Chapter 5 – Job Release by Starvation Avoidance

Some features of wafer fabrication are not found in most other job shops. Because each layer of the wafer requires exposure of a photoresistive material through a mask (the process is known as photolithography) each batch of wafers makes many visits to the photolithography work station before it is completed. Since this equipment is very expensive, this work station is often the bottleneck. The special feature of work flow in wafer fabrication is that it is re-entrant at a bottleneck work station. A scheduling policy should focus on the queue of work at the bottleneck because it will on average be the biggest queue in the factory and furthermore, whenever the bottleneck work station is forced to become idle due to lack of work, that lost time represents an irretreviable loss of final output. A desirable scheduling policy is one that allows high utilization rates of bottleneck machines while maintaining low levels of work-in-process. At first, these objectives may appear contradictory since increasing bottleneck utilization will cause work-in-process and mean delay also to increase. The rate of increase, however, will depend on the particular scheduling policy in use, since, as we will see in Chapter 6, some scheduling policies are more delay-sensitive to changes in throughput rate than others. We seek a policy with less sensitivity, i.e. a policy that outperforms others in terms of the mean delay/mean throughput tradeoff.

The literature of job shop scheduling makes little, if any, reference to job release control. Most research has assumed job release to be random, commonly a Poisson process. Recently, some indication has surfaced regarding the benefits of release control in the context of job shops with unreliable machines. Burman et al. [Bur86a] observe that there is much to be gained by reducing the variability of inter-arrival times of jobs. Wein [Wei86a] suggests that job release control plays a much more significant role than dispatching in terms of reducing mean job delay. In a paper reviewing performance analysis modeling for manufacturing systems, Fredericks

[Fre86a] suggests that controlling the number of lots in the pipeline from start to the bottleneck resource can lead to a reduction in work-in-process when compared to the case where no control takes place. Fredericks, however, gives no details of how this could be accomplished.

In this Chapter we present a class of job release control mechanisms that we call Starvation Avoidance (SA), designed to maximize utilization of critical bottleneck equipment, while at the same time controlling the growth of work-in-process.

We make the following assumptions about the production system.

- All random variables (e.g. time between machine failures, time to repair and fraction of rework) are stationary.
- The desired output rate is constant (at least for intervals long compared to mean flow time).
- A single work station is the unique bottleneck for the shop and machine time is the limiting resource. In other words, there is a single work station that has the minimum expected idle time. For work station j, the proportion of idle time is given by

$$I_j = 1 - F \times \frac{W_j}{N_j A_j} \tag{1}$$

where F is the average flow rate of new work into the shop (wafers per hour), W_j is the work load (machine hours) at station j per wafer, N_j is the number of machines at station j and

$$A_j = \frac{(MTBF_j - MTTR_j)}{MTBF_j} \tag{2}$$

is average machine availability, where $MTTR_j$ and $MTBF_j$ are mean time to repair and mean time between failures of machine j. We argue that this assumption is not restrictive because an exactly balanced shop is very difficult to achieve in practice. Machines can only be purchased in integer

quantities from among a very limited set of production rates. Average machine reliability changes slowly over time as do workloads, so even if the shop were designed to be exactly balanced, it would not remain balanced. The concept of a bottleneck work station is the key to the analysis. Idle time at the bottleneck represents a permanent loss of factory output, and output cannot be increasesd by trying to keep non-bottleneck stations busy. We later allow a second station to have proportion of idle time close to that of the bottleneck station (we call it a second bottleneck).

The central idea behind SA is simple: To reduce inventory, do not start new work. The difficulty with that idea is that eventually the bottleneck work station starves and no finished work leaves. Hence, we are lead to the *starvation avoidance* rule: Start new work just in time to avoid idling the bottleneck work station due to lack of work. An analogy can be made between work content of the queue at the bottleneck station (in the context of job shop scheduling) and inventory (in the context of inventory control). In inventory control the main objective is to optimize the tradeoff between inventory holding costs and the costs of stock outs. If demand and delivery lead time (the time lag between order and delivery) are deterministic this task is trivial. However, in practice both demand and supply (delivery lead time) are random. Protection from the uncertainty comes from the concept of safety stock. Whenever on hand inventory plus the quantity on order but not yet delivered falls below the safety stock level, an order is placed to bring the total up to the safety stock level. Provided the safety stock level is big enough, the next order will arrive in time to avoid running out of stock, with sufficiently high probability (see Fig. 5.1).

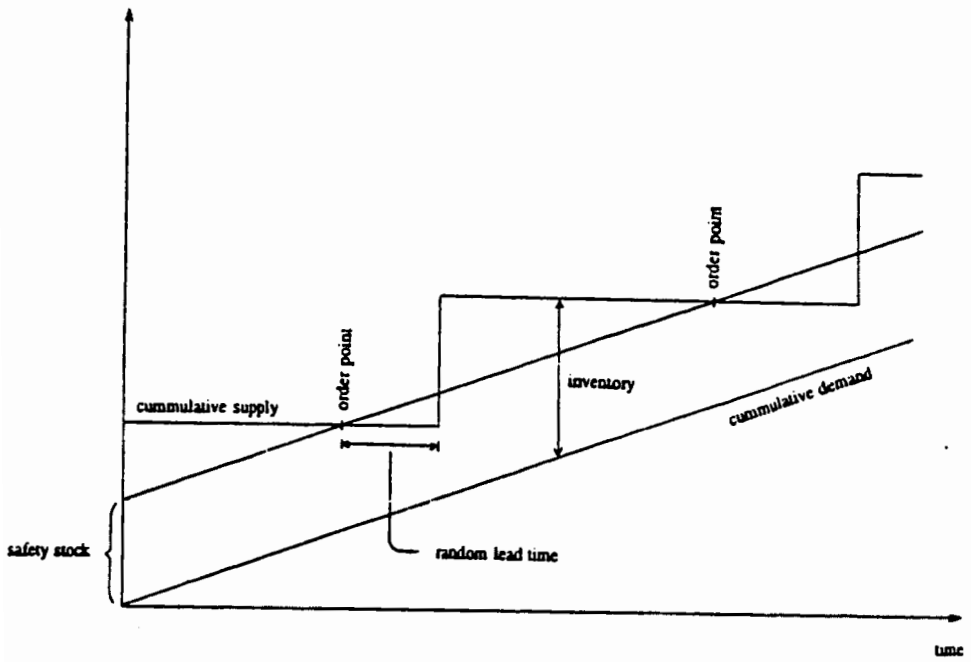

Fig. 5.1 – Safety Stock Notion in Inventory Control

In a job shop, the variability of physical inventory at the bottleneck results from the interaction of random demand (induced by the failure and repair at the bottleneck work station), the uncertain lead time required for raw wafers newly injected into the system to reach the bottleneck, and the arrival of work already in process from other work stations (a phenomenon not present in standard inventory models). We define the concept of *virtual inventory* to be the work content (measured in work-hours at the bottleneck) of all jobs either at the bottleneck work station or expected to arrive at the bottleneck within a given lead time. We take as lead time for replenishment the expected time required for a job to arrive at the bottleneck for the first time, once it has entered the shop. We account for the current repair state of machines at the bottleneck work station by including an estimate of the time to repair any failed machines as part of the virtual inventory. As in

inventory control, whenever the virtual inventory falls below a given (safety stock) level, a new job is released (see Fig. 5.2).

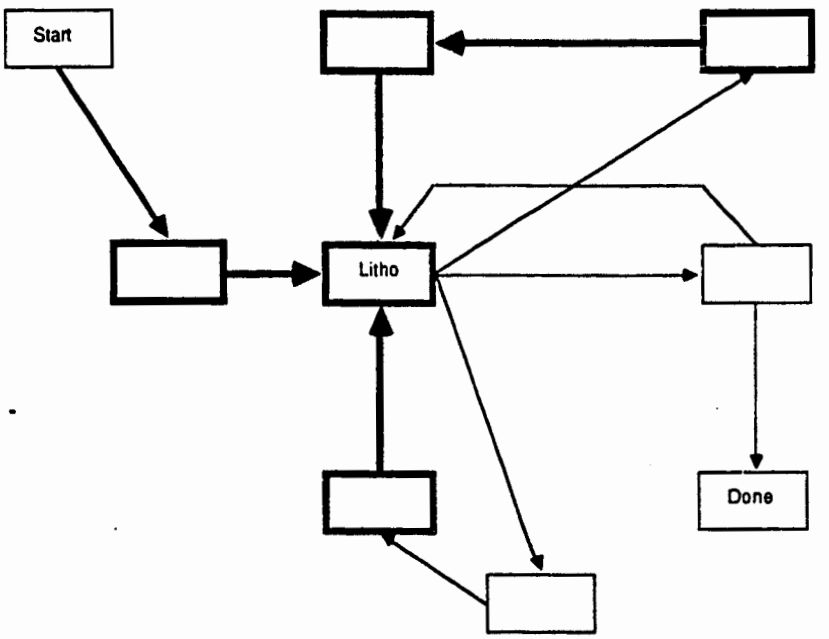

Fig. 5.2 – Virtual Inventory at Photolithography

Effective flow control requires a sufficiently large inventory of raw material so that new work can be introduced whenever desired. While such policies shift inventory from work-awaiting-process (WAP) to raw material, it is much less expensive to hold inventory in that form. The raw wafers can be made into any product, so the risk of obsolescence due to demand shifts or engineering changes is lower. They are not as subject to contamination and yield loss.

The objective is to make the new job arrive at the bottleneck *just in time* to avoid equipment starvation there. The safety stock level is the control parameter for the system. Increasing this number will increase average inventory levels but will reduce the probability of starving the bottleneck due to lack of work and hence increase average output of finished wafers. By varying the safety stock level, one can

generate a tradeoff curve, $D(t)$, of delay versus throughput.

We now formalize the ideas described above. We begin by describing a release strategy for the one product, one bottleneck work station case. We term this strategy SA(1,1). We describe an accompanying dispatching scheme, designed to boost the performance of SA(1,1). We then generalize this release policy to three other cases, namely: SA(N,1), the multi-product, one bottleneck work station case; SA(1,2), the one product, two bottleneck work station case; and finally SA(N,2), the multi-product, two bottleneck work station case.

5.1 — One Product One Bottleneck Station Case — SA(1,1)

Consider a job shop with one product and one bottleneck work station. Denote the bottleneck station by B. Station B has m unreliable machines with mean times to repair $r_1, r_2, \ldots, r_m$, respectively. Jobs are assigned a deterministic recipe of p steps

$$\rho = s_1, s_2, \ldots, s_p. \tag{5.1}$$

Let η_i be the number of jobs currently at step i (either queued or in process). The i-th step, s_i, has an associated work station w_i and duration d_i. Let i_0 be the step number corresponding to the first visit to station B,

$$i_0 = \min (i \mid w_i = B). \tag{5.2}$$

Let S_B be the set of bottleneck work station steps,

$$S_B = (s_i \mid w_i = B), \tag{5.3}$$

and let F be the set of steps prior to the first visit to station B,

$$F = (s_i \mid i = 1,\ldots,i_0-1). \tag{5.4}$$

This Page Left Blank

Define C to be the total processing time from start (step s_1) to the first visit to station B, excluding step s_{i_0},

$$C = \sum_{i=1}^{i_0-1} d_i \tag{5.5}$$

Let n_i be the step number of the next visit to station B given the job is presently at step s_i. Furthermore, let P be the set of steps such that total processing of that step plus future steps prior to the next visit to station B (excluding processing at station B) is less than βC, where $\beta > 0$,

$$P = (s_i \mid \sum_{j=i}^{n_i-1} d_j < \beta C) \tag{5.6}$$

Let $Q = F \cup P \cup S_B$ be the set of critical steps, *i.e.* steps that are within a critical time factor of the bottleneck station or steps in the pipeline from start to the first visit to B. Let $D(B)$ be the index set of machines under repair at the bottleneck work station. Define the expected total repair time of equipment at the bottleneck to be

$$R = \sum_{j \in D(B)} r_j. \tag{5.7}$$

Let W be the total work content at the next visit to the bottleneck station of jobs at steps in set Q plus the total expected repair time of equipment down at the bottleneck, R,

$$W = \frac{\left\{ R + \sum_{j \in Q} \eta_j d_{n_j} \right\}}{m}. \tag{5.8}$$

W can be interpreted as the expected time to starve the bottleneck. If W falls below a critical value $\alpha\, C$, where $\alpha > 0$ is a safety factor, there is danger that the bottleneck will starve. Release strategy SA(1,1) triggers a job start at this time, to avoid starvation of the bottleneck.

5.2 – The SA Dispatching Rule

The main objective of the SA release strategy described in Section 5.1 is the minimization of idle time at bottleneck station machines. In the description of the release policy we make no mention of dispatching policies for lots at individual stations. It is likely that local dispatching may affect SA. For example, a dispatching policy that makes an intentional effort to delay processing of jobs in the initial pipeline steps, $s_1, s_2, \ldots, s_{i_0-1}$, (such as the SRPT rule) will make a negative contribution to SA release when there is danger of bottleneck starvation. An ideal dispatching scheme would aid SA release in achieving its objective, by giving priority to jobs headed for the bottleneck station, when that station is in danger of starving, and giving priority to other lots when there is no imminent danger of starvation. We present such a rule in this Section.

The main characteristic of this SA-booster dispatching rule is its dynamic behavior. The rule combines two simple rules by means of weights and dynamically changes the weights according to the state of the system. If the bottleneck is in no danger of starvation the dispatching rule gives more weight to the shortest remaining processing time rule (SRPT) while if there is imminent danger, more weight is given to a rule (SA^+) that gives high priority to lots that are headed for the bottleneck station.

Let p_{SRPT} and $\bar{p}_{SRPT}$ be the unnormalized and normalized priority functions of SRPT, respectively. Let p_{SA^+} and $\bar{p}_{SA^+}$ the unnormalized and normalized priority functions of SA^+, respectively. Here, both normalizations are assumed to be by max-

imum value, *i.e.* the normalized priority functions are obtained by dividing the unnormalized values by the maximum absolute value the priority functions can return. Clearly the range of these functions is the real interval [0,1].

The reader can refer to Table 4.1 for a definition of the SRPT priority function. Given a lot is at its i-th process step, the SA^+ priority function is defined by the ratio of the duration of the next visit to the bottleneck to an estimate of the time until the next bottleneck visit:

$$p_{SA^+} = \frac{d_{n_i}}{\sum_{j=i}^{n_i-1} d_j} \tag{5.9}$$

If there is no future visit to the bottleneck p_{SA^+} is defined to be 0. Since we require priority functions to be decreasing, we use the complement of $\bar{p}_{SA^+}$, *i.e.* $1 - \bar{p}_{SA^+}$, as our normalized priority function. This function ($\overline{SA}^+$) assigns high priority to jobs that are close to the bottleneck station and/or that contribute a large amount of work content to the station.

The composite SA dispatching rule is a weighted mix of SRPT and $\overline{SA}^+$. Its normalized priority function is given by:

$$\bar{p}_{SA} = (1 - \gamma)\,\bar{p}_{SRPT} + \gamma\,(1 - \bar{p}_{SA^+}), \tag{5.10}$$

where $0 \le \gamma \le 1$ is the dynamic weight.

We require the dynamic weight γ to be close to 1 when there is imminent danger of bottleneck starvation (so that $\overline{SA}^+$ will dominate dispatching) and to approach 0 when the bottleneck is in not immediate danger of starving (forcing SRPT to be used to dispatch). This can be achieved in the following way.

Define τ to be the critical work content (a constant parameter) as defined in Section 5.1,

$$\tau = \alpha C, \tag{5.11}$$

and let W be the total work content at the next visit to the bottleneck station of jobs at steps in set Q plus the total expected repair time of equipment in *down* state at the bottleneck, as defined in Section 5.1. By assuming that an infinite supply of new jobs is available for release we can guarantee that the SA release strategy will never allow W to become null.

The dynamic convex weight is defined for all values of $W > 0$, as

$$\gamma = \frac{\left\{\frac{1}{1 + e^{-\tau/W}} - \frac{1}{2}\right\}}{\frac{1}{2}} \tag{5.12}$$

Definition 5.12 produces a dynamic weight function having the desired properties, *i.e.* $\gamma \to 0$ if $W \gg \tau$ and $\gamma \to 1$ if $W \ll \tau$ (Fig. 5.3 illustrates the dynamic weight function for $\tau = 10$).

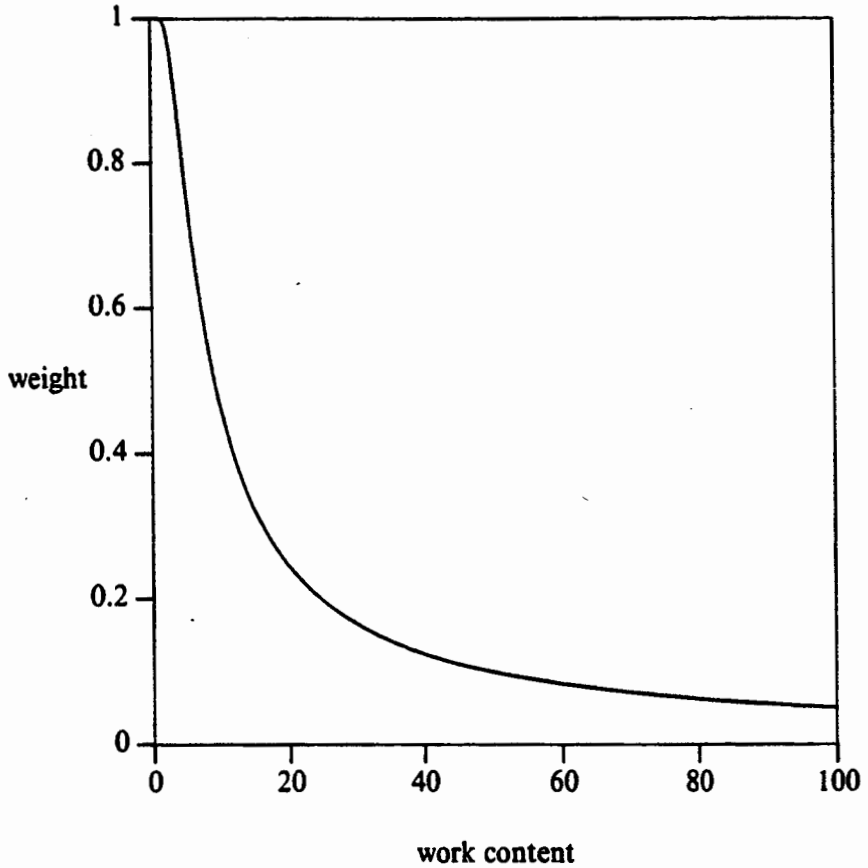

Fig. 5.3 - Dynamic Weight γ

5.3 – Multi Product One Bottleneck Station Case – SA(N,1)

We now consider the case where there is one bottleneck work station (denoted B) and N products. The definitions in this Section are the N-product generalizations of the definitions of Section 5.1. Jobs of product type j are assigned a deterministic recipe of p_j steps

$$p_j = s_{j1}, s_{j2}, \ldots, s_{jp_j}. \tag{5.13}$$

Station B has m unreliable machines with mean times to repair r_1, r_2, ...,r_m, respectively. Let η_{ji} be the number of type j jobs currently at step i (either queued or in

process) of process recipe j. The i-th step of a type j product recipe, s_{ji}, has an associated work station w_{ji} and duration d_{ji}. Let i_{j0} be the step number corresponding to the first visit of a type j job to station B,

$$i_{j0} = \min (i \mid w_{ji} = B), j = 1,..., N. \tag{5.14}$$

Let $S_B(j)$ be the set of bottleneck work station steps for type j jobs,

$$S_B(j) = (s_{ji} \mid w_{ji} = B), j = 1,..., N, \tag{5.15}$$

and let F_j be the set of type j steps prior to the first visit of a type j job to station B,

$$F_j = (s_{ji} \mid i = 1,...,i_{j0}-1), j = 1,..., N. \tag{5.16}$$

Define C_j to be the total processing time from the start of a type j job (step s_{j1}) to its first visit to station B, excluding step $s_{ji_{j0}}$,

$$C_j = \sum_{i=1}^{i_{j0}-1} d_{ji}, j = 1,..., N, \tag{5.17}$$

and let

$$C_{\max} = \max (C_j \mid j = 1,..., N). \tag{5.18}$$

Let n_{ji} be the step number of the next visit of a type j job to station B given the job is presently at step s_{ji} (i-th step of type j product recipe). Furthermore, let P_j be the set of type j steps such that total processing of that step plus all other type j steps prior to the next visit of a type j job to station B (excluding processing at station B) is less than βC_j, where $\beta > 0$,

$$P_j = (s_{ji} \mid \sum_{k=i}^{n_j-1} d_{jk} < \beta\, C_j),\ j = 1,\ldots, N. \tag{5.19}$$

Let

$$Q_j = F_j \cup P_j \cup S_B(j),\ j = 1,\ldots, N, \tag{5.20}$$

be the set of critical steps for type j jobs, *i.e.* steps that are within a critical time factor of the bottleneck station or steps in the pipeline from start to the first visit to B.

As before, let $D(B)$ be the index set of machines in *down* state at the bottleneck work station. Define the expected total repair time of equipment at the bottleneck to be given by equation 5.7.

Let W be the sum over all product types j of the total j type work content at the next visit to the bottleneck station of type j jobs at steps in set Q_j plus the total expected repair time of equipment in *down* state at the bottleneck, R,

$$W = \frac{\left\{ R + \sum_{j=1}^{N} \left(\sum_{k \in Q_j} \eta_{jk} d_{jn_k} \right) \right\}}{m}. \tag{5.21}$$

If W falls below a critical value $\alpha\, C_{\max}$, where $\alpha > 0$ is a safety factor, there is danger that the bottleneck will starve. Release strategy SA(N,1) triggers a job start at this time, to avoid starvation of the bottleneck. The job type with greatest negative deviation from its target mix is selected.

Dispatching policy *SA* can be used with release strategy SA(N,1).

5.4 – One Product Two Bottleneck Station Case – SA(1,2)

We now consider a job shop with one product but two bottleneck work stations. The notation of this Section is similar to that of Section 5.1, where lot starts were

triggered when the virtual work content at the bottleneck fell below a given value. Here, lots will be started when the virtual queue at either of the two bottleneck stations falls below given values.

Denote the bottleneck stations by B_1 and B_2. Station B_1 (B_2) has m_1 (m_2) unreliable machines with mean times to repair $r_{11}, r_{12}, \ldots, r_{1m_1}$ ($r_{21}, r_{22}, \ldots, r_{2m_2}$). Jobs are assigned a deterministic recipe of p steps

$$\rho = s_1, s_2, \ldots, s_p. \tag{5.22}$$

We again assume that η_i is the number of jobs currently at step i (either queued or in process), and that the i-th step of the recipe is carried out at work station w_i and requires d_i units of processing time. Let step i_{01} (i_{02}) be the step number corresponding to the first visit to station B_1 (B_2).

$$i_{01} = \min (i \mid w_i = B_1), \tag{5.23}$$

$$i_{02} = \min (i \mid w_i = B_2). \tag{5.24}$$

Let S_{B_1} (S_{B_2}) be the set of steps carried out at station B_1 (B_2),

$$S_{B_1} = (s_i \mid w_i = B_1), \tag{5.25}$$

$$S_{B_2} = (s_i \mid w_i = B_2), \tag{5.26}$$

and let F_1 (F_2) be the set of steps prior to the first visit to station B_1 (B_2)

$$F_1 = (s_i \mid i = 1,\ldots,i_{01}-1), \tag{5.27}$$

$$F_2 = (s_i \mid i = 1,\ldots,i_{02}-1). \tag{5.28}$$

Define C_1 (C_2) to be the total processing time from start (step s_1) to the first visit to station B_1 (B_2). We exclude step $s_{i_{01}}$ ($s_{i_{02}}$) from the computation of C_1 (C_2),

$$C_1 = \sum_{i=1}^{i_{01}-1} d_i \tag{5.29}$$

$$C_2 = \sum_{i=1}^{i_{02}-1} d_i \tag{5.30}$$

Let n_{i1} (n_{i2}) be the step number of the next visit of a job to station B_1 (B_2) given the job is presently at step s_i of its process recipe. Furthermore, define P_1 (P_2) to be the set of steps such that total processing of that step and future steps prior to the next visit to station B_1 (B_2) (excluding processing at the bottleneck) is less than $\beta_1 C_1$ ($\beta_2 C_2$) where $\beta_1 > 0$ ($\beta_2 > 0$),

$$P_1 = (s_i \mid \sum_{j=i}^{n_{i1}-1} d_j < \beta_1 C_1) \tag{5.31}$$

$$P_2 = (s_i \mid \sum_{j=i}^{n_{i2}-1} d_j < \beta_2 C_2). \tag{5.32}$$

Let

$$Q_1 = F_1 \cup P_1 \cup S_{B_1} \tag{5.33}$$

$$Q_2 = F_2 \cup P_2 \cup S_{B_2} \tag{5.34}$$

be the sets of critical steps (for bottlenecks B_1 and B_2, respectively), *i.e.* steps that are within a critical time factor of the bottleneck stations or steps in the pipeline from start to the first visit to the bottlenecks. Let $D(B)$ be the index set of machines under repair at the bottleneck work station B. Define the expected total repair time

of equipment at the bottleneck B_1 (B_2) to be

$$R_1 = \sum_{j \in D(B_1)} r_{j1}, \tag{5.35}$$

$$R_2 = \sum_{j \in D(B_2)} r_{j2}. \tag{5.36}$$

Let W_1 (W_2) be the total work content at the next visit to bottleneck station B_1 (B_2) of jobs at steps in set Q_1 (Q_2) plus the total expected repair time of equipment down at the bottleneck station, R_1 (R_2),

$$W_1 = \frac{\left\{ R_1 + \sum_{j \in Q_1} \eta_j d_{n_j 1} \right\}}{m_1}, \tag{5.37}$$

$$W_2 = \frac{\left\{ R_2 + \sum_{j \in Q_2} \eta_j d_{n_j 1} \right\}}{m_2}. \tag{5.38}$$

If W_1 (W_2) falls below a critical value $\alpha_1 C_1$ ($\alpha_2 C_2$), where $\alpha_1 > 0$ ($\alpha_2 > 0$) is a safety factor, there is danger that bottleneck B_1 (B_2) will starve. Release strategy SA(1,2) triggers a job start at this time, to avoid starvation of that bottleneck.

5.5 – Multi Product Two Bottleneck Station Case – SA(N,2)

In this Section we consider our last generalization of SA, namely when there are two bottleneck work stations (B_1 and B_2) and N products. The definitions in this Section are the N-product generalizations of the definitions of Section 5.4.

Jobs of product type j are assigned a deterministic recipe of p_j steps

$$\rho_j = s_{j1}, s_{j2}, \ldots, s_{jp_j}. \tag{5.39}$$

As before, station B_1 (B_2) has m_1 (m_2) unreliable machines with mean times to repair $r_{11}, r_{12}, ..., r_{1m_1}$ ($r_{21}, r_{22}, ..., r_{2m_2}$). Let η_{ji} be the number of type j jobs currently at step i (either queued or in process) of process recipe j. The i-th step of a type j product recipe, s_{ji}, is carried out at work station w_{ji} and requires d_{ji} units of processing time. Let i_{j1}^0 (i_{j2}^0) be the step number corresponding to the first visit of a type j job to station B_1 (B_2).

$$i_{j1}^0 = \min (i \mid w_{ji} = B_1), j = 1,..., N, \tag{5.40}$$

$$i_{j2}^0 = \min (i \mid w_{ji} = B_2), j = 1,..., N. \tag{5.41}$$

Let $S_{B_1}(j)$ ($S_{B_2}(j)$) be the set of processing steps at bottleneck station B_1 (B_2) for type j jobs,

$$S_{B_1}(j) = (s_{ji} \mid w_{ji} = B_1), j = 1,..., N, \tag{5.42}$$

$$S_{B_2}(j) = (s_{ji} \mid w_{ji} = B_2), j = 1,..., N, \tag{5.43}$$

and let F_{j1} (F_{j2}) be the set of type j steps prior to the first visit of a type j job to station B_1 (B_2),

$$F_{j1} = (s_{ji} \mid i = 1,...,i_{j1}^0 - 1), j = 1,..., N, \tag{5.44}$$

$$F_{j2} = (s_{ji} \mid i = 1,...,i_{j2}^0 - 1), j = 1,..., N. \tag{5.45}$$

Define C_{j1} (C_{j2}) to be the total processing time from the start of a type j job (step s_{j1}) to its first visit to station B_1 (B_2), excluding the first visit to each bottleneck station.

$$C_{j1} = \sum_{i=1}^{i_{j1}^0 - 1} d_{ji},\ j = 1,..., N, \tag{5.46}$$

$$C_{j2} = \sum_{i=1}^{i_{j2}^0 - 1} d_{ji},\ j = 1,..., N, \tag{5.47}$$

and let

$$C_1^{\max} = \max (C_{j1} \mid j = 1,..., N), \tag{5.48}$$

$$C_2^{\max} = \max (C_{j2} \mid j = 1,..., N). \tag{5.49}$$

Let n_{ji}^1 (n_{ji}^2) be the step number of the next visit of a type j job to station B_1 (B_2) given the job is presently at step s_{ji} (i-th step of type j product recipe). Furthermore, let P_{j1} (P_{j2}) be the sets of type j steps such that total processing of that step plus all other type j steps prior to the next visit of a type j job to station B_1 (B_2) (excluding processing at the corresponding bottleneck station) is less than $\beta_1 C_{j1}$ ($\beta_2 C_{j2}$), where $\beta_1 > 0$ ($\beta_2 > 0$).

$$P_{j1} = (s_{ji} \mid \sum_{l=i}^{n_{ji}^1 - 1} d_{jl} < \beta_1 C_{j1}),\ j = 1,..., N, \tag{5.50}$$

$$P_{j1} = (s_{ji} \mid \sum_{l=i}^{n_{ji}^1 - 1} d_{jl} < \beta_2 C_{j1}),\ j = 1,..., N. \tag{5.51}$$

Let

$$Q_{j1} = F_{j1} \cup P_{j1} \cup S_{B_1}(j),\ j = 1,..., N, \tag{5.52}$$

$$Q_{j2} = F_{j2} \cup P_{j2} \cup S_{B_2}(j),\ j = 1,..., N, \tag{5.53}$$

be the set of critical steps for type j jobs, *i.e.* steps that are within a critical time

factor of bottleneck station B_1 (B_2) or steps in the pipeline from start to the first visit to a specific bottleneck station.

As before, let $D(B)$ be the index set of machines in *down* state at work station B. Define the expected total repair time of equipment at a bottleneck to be given by equation 5.7.

Let W_1 (W_2) be the sum over all product types of the total j type work content at the next visit to bottleneck station B_1 (B_2) of type j jobs at steps in set Q_{j1} (Q_{j2}) plus the total expected repair time R_1 (R_2) of equipment in *down* state at station B_1 (B_2),

$$W_1 = \frac{\left\{ R_1 + \sum_{j=1}^{N} \left(\sum_{l \in Q_{j1}} \eta_{jl} d_{jn_j^l} \right) \right\}}{m_1}, \tag{5.54}$$

$$W_2 = \frac{\left\{ R_2 + \sum_{j=1}^{N} \left(\sum_{l \in Q_{j2}} \eta_{jl} d_{jn_j^l} \right) \right\}}{m_2}. \tag{5.55}$$

If W_1 (W_2) falls below a critical value $\alpha_1 C_1^{\max}$ ($\alpha_2 C_2^{\max}$), where $\alpha_1 > 0$ ($\alpha_2 > 0$) is a safety factor, there is danger that bottleneck station B_1 (B_2) will starve. Release strategy SA(N,2) triggers a job start at this time, to avoid starvation of that bottleneck. The job type with greatest negative deviation from its target mix is selected.

5.6 – Summary

In this Chapter we introduced a class of new job release mechanisms called Starvation Avoidance (SA). These procedures make an analogy between work content at one or more bottleneck workstation (in a job shop) and inventory (in inventory control). In inventory control, when the the difference between items (on hand and on order) and demand is less than a given value, a new item is ordered. In this analogy,

when work content (on hand) at the bottleneck and work content (on order) moving towards the bottleneck (and expected to arrive within a limited time frame) falls below a given value, a new job is released into the shop.

We present four special cases for SA: the one product one bottleneck case, SA(1,1); the multi-product one bottleneck case, SA(N,1); the one product two bottleneck case, SA(1,2); and the multi-product two bottleneck case, SA(N,2). We discuss a dispatching scheme designed to aid SA achieve its objective, *i.e.* minimize idle time on critical shop equipment.

Extensive testing was done on one case of Starvation Avoidance: SA(1,1). These results are presented in the next Chapter.