level constraint p_i is considered and adjusting the parameters in the later loops does not affect the service levels in the earlier loops due to the given job precedence constraints. Therefore, at most after total $||\Gamma||$ loops the resultant plan $\{s_i, i \in \Gamma\}$ can meet all service level constraints and has a low expected cost.

6. Numerical examples

Numerical examples are given here to verify the results. First, the perturbation analysis gradient estimator is compared with finite difference estimator. Second, to demonstrate the effectiveness of the PA-based SA algorithm, the present result are compared with a commonly used heuristic method, termed total work on the critical path (TWKCP) (Fry *et al.* 1989, Roman and Valle 1996). According to TWKCP, the planned job release times are determined by:

$$s_i = d_i - k \cdot \mu_i, \text{if } i \in \mathrm{L};$$

$$s_i = s_{\sigma(i)} - k \cdot \mu_i, \text{if } i \in \Gamma \backslash \mathrm{L},$$

where k is a constant and $\mu_i := \mathrm{E}x_i$, which is the mean processing time of job i. If $k = 1$, the TWKCP method is a typical backwards scheduling based on infinity capacity model.

Take a sequence of sample processes, $\omega_1, \omega_2, \ldots, \omega_K$. Define the finite difference by

$$\mathrm{D}J(\mathbf{s})/\mathrm{D}s_i := \frac{1}{K}\sum_{i=1}^{K}(V(s_i + \Delta, \omega_l) - V(s, \omega_l))/\Delta.$$

Denote the average sensitivity estimate from PA technique over K sample processes by

$$\partial\bar{J}(\mathbf{s})/\partial s_i = \frac{1}{K}\sum_{l=1}^{K}(\partial V(\mathbf{s}, \omega_l)/\partial s_j),$$

where $\partial V(\mathbf{s}, \omega_l)/\partial s_j$ is given in Theorem 1. By the strong large number law, one has $\lim_{\Delta\to 0}\lim_{K\to\infty}\mathrm{D}J(\mathbf{s})/\mathrm{D}s_i = \partial J(\mathbf{s})/s_i$ and $\lim_{K\to\infty}\partial\bar{J}(\mathbf{s})/\partial s_i = \mathrm{E}(\partial V(\mathbf{s}, \omega)/\partial s_i)$ almost surely. In the following examples $K = 100$.

Example 1: Consider an assembly system producing one product with product structure pictured in figure 1. It involves 12 jobs and six resources. The job sequence on each resource is given in figure 1. The product due date $d = 180$ days. All processing times are assumed to be normally distributed with means $\mu_{12} = 2.25$, $\mu_{11} = 8.55$, $\mu_{10} = 14$, $\mu_9 = 14$, $\mu_8 = 1.54$, $\mu_7 = 0.41$, $\mu_6 = 16$, $\mu_5 = 11.18$, $\mu_4 = 8.12$, $\mu_3 = 14$, $\mu_2 = 28$, and $\mu_1 = 28$ (days). The standard deviation (SD) of the processing time is assumed to be $\sigma_i = 0.3\,\mu_i$ for $i = 1, 2, 3, 4$ and $\sigma_i = 0.1\,\mu_i$ for other jobs. The holding cost coefficient of a job is assumed to be proportional to the total time spent on this job and on all previous jobs in its branch. For example, $h_{12} = 0.0225$, $h_{11} = 0.0855$, $h_{10} = 0.2480$, $h_9 = 0.3880$, $h_8 = 0.0154$, $h_7 = 0.0041$, $h_6 = 0.1594$, $h_5 = 0.1118$, $h_4 = 0.0812$, $h_3 = 0.3330$, $h_2 = 0.6130$ and $h_1 = 1.4405$. The tardiness penalty coefficient $h_1^- = 2 \cdot h_1 = 2.8809$. Take $\Delta = 0.1$ and 0.01. Then the gradient estimators of the cost function via PA technique and its finite difference estimators are given in table 1. The first column is the job code; the second column is a set of planned job release times which comes from TWKCP with $k = 1.0$; the third and forth columns are the finite difference estimators by pure simulation method with $\Delta = 0.1$ and 0.01 respectively; the fifth column is the gradient estimator by PA technique and the last column is the real service level for each job in the simulation. The average cost over K sample processes equals 58.12.

Table 1 shows that the finite difference gradient estimator is only a little bit different from the PA gradient estimators when $\Delta = 0.1$ and they become almost

i	s_i	$\mathrm{D}J(\mathbf{s})/\mathrm{D}s_i$ $(\Delta = 0.1)$	$\mathrm{D}J(\mathbf{s})/\mathrm{D}s_i$ $(\Delta = 0.01)$	$\partial \bar{J}(\mathbf{s})/\partial s_i$	Service level (%)
12	121.75	0.9542	0.6340	0.6389	49
11	115.45	1.3985	1.3747	1.3747	47
10	124.00	1.0337	0.8479	0.8479	48
9	138.00	−0.0285	−0.0268	−0.0268	34
8	136.46	−0.0842	−0.0842	−0.0842	46
7	137.59	−0.0001	0.0000	0.0000	0
6	138.00	−0.0015	0.0000	0.0000	37
5	98.82	−0.1667	−0.1650	−0.1650	48
4	101.88	−0.1003	−0.0985	−0.0976	57
3	110.00	−0.0394	−0.0364	−0.0364	44
2	124.00	0.0000	0.0000	0.0000	5
1	152.00	−0.0028	−0.0028	−0.0028	10
			$\bar{J}(\mathbf{s}) = 58.12$		

Table 1. PA gradient estimators and finite difference estimators for TWKCP with $k = 1.0$.

the same when $\Delta = 0.01$. This verifies the results given in Theorems 1 and 2. In addition, from this gradient information it is clear that the production plan **s** can be improved because some partial derivatives are big. The results of stochastic approximation using PA gradient estimator are given in table 2, where $\{s_i\}$ is a set of near-optimal planned job release times since all partial derivatives are close to zero. The average cost over K sample processes equals 23.97.

In table 2, the partial derivatives of the cost function with respect to planned job release times are close to zero and the average cost decreases to 23.97 from 58.12. One reason for this great improvement is that backwards scheduling based on infinity capacity model is not appropriate for our system. TWKCP method can be improved by carefully selecting the constant k. Table 3 compares the present results with that of TWKCP with varying parameter k. WIP holding cost (WIPC), product earliness cost (PEC), product tardiness cost (PTC) and average total cost $\bar{J}(\mathrm{s})$ are given.

i	s_i	$\partial \bar{J}(\mathbf{s})/\partial s_i$	Service level (%)
12	95.95	0.0000	0
11	90.35	−0.0110	0
10	96.56	0.0000	100
9	129.77	−0.0427	0
8	127.27	0.0843	0
7	127.87	0.0000	0
6	128.48	0.0000	0
5	87.76	0.0457	0
4	82.55	−0.0042	98
3	95.07	0.0000	0
2	103.03	0.0000	12
1	133.32	0.0000	65
		$\bar{J}(\mathbf{s}) = 23.97$	

Table 2. Optimal solution using a PA-based SA procedure.

	TWKCP							
Cost	$k=1.0$	$k=1.1$	$k=1.2$	$k=1.3$	$k=1.4$	$k=1.5$	$k=1.6$	PA + SA
WIPC	12.85	13.34	14.09	15.05	16.43	18.19	20.22	10.11
PEC	0.88	1.94	3.76	6.90	10.86	15.75	21.08	8.60
PTC	44.38	30.02	18.32	9.97	4.33	1.55	0.33	5.26
$\bar{J}(\mathbf{s})$	58.12	45.31	36.17	31.91	31.61	35.49	41.64	23.97

Table 3. WIP holding cost, product earliness and tardiness costs for TWKCP with different k

For TWKCP, the WIP holding cost and product earliness cost increase as k increases; the product tardiness cost decreases as k increases; and the total expected cost reaches its lowest value around $k = 1.4$. However, whatever the constant k is, there is a significant improvement in the total cost by using PA and SA.

Example 2: In Example 1 we did not consider service level constraint, which is why the real service levels under designed plans are very different. Some of them are as high as 100%, but some are as low as 0%. Assume that 50% service level should be achieved for each job. By using the procedure given in Section 5, one can adjust the production plan to meet all those service level constraints. Table 4 lists the results by adjusting the plan given in table 2 (from PA plus SA). The average cost over K sample processes increases to 24.41.

Table 4 shows that the resulted plan does meet the service level constraints. However, the plan has a slight higher cost than the result in table 2. In table 4 some service levels still reach 100% because the corresponding jobs have a relative big safety lead time (which is defined as the excess of the planed lead-time above average lead-time). For example, job 10's planned start time is 100.28, its due date is 128.81 and its processing time follows Normal(14, 1.4), so its safety lead time = 14.53. It should be pointed out that the safety lead-time is necessary to deal with resource constraints and to meet service level constraints.

i	s_i	$\partial\bar{J}(\mathbf{s})/\partial s_i$	Service level (%)
12	94.98	0.0000	50
11	89.38	−0.0402	83
10	100.28	0.1443	100
9	128.81	0.0676	85
8	126.30	−0.0504	100
7	126.90	0.0000	50
6	128.26	−0.0324	93
5	86.80	−0.1998	50
4	81.58	0.0126	100
3	98.02	0.0817	50
2	111.93	−0.0112	50
1	144.34	0.2713	66
		$\bar{J}(\mathbf{s}) = 24.41$	

Table 4. Adjust planned job release times tomeet servbice level constraints.

7. Conclusion

This paper has developed an effective algorithm to obtain an unbiased gradient estimator of the cost function with respect to planned job release times. By Robbins-Monro stochastic approximation approach, the optimal or near-optimal planned job release times can be found. Compared with TWKCP method, a significant improvement is achieved for any allowance constant k. The new method is promising due to its capability to deal with complex make-to-order systems and various stochastic situations. A procedure to adjust planned job release times to meet service level constraints is proposed. It provides the production designer a good estimate of the production progress and the risk of a job to be late. Technical results are confirmed by simulation.

References

Bremaud, P., Malhame, R. P. and Massoulie, L., 1997, A manufacturing system with general stationary failure process: stability and IPA of hedging control policies. *IEEE Transactions on Automatic Control*, **42**, 155–170.

Buzacott, J. A. and Shanthikumar, J. G., 1993, *Stochastic Models of Manufacturing Systems* (Englewood Cliffs: Prentice Hall).

Cassandras, C. G., 1993, *Discrete Event Systems: Modeling and Performance Analysis* (Homewood: Irwin).

Donohue, K. L. and Spearman, M. L., 1993, Improving the design of stochastic production lines: an approach using perturbation analysis. *International Journal of Production Research*, **31**, 2789–2806.

Fry, T. D., Oliff, M. D., Minor, E. D. and Leong, G. K., 1989, The effects of product structure and sequencing rule on assembly shop performance, *International Journal of Production Research*, **27**, 671–686.

Fu, M. and Hu, J. Q., 1997, *Conditional Monte Carlo: Gradient Estimation and Optimisation Applications* (Boston: Kluwer).

Glasserman, P., 1991, *Gradient Estimation via Perturbation Analysis* (Norwell: Kluwer).

Gong, L. G., Kok, T. and Ding J., 1994, Optimal lead-times planning in a serial production system, *Management Science*, **40**, 629–632.

Hasan, C. N. and Spearman, M. L., 1999, Optimal material release times in stochastic production environments. *International Journal of Production Research*, **37**, 1201–1216.

Ho, Y. C. and Cao, X. R., 1991, *Perturbation Analysis of Discrete Event Dynamic Systems* (Norwell: Kluwer).

Roman, D. B. and Valle, A. G., 1996, Dynamic assignment of due-dates in an assembly shop based in simulation. *International Journal of Production Research*, **34**, 1539–1554.

Rubinstein, R. Y., 1986, *Monte Carlo Optimisation, Simulation and Sensitivity of Queuing Networks* (New York: Wiley).

Song, D. P., Hicks, C. and Earl, C. F., 1999, Planned lead-times design in uncertain manufacturing systems via perturbation analysis. *European Journal of Operational Research*, submitted for publication.

Song, D. P. and Sun, Y. X., 1998, Gradient estimate for parameter design of threshold controllers in a failure-prone production line, *International Journal of Systems Science*, **29**, 21–32.

Tang, Q. Y. and Boukas, E. K., 1999, Adaptive control for manufacturing systems using infinitesimal perturbation analysis. *IEEE Transactions on Automatic Control*, **44**, 1719–1725.

Yan, H. M., Yin, G. and Lou, S. X. C., 1994, Using stochastic optimisation to determine threshold values for the control of unreliable manufacturing systems. *Journal of Optimisation Theory and Applications*, **83**, 511–539.

Yano, C. A., 1987a, Setting planned lead-times in serial production systems with tardiness costs, *Management Science*, **33**, 95–106.

Yano, C. A., 1987b, Stochastic lead-times in two-level assembly systems, *IIE Transactions*, **19**, 371–378.

Yano, C. A., 1987c, Stochastic lead-times in two-level distribution type networks, *Naval Research Logistics*, **34**, 831–843.