OPTIMAL CONTROL OF TANDEM REENTRANT QUEUES

Ger Koole
Department of Mathematics and Computer Science
Vrije Universiteit
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands

Rhonda Righter
Department of Operations and Management Information Systems
Santa Clara University
Santa Clara CA 95053
USA

Appeared in *Queueing Systems — Theory and Applications* **28**:337-347, 1998

Abstract

We consider optimal policies for reentrant queues in which customers may be served several times at the same station. We show that for tandem reentrant queues the last-buffer-first-served (LBFS) policy stochastically maximizes the departure process.

1 Introduction

Reentrant queues, in which customers may visit the same station several times, are important models for manufacturing systems. For example, in semiconductor manufacturing, in which the circuitry is built up in layers on wafers, a lot of wafers may return to the same station for many, if not all, of its layers. In many assembly systems a single worker or robot may be responsible for more than one machine or task. This increases the system's flexibility since the number of workers may change with changes in demand.

Unlike the classical Jackson networks, reentrant queueing models for manufacturing systems have general arrival processes and service distributions, and the service distribution at a particular station may depend on how many times the customer has already visited the station. In addition, we wish to permit general scheduling policies, including priority policies that depend on the number of customer revisits to the station. Therefore, reentrant queues do not have a product form in steady state, and are much more difficult to analyze than Jackson networks. Earlier attempts at efficient scheduling have focused on heuristics. See for example, Chevalier and Wein(1993), Glassey and Resende (1988), Harrison and Wein (1988), Lozinski and Glassey (1988), Lu, Ramaswamy and Kumar (1991), Wein (1988). Also, recent work has shown that reentrant networks under particular policies may be unstable even when the capacity exceeds the offered load at each station. See, for example, Bramson (1996), Kumar (1995), Lu and Kumar (1991), Kumar and Seidman (1990), and Seidman (1994).

Our results hold for both open and closed networks of reentrant queues. Closed networks are useful for describing systems such as flexible manufacturing systems in which there are a fixed number of pallets always circulating through the system. In these systems when a customer is completely finished its processing and leaves the system a new customer is immediately released into the system. For both open and closed systems customers may visit a station several times, and we assume that on different visits to the same station they are placed in different buffers. A buffer is associated with exactly one station. Service at each buffer is FCFS (first-come-first-served). Servers may fail, with arbitrary failure and repair distributions.

A tandem station is defined to be a single-server station that serves only buffers that are "adjacent" to each other in each customer's route. That is, station j is a tandem station if the buffers can be labeled so that for some i_j and k_j station j serves buffers $i_j, i_{j+1}, \ldots, i_{j+k_j}$, and all customers who complete service at buffer i_{j+l} are routed to buffer i_{j+l+1}, and all customers that arrive to buffer i_{j+l+1} come from buffer i_{j+l}, for $l = 0, \ldots, k_j - 1$. A tandem reentrant system is one in which all stations are tandem stations. Refer to figure 1 for an open system example with five buffers and two stations and a single route for all customers. Customers may have different routes through the system, but all customers that are served by a particular station have the same route through the buffers at that station. Thus, if we think of each tandem reentrant station as a node in a larger network, that larger network can be general in terms of customer routing. We also permit forking, that is, customers departing from a station may split into multiple customers that may be routed to different stations.

We consider the LBFS (last-buffer first-served) policy. Under this policy the server always serves the customer in its last non-blocked nonempty buffer. We show that under a broad range of fairly general assumptions, the LBFS policy at each station stochastically maximizes the departure process from all stations for tandem reentrant systems. That is, it stochastically maximizes $\{D_j(t); j = 1, \ldots, m\}_{t=0}^{\infty}$ where $D_j(t)$ is the number of departures from the last buffer of station j (buffer $i_j + k_j$) by time t, and m is the number of stations. Equivalently, all departures from the stations are jointly stochastically earlier under LBFS than under any other policy. (We use earlier, larger, etc. in the nonstrict sense.) A consequence is that, for open systems, LBFS stochastically minimizes the number of jobs in the system at any time and stochastically minimizes the cycle time, or time between the arrival and departure, for every customer. For closed systems it stochastically minimizes the cycle time, i.e., the time between returns to an arbitrary buffer at an arbitrary station. LBFS also minimizes the cumulative holding cost up to any time as long as holding costs are larger for later buffers. This is a generalization of the work of Johri and Katehakis (1988) who showed that for an open tandem system with a single perfectly reliable server, infinite buffers, Poisson arrivals, and exponential service times that the LBFS policy stochastically minimizes the number of customers in the system at any time. Our proof is also simpler.

Note that since for closed systems one tandem station may serve both customers that have almost completed their processing in the system as well as new customers to the system, the "last buffer" under LBFS for such a server would have customers that have received less total processing than other buffers that that station serves. For example, for the closed system in figure 2 station 1 should serve new customers arriving to buffer 1 whenever possible, and serve customers in buffer 4 only when buffer 1 is empty.

Our results also hold for tandem subsystems of more general monotone reentrant systems. By monotone we mean that for all stations that are not tandem reentrant stations, their operation is such that if arrivals to the buffers of those stations are earlier, departures will also be earlier. (See Righter and Shanthikumar, 1992). The LBFS policy is optimal for all tandem reentrant stations within monotone systems.

2 Basic Model

We first consider systems in which all buffers are infinite so that there is no blocking, and all stations are tandem stations. Service at each buffer is FCFS (first-come first-served). In the basic model preemption and idling are permitted without penalty. That is, servers may switch to the first customer in a different buffer or stop serving at any time. At each station the decision to idle or to serve, and of which buffer to serve, may depend on the states at all of the stations and the past history of the network.

We permit both open and closed networks of tandem stations. For open systems the arrival process may be arbitrary as long as it is independent of the state of the system and it is not possible to have an infinite number of arrivals in a finite time. Customers leaving one station may be routed to one of several different stations, where the routing is either random with fixed probabilities, or deterministic, e.g., round-robin. Thus the routing is independent of the state of the system, the time of the routing, and the customer that is being routed. More generally, a customer leaving a station may "fork" into a random number of customers, each of which is randomly routed to other stations.

There is a single server at each station. Service times may have arbitrary distributions and may depend on the station and the buffer, but they must be independent of the state of the system. Service times may depend on the customer and have dependencies across buffers in the following restricted sense. At the time a customer arrives at a station, its service times at all buffers at the station may become known, but the vector of service times at the station are independent and identically distributed for all arriving customers. Servers may fail and have failure and repair times with arbitrary distributions that may depend on the station but are again independent of the state of the system. They are also independent of the buffer and customer being served at the time of failure. If a server fails during a service we assume that when the server is repaired and next serves the customer that was being served upon failure, the server picks up where it left off on that customer; that is, failures affect services in a preempt-resume manner.

Note that given our assumptions on routing and service time distributions, customers are stochastically identical when they arrive to, or depart from, a station.

Under the LBFS policy, each server always serves the customer that is in the buffer with the largest index among all the buffers for its station. The effect of this policy is that servers "follow" customers from the station's first to its last buffer, consecutively performing all the services for that customer. Thus, idling and preemption will never occur under the optimal policy, so the LBFS policy will still be optimal when preemption and idling are not permitted. Also, all the buffers except the first one (and the release buffer) may be of an arbitrary finite size (of at least one, for the customer in service, to avoid triviality) since stations will not be blocked by down-stream stations, and under LBFS the number of customers in all but the first buffer never increases.

We first introduce the notion of monotonicity for stations and networks. We say a station is monotone if earlier arrivals cause earlier departures. That is, let $A_1, A_2, \ldots$ and $A'_1, A'_2, \ldots$ be two sets of arrival times to the station, and let $T_1, T_2, \ldots$ and $T'_1, T'_2, \ldots$ be the corresponding departure times. The station is monotone if $A_i \leq A'_i$ for all i implies that $T_i \leq T'_i$ for all i. The monotonicity of a network or subnetwork is similarly defined, where we say that the network is monotone if earlier arrivals at all stations implies earlier departures at all stations.

We need the following lemma, which is easily shown, that tandem stations under LBFS, and networks of such stations with infinite buffers are monotone. Note that tandem buffers under FBFS, for example, are *not* monotone.

Lemma 2.1 **(i)** *Tandem stations operating under the LBFS policy are monotone.*

(ii) *A network of tandem stations with infinite buffers operating under the LBFS policy is monotone.*

For (ii), we actually only need for the first buffer at a station to be infinite.

Let us consider a relaxed model in which each station has an infinite "release buffer," and the station has the option of placing a customer (or a batch of customers in the case of forking) that has completed all service at that station in the release buffer rather than immediately releasing it to the next station(s). The customer may be released at any time after it is placed in the release buffer. We will show that the LBFS policy with immediate release is optimal, so the release buffer will not be used under the optimal policy. Hence, LBFS will still be optimal even when there is no release buffer. From now on when we refer to the LBFS policy we shall mean the LBFS policy with immediate release.

Let $D_j(t)$ be the number of departures from the last buffer of station j (into the release buffer) by time t, and let m be the number of stations. We have the following.

Theorem 2.2 *For tandem reentrant systems with infinite buffers the nonidling and nonpreemptive LBFS policy at each station stochastically maximizes* $\{D_j(t); j = 1, \ldots, m\}_{t=0}^{\infty}$ *among all idling and preemptive policies.*

Proof. We assume time is discrete and we have a finite time horizon T, and use induction on T. When $T = 0$ any policy is trivially optimal. Let us suppose that for a horizon of length T the LBFS policy at each station stochastically maximizes the departure process, and consider a horizon of lenght $T + 1$. Let π be an arbitrary policy. If π does not agree with LBFS after time 0 then there is a policy that does that has a stochastically larger departure process by the induction hypothesis. Therefore suppose, without loss of generality, that π does agree with LBFS from time 1 on, but disagrees with LBFS at time 0.

Suppose first that at time 0 some server j is up and under policy π server j serves the first customer in buffer k (let us call it customer k) when there is a customer in buffer l (call the first one in buffer l customer l), where buffer l is server j's last nonempty buffer. Since π agrees with LBFS after time 0, under π server j will serve customer l the next time the server is up (at time σ say).

Define a new policy, π_j, that serves customer l at time 0 at server j, serves customer k at server j at time σ, and otherwise agrees with π for services and releases at all stations. We couple the external arrival process (for open systems) and all failure, repair, and service times, so that they are the same under both policies. With these couplings we have that all departures from server j of all customers besides customer l are at the same times under both policies. If customer l is in the last buffer at server j and from time 0 it requires only one more time unit of processing to complete, its departure will be earlier under π'_j than under π, otherwise it will be at the same time. In the former case let π_j keep customer l in its release buffer from time 1 until the time it would be released under policy π, at time $\sigma + 1$. Then all departures at all other stations will be the same under both policies.

We repeat the argument above for each server that does not serve its last nonempty unblocked buffer under policy π, constructing a policy π' that agrees with π_j at each server j that does not serve the last nonempty buffer under π, and agrees with π at all the other stations. Then π' will have a stochastically larger departure process than π. By the induction hypothesis we can construct a policy π'' that agrees with π' for its first decision and agrees with the LBFS policy thereafter such that π'' has a stochastically larger departure process than π' and therefore than π.

Similarly, if under policy π'' there are servers that idle at time 0, we can construct a policy π''' with a stochastically larger departure process that agrees with LBFS from time 1 on and such that at time 0 all servers serve the customer in their last nonempty buffer and do not idle.

Note that π''' agrees with LBFS except for possibly not releasing completed customers from some stations' release buffers at time 0. Since there is no blocking and customers are stochastically identical upon their arrival to a station, releasing all customers immediately will result in earlier arrivals at all stations. Since a network of tandem stations with infinite buffers under LBFS is monotone by lemma 2.1 the departures from all stations will be earlier than under policy π''' if all completed customers are released immediately and LBFS is followed. $\Box$

Note that our result is still true even if there are other, non-tandem stations in the network, as long as the subnetwork of non-tandem stations is monotone and the tandem stations are not blocked by them. In this case the combined network, with LBFS at tandem stations, will still be monotone. The service discipline at all non-tandem stations is fixed and hence cannot be controlled.We make the same assumptions as before about customer routing. Henceforth we permit such mixed networks with both tandem and non-tandem stations.

3 Non-FCFS Disciplines

If we relax the assumption that services at each buffer must be FCFS, so customers may "overtake" each other, then the LBFS policy may no longer be optimal. For example, it may be optimal to serve a customer, customer C, when another customer, customer D, is in a later buffer of a tandem station, if customer C is known to have short service times at all the remaining buffers while customer D has long service times. Even when all customers have the same service time distribution for each buffer and service times across buffers are independent, LBFS may not be optimal. For example, if a server has a single customer in its last buffer that it has been serving for a long time, and the remaining service time is getting larger in some stochastic sense, it may be optimal to stop serving that customer and serve one in an earlier buffer through all of its remaining services until it leaves the last buffer of the server. However, if service times across buffers are independent, and service times for a buffer have the same ILR (increasing in likelihood ratio) distribution, then as a customer is served its remaining service time is getting smaller in the likelihood ratio sense, and there is an incentive to keep serving that customer rather than switching to a different one in the same buffer. That is, the LBFS policy with FCFS within each buffer will be optimal. Righter and Shanthikumar (1992) have shown that for networks of single server stations with a single buffer and common ILR service times FCFS stochastically maximizes the departure process. Combining the ideas of their proof with the proof above we obtain the following corollary.

Corollary 3.1 *For tandem reentrant stations within monotone systems, with stochastically identical ILR service times at each buffer and independent services across buffers at each tandem station, the nonidling LBFS policy at each server and nonpreemptive FCFS at each buffer stochastically maximizes* $\{D_j(t); j = 1, \ldots, m\}_{t=0}^{\infty}$ *when any customer in a buffer is permitted to be served, preemptively, at any time and idling is permitted.*

Of course, if service times at each buffer are identical and services across buffers are independent and preemption is not permitted, then all service disciplines at a buffer are equivalent, and we may as well assume a FCFS discipline.

4 Systems with Finite Buffers and Blocking

We now suppose that buffers may be finite so that there may be blocking. External arrivals to blocked stations are assumed to be lost. Otherwise, our assumptions about the FCFS service discipline, preemption and idling, service distributions, and failure and repairs are as in the basic model.

We permit both tandem and monotone stations in our network as long as the combined network with tandem stations operating under LBFS is monotone. Routing disciplines are as in the basic model, except that we assume that if a station, station A, can be blocked by another station, station B, then station A must be the only station that routes customers to station B, although station A may route customers to other stations besides B. Station A may be an external source of arrivals. This assumption is required for monotonicity as the following example shows. Suppose station B can block two stations, A and C. Then, since A is monotone, an earlier arrival to station A will result in an earlier (potential) departure from station A. This in turn may cause an earlier arrival to station B which may block station C, causing a later departure for station C, violating monotonicity.

We also assume that for each tandem station, either the first buffer is infinite, or all the buffers at the station are shared. If there is a finite shared buffer for a station, we still think of customers on different visits to the station as being in different buffers, but there is an upper bound on the total number of customers at the station rather than individual upper bounds for each buffer. We may also have a finite shared buffer for all the stations. If we have such a shared buffer and the network is closed, we have a CONWIP (constant work-in-process) system and there is really no blocking. For open systems with a single finite shared buffer only external arrivals are blocked. We permit both communication or manufacturing blocking, and we may have different blocking mechanisms for different buffers. In communication blocking a server cannot begin service on the first customer in one buffer until there is space for it in the next buffer (or at the next station), whereas in manufacturing blocking a server can serve the first customer in a buffer, buffer B say, but when service completes the customer cannot move to the downstream buffer (or station) until there is space for it, and the server cannot serve another customer from buffer B until that customer moves. A server is not considered blocked until all the buffers at its station are blocked. The assumption of either an infinite first buffer or a finite shared buffer means there is no incentive to serve earlier buffers in order to make room for new customers.

For tandem stations with finite individual buffers that may be blocked by a down-stream station, it may not be possible to serve the customer in the last non-empty buffer. Now by LBFS, we mean that servers at tandem stations should always serve the customer in the last non-empty and non-blocked buffer. This discipline will no longer be non-preemptive when preemption is permitted, because the service of a customer at an up-stream buffer will be interrupted if there is a customer in a downstream buffer that becomes unblocked. It will still be non-idling.

We have the following.

Lemma 4.1 *Given the assumptions above, a network of tandem and non-tandem monotone stations will be monotone if the tandem stations follow the LBFS policy.*

4.1 Optimal preemptive control

We first suppose that we are permitted to interrupt service at any time.

Theorem 4.2 *For tandem reentrant stations with finite buffers within monotone systems the nonidling LBFS policy at each station stochastically maximizes* $\{D_j(t); j = 1, \ldots, m\}_{t=0}^{\infty}$ *among all idling and preemptive policies. If all finite buffers are shared within a station for tandem stations, or the system has a single shared finite buffer, the LBFS policy will be nonpreemptive.*

Proof. Our proof is similar to that of theorem 2.2, so we concentrate here on the differences. We again relax the model by supposing that each tandem station has a release buffer and that the station has the option of placing a customer that has completed all service at that station and *is not blocked by the next station* in the release buffer rather than immediately releasing it to the next station(s). If the station has a single finite buffer, the customers in the release buffer are also considered to be in this shared buffer.

Again, we use induction on the finite time horizon T, and we define policy π, server j, and buffers k and l as in the proof of theorem 2.2, except that buffers k and l must not be blocked at time 0, and buffer l is server j's last nonblocked nonempty buffer. Now, even though π agrees with LBFS after time 0, under π server j may serve some other customer besides customer l the next time the server is up. In particular, in the case of individual finite buffers, a customer in a buffer after buffer l that was blocked at time 0 may become unblocked after time 0. However, under π server j will serve customer l (at time σ say) before it serves customer k again.

Define a new policy, π_j, that serves customer l at time 0 at server j, serves customer k at server j at time σ, and otherwise agrees with π for services and releases at all stations. This is possible since between times 1 and σ π will only serve buffers after l, and π_j will be able to serve those same buffers. The rest of the proof is as in the proof of theorem 2.2, with π', π'', and π''' as defined there. □

4.2 Optimal nonpreemptive Control

Now suppose preemption and idling are not permitted at tandem stations with individual finite buffers. If a server fails during the service of a customer then when the server is repaired it must continue serving the same customer. As noted in the previous section, if preemption were permitted there may be times when preemption would be optimal. For this reason, when preemption is not permitted and idling is, it may be optimal to idle. This may occur, for example, when the last buffer is blocked and an earlier buffer is not blocked at a tandem station. Then it might better to wait for the last buffer to become unblocked than to serve a customer from an earlier buffer whose service may take longer than the time for the last buffer to become unblocked. Certainly the very next departure, which under our assumptions must be of the customer that is currently blocked in the last buffer, will be earlier if we wait to serve that customer. We henceforth assume idling, as well as preemption, is not permitted.

We must also assume in this subsection that service times at all buffers of tandem stations with individual finite buffers are identically distributed. If this were not the case, when the last buffer is blocked and idling and preemption are not permitted, it might be optimal to serve an earlier rather than a later buffer if the earlier buffer has smaller service times. The last buffer might become unblocked during the service, so serving a buffer with a small service time would allow the server to switch to the last buffer sooner.

Except for assuming that preemption and idling are not permitted, and that service times at all buffers of tandem stations with individual finite buffers are identically distributed, our assumptions are otherwise as in the previous subsection.

Theorem 4.3 *For monotone networks with tandem reentrant systems, when preemption and idling are not permitted, the LBFS policy at each station stochastically maximizes* $\{D_j(t); j = 1, \ldots, m\}_{t=0}^{\infty}$ *among all nonidling and nonpreemptive policies.*

Proof. Our proof is similar to that of the proof of theorem 4.2. Now we use induction on the number of decision points, where we assume there are a finite number of remaining (potential) decision points, N, and potential decision points occur whenever there is a service completion, an arrival to an empty station, or a server repair completion if the server failed when the station was empty. When $N = 0$ any policy is trivially optimal. Let us suppose that when there are N remaining decision points the LBFS policy at each station stochastically maximizes the departure process, and let the number of remaining decision points be $N + 1$. Suppose the first decision point occurs at time 0. Define policies π and π_j, server j, time σ, and buffers k and l as in the proof of theorem 4.2. For tandem stations that do not have finite individual buffers, the proof is as before, so suppose station j does have finite individual buffers. Then π and π_j may serve buffers after buffer l (nonpreemptively) before time σ.

We couple the external arrival process (for open systems) and all failure and repair times, and all service times except those of customers k and l at server j, so that they are the same under both policies. (The assumption of an infinite buffer for arrivals or a single finite shared buffer permits this coupling.) Since we assume all services by server j have the same distribution, we couple the service time at server j of customer k under π with the service time of customer l under π_j and vice versa. With these couplings we have service completions occuring at the same times at server j under both policies, and all departures from server j of all customers besides customer l are at the same times under both policies. If customer l is in the last buffer at server j its departure will be earlier under π'_j than under π, otherwise it will be at the same time. If customer l is in the last buffer at server j let π'_j keep it in its release buffer until the time it would have been released under policy π (the time at which both customers k and l are completed under both policies). Then all departures at all stations will be the same under both policies.

The rest of the argument follows as before. □

5 Corollaries and Extensions

We have several corollaries that follow easily from the results above. For open systems we define the cycle time of a customer as the time between the customer's arrival to and departure from the system. For a closed system the cycle time is defined as the time between returns to the first buffer of an arbitrary server. In the following corollaries if preemption is permitted, or if the last buffer of a station is infinite or there is a single shared buffer for the station or for the whole system, we assume idling is permitted at that station, otherwise it is not.

Corollary 5.1 *LBFS stochastically minimizes the cycle time for tandem reentrant systems, and for open systems it stochastically minimizes the number of jobs in the system at any time. For open systems in which there is a finite shared buffer for the first station or for the entire system, LBFS stochastically minimizes the number of lost customers.*

Let L be the total number of buffers in the system and let $h_i(t)$ be the holding cost for buffer i at time t. We assume that $h_1(t) \leq h_2(t) \leq \cdots \leq h_L(t)$ for all t. This is typically the case in manufacturing systems in which value is added at each stage.

Corollary 5.2 *If $h_1(t) \leq h_2(t) \leq \cdots \leq h_L(t)$ for all t and the first buffer is infinite then the LBFS policy stochastically minimizes the cumulative holding cost until any time s. If the system is closed, we assume that buffers 1 and L are served by different stations.*

It is easy to see that all of our results also hold for tandem subsystems of more general monotone systems. By monotone system we mean that for all stations that are not tandem stations, their operation is such that if arrivals to the buffers of those stations are earlier, departures will also be earlier. The LBFS policy is optimal for all tandem stations within monotone systems.

Acknowledgements We would like to thank the referee whose comments greatly improved the presentation of our results.

References

Bramson, M. (1996). Instability of FIFO queueing networks. *Ann. Appl. Prob.* 6: 414-431.

Chevalier, P. B., and L. M. Wein (1993). Scheduling networks of queues: Heavy traffic analysis of a multi-station closed network. *Opns. Res.* 41: 73-758.

Glassey, C. R., and M. Resende (1988). Closed-loop job release control for VLSI circuit manufacturing," *IEEE Trans. Semicond. Manuf.* 1: 36-46.

Harrison, J. M., and L. M. Wein (1988). Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Opns. Res.* 38: 1052-1064.

Johri, P. K., and M. N. Katehakis (1988). Scheduling service in tandem queues attended by a single server. *Stoch. Anal. Appl.* 6: 279-288.

Kumar, P. R. (1995). A tutorial on some new methods for performance evaluation of queueing networks. *IEEE J. Selected Areas in Comm.* 13: 970-980.

Kumar, P. R., and T. I. Seidman (1990). Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Trans. Aut. Cont.* 35: 289-298.

Lozinski, C., and C. R. Glassey (1988). Bottleneck starvation indicators for shop floor control. *IEEE Trans. Semicond. Manuf.* 1: 147-153.

Liu, Z., and R. Righter (1997). Optimal scheduling on parallel processors with precedence constraints and general costs. *Prob. Eng. Inf. Sci.* To appear.

Lu, S. C. H., and P. R. Kumar (1991). Distributed scheduling based on due dates and buffer priorities. *IEEE Trans. Aut. Cont.* 36: 1406-1416.

Lu, S. C. H., D. Ramaswamy, and P. R. Kumar (1994). Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. *IEEE Trans. Semicond. Manuf.* 7: 374-388.

Righter, R., and J. G. Shanthikumar (1992). Extremal properties of the FIFO discipline in queueing networks. *J. Appl. Prob.* 29: 967-978.

Seidman, T. I. (1994). First come first serve can be unstable. *IEEE Trans. Aut. Cont.* 39: 2166-2170.

Wein, L. M. (1988). Scheduling semiconductor wafer fabrication. *IEEE Trans. Semicond. Manuf.* 1: 115-130.