A SIMPLE RELEASE POLICY FOR NETWORKS OF QUEUES WITH CONTROLLABLE INPUTS

Izak Duenyas
Department of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 92-45

August 1992
Revised June 1993

A Simple Release Policy For Networks Of Queues With Controllable Inputs

Izak Duenyas

Department of Industrial and Operations Engineering

University of Michigan, Ann Arbor, MI 48109

Technical Report 92-45

August, 1992

Revised, June, 1993

A Simple Release Policy for Networks of Queues with Controllable Inputs

IZAK DUENYAS

Department of Industrial and Operations Engineering

The University of Michigan

Ann Arbor, Michigan 48109

Abstract

In a recent paper, Wein (1992) addressed the problem of scheduling a network of queues. Given a multistation, multiclass queueing network, the problem involves deciding when to release a job to the network as well as how to sequence jobs at each machine in the network in order to meet a desired throughput level. By approximating this problem by a control problem involving Brownian motion, Wein derived effective heuristics, which easily outperformed traditional work release and sequencing rules. However, Wein's work release rules are complex and his sequencing rules are dynamic. In this paper, we test the performance of a simpler work release policy based on CONWIP (Constant Work-in-Process) in conjunction with static sequencing rules. The results of our simulation study indicate that this simple work release rule can be rather effective.

1 Introduction

In a recent paper, Wein (1992) addressed the problem of input control (subject to a specified input mix) and priority sequencing in a multistation, multiclass queueing network with general service time distributions and a general routing structure. Wein explains that this problem is motivated by a scheduling problem encountered in many factories. The factory is modeled as

a network of workstations where each workstation consists of a single machine. It is assumed that the factory produces k different types of products. Furthermore, a throughput rate of λ is required for the system. It is also required that product i constitute a proportion of q_i of the total releases into the system and hence of the throughput. It is assumed that the raw material for each type of product is always available. The objective is to meet the desired production rates and to minimize the average *cycle time* of the jobs where the cycle time is the amount of time that the jobs spend in the factory. The input control problem is the problem of deciding when to release a job to the system. Also, a decision must be made as to which type of job will be released to the system. The priority sequencing rule decides in what sequence jobs should be processed at each machine.

Wein(1992) addresses the above problem by approximating the problem as a control problem involving Brownian motion. The solution procedure involves reformulating the Brownian control problem to get a workload formulation, and then to solve an embedded linear program to obtain dynamic reduced costs that specify the sequencing policy, and then to use a finite difference approximation to the multi-dimensional constrained singular control problem to obtain another linear program that derives the reflecting boundary. The resulting input and sequencing policies require the knowledge of the queue lengths at each station at all times (i.e., deciding what to process at a certain station requires the knowledge of the number of jobs of each class at all the other stations.)

As Wein(1990a) notes, the workload regulating policy derived in Wein(1990a) and Wein(1992) can be viewed as an extension of the *pull* system from its traditional setting of a flow line operation to the more complex setting of a job shop operation. However, Wein(1992) points out that the scheduling policy proposed in Wein(1992), and in particular the workload regulating release policy is very tedious, and hence may not be practical for implementation purposes. In fact, the dynamic nature of the sequencing rules where the priority of jobs at a certain station depends on the workload at all the stations also requires a fairly sophisticated information system. Furthermore, as Wein and Chevalier (1992) note, the dynamic nature of the release

and sequencing policies make it very difficult to use these policies in conjunction with due date setting policies, in make-to-order environments where incoming orders are assigned due dates. In fact, due to this difficulty, when due dates have to be quoted in systems with more than 2 machines, Wein and Chevalier (1992) recommend using a more traditional closed-loop release policy along with static sequencing rules derived in Chevalier and Wein (1990).

In this paper, we test the performance of a simple work release rule that is based on the CONWIP release mechanism. We illustrate why this release rule should perform better than the traditional closed loop input policy that Wein(1992) compared his heuristics against. In a simulation study, where the test problems include the example networks from Wein(1992) and Wein(1990a), we compare the performance of this release rule to the more complicated release rules in Wein (1992), and find that this release rule can be rather effective.

The rest of this paper is organized as follows. In the next section, we briefly describe the modified CONWIP release policy. In Section 3, we report our simulation results. The paper concludes in Section 4.

2 CONWIP Release Policies

In this section, we briefly describe the CONWIP release policy. CONWIP is a generally applicable pull system similar to Kanban. In a Kanban system, jobs are pulled by each station from the previous station. That is, the downstream station authorizes the previous station to start work on a new unit. This can be done by cards (kanbans) sent to the previous stations. However, in CONWIP, whenever a job is completed at the last station, the last station sends a card to the first station authorizing it to start work on a new unit. This control mechanism dates to Jackson's ground-breaking work (1963), and has recently gained a lot of attention (see, Spearman et al. (1989,1990), Spearman and Zazanis (1992)). For the simple system described above, we make the following observation:

Observation 1: *In a CONWIP system, a card sent to the first station from the last station*

authorizes the machine to release a new job to the system. However, there is no point in releasing this job to the system until the time when its processing at the first station is going to begin.

The above observation just states the obvious fact that in a simple CONWIP system that produces a single product, and where each job visits a station only once, there is no need to have a queue of jobs at station 1. Whenever a job is completed at the first station, a new job can be released into the network if there is a card authorizing this release. There is no point in releasing the job earlier than that since the job will just have to wait until the machine is ready to process it. Note that since sequencing is not an issue in this case (jobs are processed according to the FCFS rule), the release rule does not involve a decision on which type of job should be released.

There are two types of input control that could be used when implementing a CONWIP system with multiple products and general routes. One type of input control is the control structure that was described by Wein (1992) and Wein (1990a) as the single chain closed loop input policy. In this control rule, whenever the processing of a job is completed at the last station on that job's route, the release of a new job is authorized. However, this new job is not necessarily of the type that was just finished. Jobs are released according to a predetermined work release sequence that satisfies the constraint that the long run proportion of jobs of type i has to be q_i. For example, if 2 types of jobs are being produced in equal proportions, then jobs may be released according to the sequence ABABAB...However, we again note that for this multi-product system, we can make the same observation that we made previously for the single-product system. That is, there is actually no point in releasing the new job until the time when its processing is going to begin. In this case, this release time will depend on the sequencing rule used at the first machine. A new job would be released only if that job's release had the highest priority among all the options available to the machine at that time, and the job's processing would begin immediately after its release. We will refer to this release rule as the single chain closed loop input policy (S-CLOSED).

The release rule described above is essentially the rule that Wein(1992) tested except for the fact that in his simulation study, the above observation about not releasing a job at the first machine until the time when the job's processing is going to begin was not made use of. We note that both in Wein (1992, 1990a) and also in this paper, it is assumed that raw material is always available. One of the motivations of minimizing WIP, as Wein (1990a) has written, is that "by reducing the number of jobs on the factory floor, the benefits of Just-In-Time manufacturing can be realized. For example, quality problems will be detected faster, and thus there will be less rework and scrap of jobs. By reducing the cycle time of jobs, the factory can gain *flexibility:* The system will be more capable of very fast turnaround on individual orders, and the factory may more readily adapt to a changed order because the corresponding job may not have begun its processing." (Wein, 1990a, p.1068). Clearly, even if jobs are released to the first machine on their route before the machine is ready to process them, under the above assumption that raw material is always available, this amounts to moving the raw material from raw material storage to the queue in front of the first machine. Clearly, no flexibility is lost, and no value is added to the inventory until processing of the material begins. Therefore, under the assumption that raw material is always available, when comparing the effectiveness of release mechanisms, it is more appropriate to compare cycle times from the point when some value has been added to the unit since quality problems, loss of flexibility etc. do not occur until processing begins. For these reasons, in our simulation study, in Section 4, we compare the cycle time under different release mechanisms from the point when processing of the unit is started for the first time.

An alternative scheme to the S-CLOSED input policy, described above, is to set separate card counts (WIP levels) for each type of product, and to authorize the release of a new job of type A only when a job of type A has just been finished. Again, the arrival of an authorization to release a new unit of type A does not mean that a new unit will be immediately released. We refer to this scheme as the multiple chain closed loop input policy (M-CLOSED). In this case, card counts for each type of product are set to achieve the throughput level and the

product mix desired.

The M-CLOSED policy described above has certain advantages with respect to the S-CLOSED policy. We demonstrate these advantages with a specific intuitive example. Consider a network with two product types and four machines. Product A requires processing at machines 1,2, and 3 while Product B requires processing at machines 3 and 4. Assume that the desired product mix is to have 50% of the products to be of type A. Now consider a situation where a job of type A experiences a very long delay at machine 2. For example, this may be due to the failure of the machine. In this case, releasing jobs in the order A-B-A-B would eventually lead to a very high number of jobs of type A queueing in front of machine 2. This is due to the fact that while all jobs of type B are quickly processed and leave the system, jobs of type A are awaiting the repair of the machine at station 2. Since the total number of jobs in the network is bounded above by the card count, if this delay at machine 2 is long enough, eventually the network will have all the jobs in front of machine 2 and machines 3 and 4 will be starved. Essentially, this release mechanism could cause the whole plant to shut down if a failure at a certain machine lasted long enough. Obviously, there is no point in shutting down the production of units of type B because of the failure of machine 2 since products of type B do not require processing at machine 2. The M-CLOSED policy avoids this situation because it has separate cards for products A and B. Hence, in this case, the maximum number of units waiting in front of machine 2 would be equal to the number of cards for product A in the system. Machines 3 and 4 would not be starved because of the failure of machine 2. Since a certain throughput level is desired for both products, the S-CLOSED policy would try to avoid the situation described above by having a high number of cards in the system. This is due to the fact that the greater the number of cards in the system, the longer it takes machines 3 and 4 to starve. However since setting multiple WIP levels avoids this situation, one would expect the M-CLOSED policy to require less WIP and result in lower cycle times to achieve a desired throughput level. It is well-known that jobs spend a major proportion of their cycle times waiting to be processed at bottleneck machines. The above example indicates

that especially in networks where the machines that different jobs spend the majority of their waiting times are different, M-CLOSED should perform much better than S-CLOSED. We verify this conjecture empirically in our simulation study in the next section.

The work-regulating (WR) release rule that Wein (1992, 1990a) derives uses a more complicated mechanism for release of jobs. Releases are only authorized when machines face the threat of starvation, and there is not too much work already present in the system. Once a release is authorized, however, the class of the entering job type has to be decided. The first step of the procedure to select the class of the job type to be released is to check to see if the actual mix of jobs already released into the network is sufficiently close to the desired mix. As an example, consider a network where two types of jobs A and B, in proportions q_A and q_B (where $q_A + q_B = 1$) have to be produced. Suppose $N_A(t)$ and $N_B(t)$ units of jobs A and B have been released into the network until time t. Then, we can also compute the mix of jobs released into the network until time t by computing $q'_A = \frac{N_A(t)}{N_A(t)+N_B(t)}$, and $q'_B = 1 - q'_A$. If the actual mix is sufficiently close to the desired mix, (e.g., $\frac{|q'_a - q_a|}{q_a} < \delta$), then the WR release rule selects the class of job to be released by trying to balance the workload [1]. However, if the actual mix at time t is not sufficiently close to the desired mix, then the class of job that is farthest from the target is released.

Despite the fact that the WR rule is much more sophisticated than the S-CLOSED policy, the additional controls imposed on the *timing* of releases, and the type of job to be released in the WR release rule do not ensure that failures at one machine will not cause starvation in an unrelated machine. As an example, we reconsider the case described above with 2 jobs

[1] We note that this is slightly different than the expression in equation (108) of Wein(1992) where it is suggested that a parameter N^* should be selected that specifies how close the actual entering class mix must stay to the target mix q such that for each entering class j and k, $q_j N_j(t) - q_k N_k(t) \leq N^*$. This equation (i.e., equation (108) of Wein(1992)) is wrong. The constraint $N_j(t)/q_j - N_k(t)/q_k \leq N^*$ will work, but then N^* will depend on the total duration of time that the system is run (i.e. N^* depends on the total simulation run time.). Checking that the percentage difference between desired and actual mix is never greater than a specified parameter, as described above, is therefore preferable.

on 4 machines. When machine 2 fails, this does not immediately cause a problem for jobs of type B, if the proportion of jobs of type B released until that point in time is close to the desired proportion. The WR release rule will keep releasing type-B jobs, and will stop releasing type-A jobs. However, if the failure is long enough, then the actual proportion of type-B jobs released into the system will be greater than the desired proportion. At that point, if a release is authorized because machines 3 and 4 are facing the threat of starvation, the input rule will require a type-A job to be released, and not a type-B job, to achieve the required mix in releases, even though this will only result in greater congestion at machine 2. Hence, the WR policy postpones the problem faced by the S-CLOSED policy, and thus should result in better performance than S-CLOSED, however, the problem still occurs if a failure lasts long enough (or processing of a job lasts considerably longer than average in systems with no failures).

We note that the main disadvantage of the M-CLOSED policy compared to the WR or the S-CLOSED rules is that it is, in general, not easy to estimate the amount of WIP necessary to achieve a given throughput level and mix. For example, in the S-CLOSED policy, the only parameter that has to be set is the total number of cards in the system. Furthermore, adding a card to the system always increases the total throughput. In contrast, under the M-CLOSED policy, card counts (WIP levels) for each type of product have to be set and an increase in the card count for one type of job will increase the throughput for that type of job, but not necessarily the total system throughput. Hence, the WR and S-CLOSED rules have the advantage with respect to the M-CLOSED rules that the output mix under these policies is more predictable. However, as we previously mentioned, the WR policy requires a very sophisticated information and computation system, and for make-to-order systems with more than 2 bottleneck machines, it is very difficult to use the WR release rule in conjunction with a rule for quoting due dates to customers. The S-CLOSED rule is the simplest of all three rules and is desirable due to its ease of implementation. However, as we will demonstrate in the next section, its performance is worse than the other two rules.

The above discussion indicates that it is very desirable to identify the key network charac-

teristics that favor one release rule over another. For example, in networks where one would not expect the M-CLOSED rule to outperform the S-CLOSED rule significantly, it would be preferable to use the S-CLOSED rule since it is simpler to implement. Similarly, if a company, operating in a make-to-stock environment, has already invested in a sophisticated information system and is thus capable of implementing the WR release rule, the WR rule may be preferable to the M-CLOSED rule, if the network characteristics indicate that the M-CLOSED rule would not outperform WR significantly. In our simulation study in the next section, we test the three rules on different networks to gain an understanding of the characteristics that favor a particular rule.

3 A Simulation Study

In this section, we report the results of our simulation study in which we tested the release policies described in the previous section. We present our simulation results for the example networks provided in Wein(1992), and Wein(1990a) as well as six other representative 2-machine networks. For each case, we ran 10 independent runs, each of duration 22000 time units, and the first 2000 time units were truncated. We assumed that preemption (with resume) was allowed in all examples.

The first example is the three machine, three job type example from Wein(1992). Table 1 displays the route for each job and the mean processing times. For this problem, the desired throughput was 0.149 and the proportion of each type of product was 1/3. Hence, this corresponds to a desired production rate of 0.04967 units per unit time for each type of product. It is assumed that all processing times are exponential.

Wein(1992) tested five different policies for this example. The policies tested by Wein and the results that he obtained are displayed in Table 2. The first two policies that Wein tested released work into the system at deterministic time intervals. In the first case, first-come-first-served rule was used, and in the second case the shortest expected remaining time rule was used. The second and third policies correspond to what we referred in the previous section as

Customer Type	Route	Mean Service Times
A	$3 \rightarrow 1 \rightarrow 2$	6.0 4.0 1.0
B	$1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2$	8.0 6.0 1.0 2.0 7.0
C	$2 \rightarrow 3 \rightarrow 1 \rightarrow 3$	4.0 9.0 4.0 2.0

Table 1: Example Problem 1 (Wein(1992))

Input Rule	Sequencing Rule	Throughput Rate	Cycle Time (95 % CI)
DETERMINISTIC	FIFO	0.149	144 (10.4)
DETERMINISTIC	SRPT	0.149	182 (15.7)
S-CLOSED(18)	FIFO	0.149	120 (0.8)
S-CLOSED(25)	SRPT	0.149	166 (1.1)
WR (Wein,1992)	DRC	0.149	85.4 (1.1)

Table 2: Results for Policies Tested in Wein(1992)

a single-chain closed-loop release policy (S-CLOSED). However, we remind the reader that in Wein's study, Observation 1 is not used. Finally, the last policy displayed in Table 2 is the dynamic release and input heuristic derived by Wein(1992). Table 2 displays the throughput levels, and the average cycle times, along with 95 % confidence intervals for each policy tested in Wein(1992).

The policies that we tested for this problem are displayed in Table 3. Before we report the results of our study, we define the notation used in Table 3. As noted previously, M-CLOSED is the CONWIP policy which sets separate card counts for each job, and which results in a network that can be represented as a multiple chain closed queueing network. Furthermore, we also present the card counts that achieved the desired throughput levels.

We note that since card counts can only be set in discrete quantities, it may not be possible to get the exact desired throughput levels. This is due to the fact that having x_i cards for

product i may result in less than the desired throughput for product i, while having x_i+1 cards may result in greater than the desired throughput. However this problem can be solved in the following manner. Let $z_i(t)$ denote the number of jobs of type i released into the network until time t. Every time $z_i(t)$ mod $l_i = 0$, an extra unit can be released into the network. However, when this extra unit has been finished, it does not automatically trigger the release of a new *extra* unit. Rather, another extra unit is released when $z_i(t)$ mod $l_i = 0$, once again. The appropriate value of l_i can be set by simulation. The notation M-CLOSED$(x, y+w, v+u)$ denotes that there are x,y and v cards respectively for job types 1,2 and 3.Furthermore every time $z_2(t)$ mod $w = 0$, an extra unit of type 2, and every time $z_3(t)$ mod $u = 0$, an extra unit of type 3 is released.

We also note that in some cases, even having a single card for a particular job may result in greater than desired throughput for that job. This may be the case when the total processing time that a certain product requires is very little or its desired throughput is very low. In this study, for such products, we used *deterministic* release where a release is authorized deterministically every $1/\lambda_i$ time units, and λ_i is the desired production rate for product i. However, as before, the job is only released when the first machine is ready to process it. Hence, the notation $M - CLOSED(Det, x, y)$ means that jobs for product 1 are authorized to be released at deterministic time intervals, with WIP levels of x and y for products 2 and 3. If the system has many products that were run infrequently, it may be more appropriate to set a total WIP level for those infrequent products, and release these infrequent products according to a set sequence, while setting separate WIP levels for each of the more frequently run products.

Finally, the sign $OB1$ next to any policy indicates that Observation 1 is being used. Otherwise, we have used the policy of releasing a job to the first machine immediately at the arrival of a card authorizing its release, as in Wein(1992) and Wein(1990a). In Table 3, we tabulate the production rates for each type of product along with the average cycle time (and the 95% confidence intervals). We note that the desired production rate for each type of product is

Input Rule	Sequencing	Thr1	Thr2	Thr3	Cycle Time
DETERMINISTIC	SEPT	0.0497 (0.0001)	0.0497 (0.0001)	0.0497(0.0001)	87.4 (10.0)
S-CLOSED(13)	SEPT	0.0503(0.0016)	0.0503 (0.0016)	0.0503(0.0018)	87.2 (3.5)
M-CLOSED(Det, 7+15,2)	SEPT	0.0497 (0.0001)	0.0497 (0.0015)	0.0497 (0.0014)	71.5 (2.9
S-CLOSED(10)	WBAL	0.0497 (0.0002)	0.0497 (0.0002)	0.0497 (0.0002)	67.1 (2.(
M-CLOSED(Det, 6, 1+2.5)	WBAL	0.0497(0.0001)	0.0503 (0.0012)	0.0498 (0.0012)	59.4 (2.4
M-CLOSED(Det, 6, 1+2.5),OB1	WBAL	0.0497(0.0001)	0.0503(0.0012)	0.0498 (0.0012)	44.4 (2.1

Table 3: Results of Simulation Study for Example 1.

0.04967 products per unit time.

The first two rules that we tested used shortest expected processing time (SEPT) as a sequencing rule. The first policy that we tested released jobs of types A,B and C deterministically at multiples of time 3/0.149. To test how the SEPT policy performs in this problem (note that Wein(1992) did not test this sequencing rule) we first released jobs immediately into the system as in Wein's study (that is, we did not make use of Observation 1). This naive release policy, along with the SEPT rule led to an average cycle time of 87.4. Next, we tested the SEPT sequencing rule with the S-CLOSED release policy where we released jobs according to the sequence ABCABCABC. Again, we did not make use of Observation 1 and released jobs into the system immediately at the first machine. In this case, a total card count of 13 was enough and the average cycle time was 87.2.

We next tested the performance of the M-CLOSED release rule. We note that the results reported here do not necessarily represent the best that this rule can do for this problem. We searched for the card counts by simulation. We stopped when we found the first allocation of cards that resulted in production rates which were close to the desired production rates. Hence, these card counts do not necessarily represent the *optimal* card counts. We note that setting appropriate card counts is a problem that has to be addressed in any implementation of a pull system. In particular, implementing a kanban system requires setting card counts for each machine and for every product type, a vastly more complicated task in a large network.

We expected the M-CLOSED rule to result in better average cycle time than the S-CLOSED rule. The results verified our intuition. When we used the SEPT sequencing rule with the M-CLOSED rule, we obtained an average cycle time of 71.5 without making use of Observation 1. (Note that even having one card for product 1 was too high in this case, hence we used deterministic release for product 1, as described above).

We also tested the performance of the static work balancing sequencing rule (WBAL) derived in conjunction with S-CLOSED release rule. Harrison and Wein(1990) derived this sequencing rule for systems with 2 machines, and Chevalier and Wein (1990) extended the rule to systems with any number of machines. WBAL was derived with the purpose of minimizing average cycle time when the S-CLOSED release rule is used. Hence, in one sense, the cycle time obtained when the S-CLOSED rule is used in conjunction with the WBAL sequencing rule represents the best that this rule can do (at least among known static sequencing rules). The readers are referred to Harrison and Wein (1990) and Chevalier and Wein (1990) for the derivation of these rules.

Assuming that jobs change classes at each machine on their route in such a way that a job of type A is of class A1 on the first machine on its route, of class A2 on the second machine on its route, and so on, there would be twelve classes (A1,A2,A3,B1,....,B5,C1,....C4). For this example, Chevalier and Wein (1990) report that the WBAL sequencing rule would assign priorities to the jobs in the order B4,C3,A2,B1 at machine 1 (with the first one having the highest priority), in the order A3,C1,B5,B2 in the second machine and in the order B3,C4,A1 and C2 at the third machine. When we used this static sequencing rule along with the S-CLOSED release rule, we found that a total maximum WIP level of 10 was enough to achieve the throughput, and without making use of Observation 1, the average cycle time was 67.1. Interestingly, this is a much lower cycle time than the cycle time of 85.4, obtained by using the more complicated dynamic release and sequencing rules in Wein(1992).

We next tested the performance of the M-CLOSED policy with the WBAL sequencing rule. We note that whereas the WBAL sequencing rule was derived in conjunction with the

S-CLOSED release rule in order to minimize the average cycle time of jobs, we do not know of any sequencing rules that have been derived in conjunction with the M-CLOSED release rule. Hence, the performance of the M-CLOSED rule with the WBAL sequencing rule does not, in any way, represent the best that this rule can do. However, we conjectured that the WBAL sequencing rule should still work very well when used with the M-CLOSED release rule, since its main purpose is to sequence jobs in such a way that machine idleness is minimized in a closed queueing network.

Our simulation results verified our conjecture that the static WBAL sequencing rule would work very well along with the static M-CLOSED release rule. The average cycle time was 59.4, without the use of Observation 1. When we used Observation 1, the cycle time was decreased to 44.4. Since the result in Wein(1992) was obtained without the use of Observation 1, the cycle time of 59.4 obtained by M-CLOSED without the use of Observation 1 is more appropriate for comparison purposes in this case, since using Observation 1 would also decrease the cycle time obtained in the (WR,DRC) case. Hence, the much more sophisticated dynamic release and sequencing rules in Wein (1992) result in nearly 44 % greater average cycle time for this problem.

In an effort to characterize the cases where either of the three release rules may be more appropriate, we next focused on examples with 2 bottleneck machines. Examples 2-6 include cases with deterministic routing, while examples 7 and 8 include jobs that have probabilistic routing corresponding to the cases where the manufacturing facility may face problems of scrap or rework. The data for Examples 2 through 7 is in Table 4. In Examples 2-5, all processing times are exponential. In Example 6, all processing times at machine 1 are deterministic, and all processing times at machine 2 are exponential. In Example 7, all machines have exponential distributions. Jobs of type A visit the machines 1 and 2 in the order 1,2,1,2. However, after being processed at machine 2, jobs of type A are inspected, and with probability 0.2 require rework at machine 2. Jobs are not inspected again after the rework and both the original processing and the rework takes an exponential amount of time with mean 5.0. In Example

Case	Utilization	Job	Route	Mean Service Times	Thrput
2	(0.889,0.889)	A	$1 \rightarrow 2$	4.0 1.0	0.0635
		B	$1 \rightarrow 2 \rightarrow 1 \rightarrow 2$	8.0 6.0 2.0 7.0	0.0635
3	(0.97, 0.97)	A	$1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1$	1.0 2.0 3.0 5.0 2	0.0882
		B	$1 \rightarrow 2 \rightarrow 1$	3.0 4.0 2.0	0.0882
4	(0.9, 0.9)	A	$1 \rightarrow 2 \rightarrow 1 \rightarrow 2$	15.0 8.0 9.0 4.0	0.025
		B	$1 \rightarrow 2 \rightarrow 1 \rightarrow 2$	1.5 3.0 3.0 6.0	0.06666
5	(0.9075,0.9075)	A	$1 \rightarrow 2 \rightarrow 1$	3.0 2.0 8.0	0.0605
		B	$2 \rightarrow 1 \rightarrow 2$	3.0 3.0 8.0	0.0605
		C	$1 \rightarrow 2$	1.0 2.0	0.0605
6	(0.98,0.98)	A	$1 \rightarrow 2 \rightarrow 1$	3.0 8.0 3.0	0.08166
		B	$1 \rightarrow 2 \rightarrow 1$	3.0 4.0 3.0	0.08166
7	(0.965, 0.965)	A	$1 \rightarrow 2 \rightarrow 1 \rightarrow 2$	2.0 5.0(+5.0 w.p 0.2) 2.0 5.0(+5.0 w.p. 0.2)	0.06585
		B	$2 \rightarrow 1 \rightarrow 2 \rightarrow 1$	1.0 4.0 1.0 4.0	0.0877

Table 4: Data for Examples 2-7

8, there are two types of jobs A and B. Jobs of type A first visit machines 1 and 2, (with mean processing time 4.0 at machine 1 and 1.0 at machine 2). After this initial processing is completed at machine 2, the job is inspected. With probability 0.2, the job has to visit machines 1 and 2 again (with mean processing times 5 and 2), and with probability 0.8, the job has to visit machine 2 only (with mean processing time 2). Jobs of type B follow the deterministic route 1-2-1-2 with mean processing times 2.0, 3.0, 1.0 and 2.0. In this example, all processing times at machine 1 are exponential, and all processing times at machine 2 are deterministic. The desired throughput is 0.24625, and the desired mix of product A is 0.5, thus the desired production rates for both products is 0.123125. This throughput rate and mix results in a utilization level of 0.985 for both machines 1 and 2.

Given the success of the WBAL sequencing rule with the different release rules in the three machine example above, we compared the performance of different release rules in conjunction with this sequencing rule. This rule is rather simple in the 2-machine case. For a network with

2 machines and K classes, define M_{ik}, $i = 1,\ldots,2$ $k = 1,\ldots,K$ as the expected total amount of time that machine i must devote to a class k job until it completes its route. Also, let ρ_1 and ρ_2 be the utilizations for machines 1 and 2. The WBAL sequencing rule ranks each class $k = 1,..K$, by the index $\rho_2 M_{1k} - \rho_1 M_{2k}$ and awards higher priority at station 1 (respectively, at station 2) to the class with the smaller (respectively, larger) values of the index. As we mentioned above, this policy was derived in conjunction with the S-CLOSED rule in Harrison and Wein (1990) and was shown by Wein(1990) to work well with the work-regulating release rule. We also tested the dynamic DYN sequencing rule derived in Wein(1990a) with the WR release rule, but found no statistical difference between using DYN and WBAL. Since the same sequencing rule is being used in Examples 2-8, the differences in the average cycle times obtained can be completely attributed to the performance of the different release rules.

We report the simulation results for Examples 2-8 in Table 4. We tested the WR, S-CLOSED and M-CLOSED release rules for each example in conjunction with the WBAL sequencing rule. The WR release rule used both the rules derived in Wein (1990a) for the 2-machine problem for deciding when a job should be released and also the input heuristic derived in Wein(1992) for deciding which job to release. As in the simulation study in Wein(1992), we required that the entering job mix be within one-half of 1 % of the desired mix. For example, in a problem with two job types (A and B), if the desired mix is 50-50, then if the actual mix of A's is less than 49.75 then the rule requires an A to be released, if it is greater than 50.25 then a B has to be released, when a release is authorized. If the actual mix is between 49.75 and 50.25, then we choose the class which balances the workload as described in equations (110-111) of Wein(1992). The release rules of Wein (1990a) require the setting of two parameters ϵ_1 and ϵ_2 by simulation which correspond to an enlargement of the release region. We report the parameters that achieved the desired throughput levels in Table 4.

Examples 2 and 3 are two examples where we would not expect the M-CLOSED rule to do better than the WR rule. In Example 2, (taken from Wein (1990a)), job B accounts for the great majority of both machines' utilization, and its total processing requirement (i.e., the sum

of its processing times at all machines) is much greater than job A's. Hence, the average cycle time is very highly dependent on the cycle times of B-type jobs, which have significantly long processing times at both machines. Thus, even if the processing of a certain job of type A takes much longer than average on machine 1, it is very hard for that to cause the network to starve for type B jobs, since type-B jobs have so much longer processing times. In fact, as we expected, the average cycle times obtained by all three rules were very close in this case. Note that the results reported in Wein (1990a, 1992) for this case do not make use of Observation 1. We resimulated the WR rule by making use of Observation 1, and found that the difference in the cycle time obtained by the M-CLOSED and WR rules were statistically insignificant. Example 3 is also a case where we would not expect significant differences between the rules. This is due to the fact that all the jobs in this example require pretty much the same amount of processing at all machines. In fact, there was no statistical difference between the performance of the WR and M-CLOSED rules, although both rules significantly outperformed the S-CLOSED rule.

We expected the M-CLOSED rule to have an advantage in Examples 4-6. In example 4, 66% of the total processing time of job A is at machine 1, and 33 % is at machine 2. In example 5, this ratio is similar. Of the total (raw processing) time it takes to process a unit of A, 66 % is at machine 1, and 33 % is at machine 2. In example 6, this imbalance is less. 60 % of job B's total processing time is at machine 1, and 40 % is at machine 2. In fact, in Example 6, machine 1 spends an equal amount of time processing jobs of type A and B.

The results in Table 5 demonstrate the significant difference in the average cycle times obtained by the three rules in Examples 4-8. In particular, in Example 6, the WR rule resulted in nearly 35% greater average cycle time, and the S-CLOSED rule resulted in nearly 100 % greater average cycle time than the M-CLOSED rule. Note that the average cycle times include the (controllable) average queueing time and the (uncontrollable) average processing time. Hence, if we compare the average queueing times, we find that the difference is even greater. We note that as the utilization of the machines increases, one would expect the difference in the performance of the three rules to increase. In fact, in examples 4 and 5, the utilization of

the two machines equals 0.9, while in example 6 (which is actually a more balanced example in terms of how the total processing times of each job is allocated to machines) the utilization equals 0.98, which explains the greater percentage difference in average cycle times obtained by the M-CLOSED rule and the other two rules in Example 6. The results were similar for Examples 7 and 8 which contain probabilistic routing due to rework. The M-CLOSED rule again significantly outperformed the other two rules.

These examples are representative of our experience with the WR, M-CLOSED and S-CLOSED rules. We found that there is little difference between the M-CLOSED and WR rules when

1. All jobs in the network have similar processing requirements at the bottleneck machines in the network, resulting in each job spending nearly equal amounts of time at the different machines.

2. One job type has much greater processing requirements than the other jobs in the network, thus resulting in a network where all machines process one job type a large proportion of the time.

If either of the above conditions did not hold, we found that the M-CLOSED rule outperformed the WR rule, in many cases rather significantly. In fact, Example 6 demonstrates that even a small imbalance in how the processing times of jobs are allocated to the different machines seems to give the M-CLOSED rule a large advantage. Furthermore, in all of our test examples, the M-CLOSED rule greatly outperformed the S-CLOSED rule.

4 CONCLUSIONS AND FURTHER RESEARCH

In this paper, we tested the performance of a simple release policy based on the Constant WOrk-In-Process (CONWIP) release mechanism. When used with the static work balancing sequencing rules developed in Harrison and Wein (1990) and Chevalier and Wein (1990), this release policy worked very well. In particular, in our simulation study, this rule outperformed

Case	Input Rule	Thr1	Thr2	Thr3	Cycle Time
2	WR (1,1) (Wein 1992, 1990a)	0.0635	0.0635	-	34.7
	WR, OB1	0.0635	0.0635		29.8 (1.5)
	M-CLOSED(Det, 3+7), OB1	0.0635	0.0640		29.7 (1.4)
	S-CLOSED(5), OB1	0.0640	0.0640		30.6 (1.4)
3	M-CLOSED(11,3+40),OB1	0.0880(0.005)	0.0886 (0.005)	-	79.7(1.2)
	S-CLOSED(20), OB1	0.0883(0.003)	0.0883 (0.003)		113.4 (1.4)
	WR(12,12), OB1	0.0880 (0.002)	0.0882 (0.002)		81.1 (2.3)
4	M-CLOSED (2+18, 2+22), OB1	0.025 (0.001)	0.0668 (0.0032)	-	39.8 (1.0)
	S-CLOSED (6), OB1,	0.0251 (0.0008)	0.0669 (0.0024)		49.5 (1.6)
	WR(8,8), OB1	0.0249 (0.0009)	0.0666 (0.0019)		43.3 (1.7)
5	M-CLOSED(1+7;2+3;Det), OB1	0.0608 (0.0010)	0.0607 (0.0010)	0.0605(0)	15.3 (0.4)
	S-CLOSED(5), OB1	0.0606 (0.0010)	0.0606 (0.0010)	0.0606 (0.0007)	18.4 (0.4)
	WR(7,7), OB1	0.0607(0.008)	0.0607 (0.008)	0.0607 (0.008)	17.1 (0.4)
6	M-CLOSED(7,2), OB1	0.0814 (0.0030)	0.0818 (0.0037)	-	55.2 (0.4)
	S-CLOSED(18), OB1	0.0815 (0.0015)	0.0816 (0.0015)	-	109.5 (4.1)
	WR(4.5,4.5), OB1	0.0814 (0.0012)	0.0813 (0.0018)	-	75.2 (4.9)
7	M-CLOSED(2+4;2), OB1	0.0659 (0.0013)	0.0876 (0.0020)	-	30.9 (0.8)
	S-CLOSED(8), OB1	0.0657 (0.0008)	0.0878 (0.0012)	-	52.2 (1.3)
	WR(11,11), OB1	0.0658 (0.0016)	0.0876 (0.0020)	-	41.4 (2.5)
8	M-CLOSED(4,3+7), OB1	0.1236(0.0030)	0.1233 (0.0028)	-	27.3(0.4)
	S-CLOSED(15), OB1	0.1231(0.0007)	0.1231(0.0007)	-	57.2 (0.8)
	WR(1,1),OB1	0.1235(0.0011)	0.1229(0.0016)	-	38.1 (3.6)

Table 5: Results for Examples 2-8.

the other rules tested by as much as 40 %. These results indicate that this release policy can be rather effective. However, we have also noted that it is harder to estimate the product mix and throughput when the M-CLOSED rule is used, and hence it might be preferable to use the WR rule unless the M-CLOSED rule results in significantly lower cycle times and if the more complex information system necessary to implement it is available.

Further research should address the following questions:

1. As we note above, it is difficult to estimate the throughput and product mix from multi-chain closed queueing networks. Further research is necessary to develop robust approximations for these measures.

2. Wein and Chevalier (1992) have noted the difficulty of using the WR release rule in conjunction with a due date setting rule in a make-to-order environment where incoming orders are assigned due dates when the system has more than 2 machines. They recommended using the S-CLOSED release rule along with the WBAL sequencing rule. However, the superiority of of the M-CLOSED rule over the S-CLOSED rule in our experiments indicates that its use might lead to better due date setting as well as lower cycle times in a make-to-order environment. Further research is necessary to test the performance of the M-CLOSED rule in this environment.

Acknowledgements:

I wish to thank Professor Lawrence Wein for his helpful comments on an earlier version of this paper. I am grateful to an associate editor and two referees for their detailed comments which greatly improved the paper.

Bibliography

[1] Chevalier, P.B., and L.M. Wein. 1990. "Scheduling Networks of Queues: Heavy Traffic Analysis of

a Multistation Closed Network," Technical Report, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139.

[2] Glassey, C.R., and M.G. Resende, 1988. "Closed-Loop Job Release Control for VLSI Circuit Manufacturing," *IEEE Transactions on Semiconductor Manufacturing,* **1**, 36-46.

[3] Harrison, J.M., and L.M. Wein. 1990. "Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network," *Operations Research* **38**, 1052-1065.

[4] Jackson, J.R. 1963. "Jobshop-like queueing systems," *Management Science*, **10**, 131-142.

[5] Spearman, M.L., W.J. Hopp, and D.L. Woodruff, 1989 "A Hierarchical Control Architecture for CONWIP Production Systems," *Journal of Manufacturing and Operations Management,* **3**, 147-171.

[6] Spearman, M.L., D.L. Woodruff, and W.J. Hopp. 1990 "CONWIP: A pull alternative to kanban," *International Journal of Production Research,* **28**, 879-894.

[7] Spearman, M.L., and M.A. Zazanis. 1992 "Push and Pull Production Systems: Issues and Comparisons," *Operations Research,* **40**, 521-532.

[8] Wein, L.M. 1988. "Scheduling Semiconductor Wafer Fabrication," *IEEE Transactions on Semiconductor Manufacturing,* **1** 115-129.

[9] Wein, L.M. 1990a. "Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Network with Controllable Inputs," *Operations Research,* **38**, 1065-1078.

[10] Wein, L.M. 1990b. "Optimal Control of a Two Station Brownian Network," *Mathematics of Operations Research* **15**, 215-242.

[11] Wein, L.M. 1992 "Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Network with Controllable Inputs," *Operations Research,* **40**, S312-S334.

[12] Wein, L.M., and P.B. Chevalier, 1992. "A Broader View of the Job-Shop Scheduling Problem," *Management Science*, **38**, 1018-1033.