Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 170 (2006) 541–566

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

Optimal dynamic assignment of a flexible worker on an open production line with specialists

Linn I. Sennott [a], Mark P. Van Oyen [b,*], Seyed M.R. Iravani [c]

[a] Department of Mathematics, Illinois State University, Normal, IL 61790-4520, USA
[b] Department of Information Systems and Operations Management, Loyola University Chicago, Chicago, IL 60611-2196, USA
[c] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, USA

Accepted 24 June 2004
Available online 18 September 2004

Abstract

This paper models and analyzes serial production lines with specialists at each station and a single, cross-trained floating worker who can work at any station. We formulate Markov decision process models of K-station production lines in which (1) workers do not collaborate on the same job, and (2) two workers can work at the same task/workstation on different jobs at the same time. Our model includes holding costs, set-up costs, and set-up times at each station. We rigorously compute finite state regions of an optimal policy that are valid with an infinite state space, as well as an optimal average cost and the worker utilizations. We also perform a numerical study for lines with two and three station. Computations and bounds insightfully expose the performance opportunity gained through capacity balancing and variability buffering.

Keywords: Control of queueing networks; Labor cross-training; Floating workers; Production flexibility; Markov decision processes

1. Introduction

The development of mass production, most visibly expounded by Henry Ford, gave us the serial production line as an architecture upon which most production systems are built. The Just In Time (JIT) and lean production revolutions have since augmented and redirected our thinking about production lines. Among other things, the JIT philosophy places emphasis on reduced work in process (WIP) and finished goods inventory levels, shorter and cheaper set-ups, shorter cycle times, higher quality, and broadened task assignments and responsibilities for workers. The

* Corresponding author. Tel.: +1 312 915 6667; fax: +1 312 915 8508.
E-mail addresses: sennott@math.ilstu.edu (L.I. Sennott), mvanoye@orion.luc.edu (M.P. Van Oyen), s-iravani@northwestern.edu (S.M.R. Iravani).

0377-2217/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.ejor.2004.06.030

Japanese word *shojinka,* a composition of *sho* (reduce) + *jin* (worker) + *ka* (to change), describes a practice within the Toyota Production System that sought to endow teams with a greater range of skills to achieve agility that would result in equal or greater productivity without additional workers. The investment in cross-training and effective labor coordination mechanisms can yield dividends that include not only greater operational efficiency, improved job satisfaction, and higher quality, but also increased organizational flexibility to deal with unforeseen change. Upton (1995) closely examined the fine paper manufacturing industry in North America and concluded that rather than technological sophistication being the key driver, "Operational flexibility is determined primarily by a plant's operators and the extent to which managers cultivate, measure, and communicate with them".

Some literature has developed a conceptual approach to cross-training. For example, Hopp and Van Oyen (2004) proposes both strategic and tactical level frameworks for the design of effective cross-training architectures. Our focus is on mathematical modeling and insights for the efficient use of cross-trained labor to achieve production agility. Achieving this agility involves two key elements: (1) the issue of skill assignment to workers, and (2) a policy for coordinating workers.

In this work, we consider a serial line in which each workstation is attended by a specialist (i.e. a worker who is not cross-trained). To provide agility with a modest amount of cross-training, our model has only one of the workers cross-trained. This worker can perform every task in the line. It is often the case that a floating worker represents a more experienced worker with a higher level of motivation and a higher wage level. The practice of having a limited number of generalists (or a limited number of cross-trained workers) supporting a larger number of specialists is common in many lines, including paced assembly lines. At Ford Motor Co., such highly cross-trained floating workers are often called "utility" workers. Two common uses of a utility worker are to replace either (i) an absent worker, or (ii) a specialized worker when he or she takes a meal or break. In both cases, the floater actually behaves as a specialist while she or he is working at a station. Here, our focus is on the use of floaters to dynamically respond to congestion, floating between two or more stations. Due to the traditional use of floaters to keep lines running during breaks and meals, we have observed that many managers think of using the floater structure for dynamic line balancing as well. We saw this, for example in a flexible, high product variety lighting manufacturing line. At a Chicago area manufacturer of plastic containers, experienced supervisors fill the floating worker roles. In contrast to minimally skilled operators who perform one function, a floating worker serves in a supervisory role and is responsible for troubleshooting groups of workstations when the line goes down. In summary, we conjecture that the use of utility workers in industry stems from the fact that a modest amount of flexible capacity is often sufficient to reap significant benefits, a conjecture that is supported by our numerical results. Our results give evidence of a potentially dramatic effect of even a small amount of flexibility (in our case, a single utility worker in a line of three or four workers).

A second element of agility, once the skill assignment is fixed, is to design a policy or operating procedure by which cross-trained and specialized workers will interact so as to maximize system performance and effectiveness. An effective policy to coordinate cross-trained workers must carefully match the detailed characteristics of the operation. In this paper, we identify the fundamental limits by analyzing optimal policies in the context of asynchronous flowlines, i.e., those in which WIP is buffered and job transfers from any station to its downstream station are not synchronized with other stations. Variability necessitates capacity buffering to absorb the variability of task/workstation processing times. This buffering has traditionally been manifested through WIP buffers at the input of each workstation. We will see that cross-trained workers can be organized in a worksharing system to achieve a capacity buffer against variability as a tradeoff for reduced WIP buffering.

Our analysis treats K-station serial production lines modeled as open queueing networks, with a numerical analysis of 2- and 3-station lines. Such “open” models are typically associated with make to order (MTO) systems and are sometimes referred to as “demand constrained” as opposed to “capacity constrained” systems. Each station has a dedicated worker, and the system contains one fully cross-trained Flexible Worker or Floater denoted by FW. In addition to the methodological focus of this paper, we construct our models and pursue their analysis to address a number of managerial issues. *Issue 1*: does a limited amount of floater-type cross-training under optimal control yield a significant performance impact (“opportunity”) in terms of WIP, cycle time, and cost? *Issue 2*: how well can a floater stabilize a system, and how does it compare to the more expensive approach of putting two dedicated specialists at each station? *Issue 3*: can we gain insight into how set-up times and set-up costs impact both optimal cost and optimal policy? *Issue 4*: does the floater end up doing an unfair fraction of the work, a characteristic that might discourage or at least complicate the implementation of a floater-type system?

2. Literature survey

We start our literature survey by introducing a number of *characteristic modeling assumptions* made about the operating environment of the system that are particularly helpful in organizing the literature on analytical modeling of asynchronous flow lines with cross-training.

(1) *Degree of collaboration*: By degree of collaboration we define the number of workers that can simultaneously work on the same job and task.
(2) *Number of workers per station*: We identify the number of workers allowed per station. We assume that stations are defined by a unique task that is performed at them, so this assumption relates to the relative cost of labor versus the costliness of equipping each station to allow two or more workers.
(3) *Number of skills per worker*: We roughly describe the number of skills per worker as either full cross-training, partial cross-training, or specialization.
(4) *Amount of WIP*: The amount of WIP is key, ranging from models which set a limit on the amount of WIP (see the CONWIP release policy of Hopp and Spearman, 2000), to models with unlimited WIP per station.
(5) *Walking/set-up times*: The length of the walk times to move to another station may be zero, which is typical, or may be positive and possibly stochastic.

We now refer to the numbered characteristics above to survey the most relevant literature, emphasizing work using stochastic models with at least one worker having two or more skills. We begin by looking at the first assumption, considering a degree of collaboration of at least 2 workers per job. While most models assume no collaboration between workers on any given job, the work of Van Oyen et al. (2001) focuses on systems in which (1) workers may collaborate with a linear speedup of their work rate, (2) tooling is sufficiently inexpensive that any number of workers can work at the same task/workstation at the same time, (3) each worker is fully cross-trained to perform every task on the line, (4) both open systems with possibly finite buffers as well as CONWIP systems are modeled, (5) walk times are zero, and the number of stations is equal to the number of tasks (resulting in a station/worker ratio of at most 1.0). This paper establishes the sample-path optimality of the *expedite* policy in which all workers work on the same job and process it from the start at station 1 to completion at station K. Expedite maximizes throughput in CONWIP systems and minimizes cycle times in open systems. Other papers with collaboration of multiple servers on the same job, (1), include Andradottir et al. (2001) and Mandelbaum and Reiman (1998). Mandelbaum and Reiman (1998) treat open models and focus on analyzing the performance impact of pooled servers on Jackson networks of queues in terms of WIP and cycle time. Andradottir et al. (2001) examine lines with limited buffers in which workers are dynamically reassigned to respond to congestion (with manufacturing blocking being the key concern) so as to maximize system throughput.

We resume our survey with assumption (1) limited to no collaboration, that is, only one worker per job at any time. With assumptions (2) through (5) as given above, Van Oyen et al. (2001) also considers a policy called *Pick and Run* (PR), in which workers operate in parallel with one job per worker. As in the expedite policy, workers work on the same job and process it from the start at station 1 to completion at station K. This practice is also commonly referred to as *craft* production. For closed, CONWIP models, this craft mode provides an optimal system throughput provided there is at least one job per worker at all times. For open models, an example is given with deterministic processing times and a non-Poisson renewal arrival process in which craft production is at least 0.58% suboptimal with regard to cycle time. Although the effectiveness of craft policies in open systems has not been systematically studied, the second author's experience suggests that it will be very difficult to find better policies. In a setting with an equal number of workers and stations, Ahn et al. (1999) analyze a two-station line with two fully cross-trained servers under the assumptions given above, but with assumption (4) focused primarily on a system without arrivals and the issue of how to best clear the jobs originally in the system. They derive properties of an optimal policy to minimize the expected holding cost. Some numerical results suggest that when Poisson arrivals are included, a non-idling policy that gives priority service to either queue 1 or 2 (based on a closed-form condition) can effectively respond to the relative magnitudes of the holding costs applied to queue 1 versus queue 2. Given the limitation of (2), only one worker per station and worker blocking (no passing), Bischak (1996) used simulation to compare the throughput of U-shaped flexible worker manufacturing modules with that of serial lines with one stationary worker per machine, with and without the use of buffers. In particular, that work quantifies the utilization of workers under both a Toyota Sewn Products Management System (TSS) model and a cyclic service policy very similar to craft production. Both Bischak (1996) and Zavadlav et al. (1996) give evidence that only small buffers, or even no buffers, are sufficient to keep workers highly utilized. Also using simulation, Downey and Leonard (1992) studied a line with fewer workers than stations, allowing starved workers to move to the unoccupied station with the greatest workload. This work assumed (1) no collaboration, (2) one worker per station, (3) full cross-training for all workers, (4) finite buffer sizes (which are optimized), and (5) positive walk times.

There is a considerable body of literature that focuses on systems in which the skill/worker assignments are made on the basis of fixed, adjacent zones. For example, worker 1 may be skilled for stations 1 and 2, while worker 2 is skilled for stations 2 and 3. With this arrangement, station 2 is the station at which dynamic worksharing must take place. We begin with work characterized by (1) no collaboration, (2) two workers can work at the same task/workstation at stations shared by two zones and otherwise there is only one worker per station, (3) each worker is partially cross-trained to perform every task within her or his limited zone, (4) WIP in excess of one job per worker is intentionally used as a worker/task allocation trigger, and (5) zero walk times. The number of stations typically exceeds the number of workers, and early papers include McClain et al. (2000, 1992), Ostolaza et al. (1990), Schultz et al. (1998), and Zavadlav et al. (1996). They study a zoned pattern of cross-training across a number of adjacent tasks/stations such that the end stations of zones interior to the line are shared by two zones, which is similar to the Toyota Sewn Products Management System (TSS). The worker allocation rule dynamically uses information on the number of jobs waiting at the overlapping stations to determine which worker is "ahead" of schedule, so that the "ahead" worker will perform the overlapping task in an effort to help out the worker who is "behind" schedule. McClain et al. (2000, 1992) and Ostolaza et al. (1990) employ the notion of using a half-full buffer as the measure of system balance. That is, workers choose jobs to try to keep inter-station buffers half full. Gel et al. (2000) used a combination of MDP's and simulation in developing and testing a near-optimal heuristic called the 50–50 work content heuristic and generalized the Half Full Buffer rule first introduced in Ostolaza et al. (1990). These heuristics have been subsequently analyzed and additional ones developed in Askin and Chen (2002). Hopp et al. (2004) gained managerial insights for CONWIP queueing models of lines with two skill pattern strategies: cherry picking, which seeks to balance a line with a minimal number of skills, and two-skill chaining, which trains each worker for a unique base station and also for the immediate downstream station. They found that two-skill chaining is superior to cherry picking in the presence of process variability, and it is robustly effective across a number of queue-length based policies. Further, there is significant value in completing the chain, even though the line is balanced without cross-training the worker(s) that are based at any bottleneck station(s). In the spirit of zoned worksharing, Gel et al. (2002) used an MDP framework to analyze the hierarchical zoned skill pattern, in which the skills of some workers are subsets of the skills of other workers (a feature that is shared by the floater-type model of this paper). Here, there are two or three tasks in the lines considered, and senior workers will possess all the skills of junior workers. Using sample-path coupling they establish a "fixed before shared" principle, which has the broadly skilled workers give strict priority to the task-types for which only they are trained. In this way, the less-skilled workers are protected from starving for lack of tasks for which they are trained. Iravani et al. (1997a,b) continue in this vein, but they allow (5) positive set-up costs and set-up times, and they assume distinct, non-overlapping zones set in a tandem or a U-shaped line. They carefully studied the effect of set-up costs and walking times, in addition to station-dependent holding costs, by decomposing the line into a number of tandem queues, each attended by a single moving server that used the appropriate amount of job batching so as to limit the set-up penalties. It is shown in Iravani et al. (2002) that when set-up times are insignificant (in compared with job processing times), a near-optimal policy can be obtained only using the first moments of job processing and arrival times.

For the next topic in the literature, we dispense with the notion of allowing multiple workers to work at the same station; rather we assume: (1) no collaboration, (2) at most one worker can work at the same task/workstation at the same time, (3) each worker is fully cross-trained to perform every task of the line, (4) WIP is limited to one job per worker, and (5) walk times are zero. Bartholdi et al. (1999), Bartholdi and Eisenstein (1996), and Bartholdi et al. (2001) are important expositions of the *bucket brigade* policy for coordinating workers, particularly when workers possess distinctly different production speeds. Although distinct from the Toyota Sewn Products Management System (TSS), bucket brigades can be applied in similar environments and in warehouse order picking. The efficient operation of a bucket brigade system depends in part on its effective use of task preemption, because workers in that system frequently take over jobs initiated by another worker. For this to be efficient, it is key that the number of stations exceeds the number of workers. For example, Toyota intentionally trained workers to be able to handoff a piece being sewn without stopping the sewing machine or losing quality. McClain et al. (2000) carefully considers this class of models (with and without job preemption) and examines a variety of policies including the bucket brigade and a very effective variation on that theme which allows workers to drop jobs midstream, thereby using WIP to buffer variability.

Farrar (1993) considers a two-station line similar to ours; however, the FW has a distinct service rate, and (5) there is no walking or set-up time. The system is assumed to contain an initial number of jobs (extensive use of WIP), no new jobs are allowed, and it is desired to find the policy that minimizes the expected holding cost to clear the system. Some properties of an optimal policy are proved, but it is not numerically determined.

Andradottir et al. (2003) wrote a paper concurrently with this one, and it addresses the system capacity through stability analysis of queueing networks with flexible servers and probabilistic routings. They provide an upper bound for system capacity and show how generalized round-robin policies may be constructed to yield a capacity arbitrary close to the maximal capacity. A noteworthy feature of their work is the fact that they permit generality in four of the five characteristics modeling assumptions—the exception being that their model treats only open queueing systems.

This paper considers Markov models of K-station lines under the assumptions of: (1) no worker collaboration on the same job, (2) two workers can work at the same task/workstation on different jobs at the same time, (3) one *floating*, flexible worker is fully cross-trained to perform every task of the line, while there is also one specialist dedicated to each station on the line, (4) WIP in excess of one job per worker is used to buffer variability, and (5) both zero and positive set-up times are allowed. In addition to set-up times, we include holding costs and set-up costs at each station. In terms of the skill assignment pattern, this model can be viewed as an extension of the hierarchical skill assignment pattern studied for different models in Gel et al. (2002). In addition to gaining intuition and insight into the important, yet overlooked floater approach to cross-training, this paper seeks to be especially careful in its use of numerical MDP computations.

3. First model: FW-1

Model FW-1 describes a serial production line with $K+1$ workers and K stations or stages labeled $1, 2, \ldots, K$, each having an infinite buffer to hold jobs. The stations may be arranged in a line, a U-shape, or another convenient facility layout. At each station there are two single-person workstations with one of the workstations permanently attended by a specialized worker and the other available to the FW as needed. The time to complete a job on a workstation at station k is exponentially distributed with rate μ_k, regardless of whether the specialist or the FW is working there. FW-1 models an open (push) production system with jobs arriving according to a Poisson process with rate λ.

If the FW moves to station k, she or he may begin to serve a second job at station k provided that a second job is present at that station. The FW is not allowed to collaborate on the same job with any specialized worker, and if there is only one job at a station, we assume that the specialist, not FW, works on it. The time to service a second job at station k is also exponentially distributed with rate μ_k. We may think of this assumption as modeling either identical workers or tasks with machine-dependent (as opposed to worker-dependent) service rates.

In the models in this paper, we consider situations where the FW must always reside at one of the stations, the one that is either already set up or being set up. In the first model, FW-1, there is no set-up time or cost for the FW to move from one station to any station; rather, this additional complexity is introduced in model FW-2 through the addition of a set-up time (that includes walking) and a switchover cost. We begin with model FW-1 since it has a more transparent model structure and its cost will serve as a benchmark (a lower bound) for the cost of more complex model, FW-2.

3.1. The CTMDP formulation of FW-1

We may construct model FW-1 as a continuous time Markov decision process (CTMDP) denoted as Ψ. For the CTMDP, we have:

- *State of the system*: The K-dimensional vector $\mathbf{i}$ denotes the buffer occupancies at each station (buffer occupancies always include jobs undergoing processing). The system is in state $\mathbf{i}$ if the buffer occupancies are given by $\mathbf{i}$ (with i_s denoting the number at station s).
- *Decision epochs*: Decision epochs are epochs when a new job has just entered the system (always entering buffer 1), or a job completion has just occurred at station s and that job has been subtracted from the content of buffer s and added to the content of the next buffer $s+1$, if any.
- *Action space*: Action space includes $a \in \{1, 2, \ldots, K\}$, where choosing a means that the FW will (instantaneously) move to station a (or remain at a, if she or he is presently there).

Our model allows job preemption, meaning that if the FW is currently processing a job and chooses to move to another station, then that job is preempted and will be started over (or be resumed with the same exponential distribution on remaining process time) later as determined by the policy. Throughout, the variable s denotes a generic station and, unless otherwise indicated, summation takes place over all the stations. There is a holding cost rate $H_s > 0$ charged for each unit of time that a job is at station s. If the current state is $\mathbf{i}$ and action a is chosen, then the cost rate is given by $g(\mathbf{i}, a) = \sum_s H_s i_s := g(\mathbf{i})$. There are no instantaneous costs. The time that elapses before a transition to another state is exponentially distributed with rate

$$v(\mathbf{i}, a) = \lambda + \sum \mu_s I\ (i_s \geqslant 1) + \mu_a I\ (i_a \geqslant 2), \tag{1}$$

where $I(E)$ is the indicator function for event E. The first term on the right hand side (1) is the arrival rate. The second term is the total service rate due to the specialized workers, and the last term is the service rate due to the FW.

For computational purposes, we now transform Ψ into a (discrete-time) Markov Decision Process (MDP), Δ, using uniformization. This technique was introduced in Schweitzer (1971), and a complete and detailed reference for our treatment is Sennott (1999, pp. 241–248).

The state space for Δ is the same as that for Ψ. Now choose and fix τ, the uniformized system transition period (in units of time per transition), satisfying

$$\tau = \left(\lambda + \sum \mu_s + \max\{\mu_s\}\right)^{-1}. \tag{2}$$

Assume that the system is in state $\mathbf{i}$ at the beginning of a slot and that decision a is made. At the beginning of the next slot, the system will transition to $\mathbf{i} + \mathbf{e}_1$ with probability $\tau\lambda$, will transition to $\mathbf{i} - \mathbf{e}_s + \mathbf{e}_{s+1}$ with probability $\tau\mu_s[I(i_s \geqslant 1) + I(a = s, i_s \geqslant 2)]$, and from (1) will remain in $\mathbf{i}$ with probability $1 - \tau v(\mathbf{i}, a)$. We define $\mathbf{e}_{K+1}$ to be the zero vector to handle the case of job completion at the last station. The cost incurred at the beginning of a time slot is $C(\mathbf{i}, a) = g(\mathbf{i})$.

Because of the difficulty of this class of models, we must rely on numerical calculations. The key problem here is that Δ has infinite buffers, and we must find a way to implement our computations rigorously with only a finite state space. We employ the *Approximating Sequence* (AS) method introduced in Sennott (1999) to replace Δ with a sequence (Δ_N) of finite state space MDPs in which computation takes place.

The state space of Δ_N consists of vectors $\mathbf{i}$ such that $i_s \leqslant N$ (a maximum buffer size of N jobs). The AS method is not a naive truncation scheme—we redistribute "excess probability" that causes the system to go out of bounds, rather than throwing it away. The system can only escape Δ_N from a boundary state for which some buffer is full. If $i_1 = N$, an overflow would occur in buffer 1 if a new job arrives, which happens with probability $\tau\lambda$. This excess probability is assigned to the state $\mathbf{i}$, effectively denying entry to the new job and holding station 1 at buffer level N until it completes a job. Similarly, if $i_s = N$, for some s, $2 \leqslant s \leqslant K$, then with probability $\tau\mu_{s-1}[I(i_{s-1} \geqslant 1) + I(a = s-1, i_{s-1} \geqslant 2)]$ the next state is $\mathbf{j}$, where $j_{s-1} = i_{s-1} - 1$, and all other coordinates of $\mathbf{j}$ equal those of $\mathbf{i}$. The effect of this assignment is to remove the completed job at $s-1$ from the previous buffer and discard it. The one stage costs in Δ_N are the same as those in Δ.

To summarize, we have reduced the CTMDP Ψ modeling FW-1 to the associated discrete-time infinite state space MDP Δ, which we then approximated by the finite state space sequence (Δ_N) for computation.

3.2. Optimality, stability, and bounds for FW-1

The objective is to determine a dynamic policy that achieves the minimum long run expected average cost per unit time (denoted *average cost*, for short). There are a number of equivalent ways to define stability for the scope of this paper. In particular, we take stability to mean that the system has a finite time-average (system) queue length. When the model includes a cost formulation, this definition is revised to require a finite time-average system cost. Assume that Ψ is controlled under policy θ. Let $F(t)$ be the total cost incurred during the interval $[0, t)$. For initial state $\mathbf{i}$, the average cost under θ and the optimal average cost, respectively, are defined as

$$J_\theta^{(1)}(\mathbf{i}) = \limsup_{t\to\infty} E_\theta^{(\mathbf{i})}\left[\frac{F(t)}{t}\right], \tag{3}$$

$$J^{(1)}(\mathbf{i}) = \inf_\theta J_\theta^{(1)}(\mathbf{i}). \tag{4}$$

A stationary policy (as opposed to a randomized policy or a time-varying policy) makes the same decision in state **i** every time state **i** recurs. It can be shown that our model is ergodic under our stability conditions (see Theorem 2), so the average cost will be independent of the initial state. To summarize, we seek a constant $J^{(1)}$ and a stationary policy f such that

$$J^{(1)} \equiv J_f^{(1)}(\mathbf{i}) = J^{(1)}(\mathbf{i}) \quad \text{for all } \mathbf{i}. \tag{5}$$

In an open system that is stable, the throughput (output) rate under any stationary policy equals the input rate λ. In the special case $H_s \equiv 1$, then $J^{(1)}$ equals the minimum average number of jobs on the line, and an optimal policy guarantees minimum average WIP. Little's Law shows that, in this case, an optimal policy also minimizes the average time a job spends in the line, i.e. cycle time (CT), where the details can be found in Gross and Harris (1998) or Hopp and Spearman (2000).

To gain insight and to demonstrate the existence of an optimal stationary policy, our first step is to define and to understand the stability region. Let $r_s = \lambda/\mu_s$, which measures the offered load at station s, and let $q_K = \sum_{k=1}^{K} r_k = \sum_{k=1}^{K} \lambda/\mu_k$ which can be interpreted as the offered load for the entire system. Implicit in our stability definition is that a system is stable if, and only if, every queue is stable. An intuitive conjecture for the stability condition is simply $q_K < K+1$ and $r_s < 2$ for all s. This makes sense by requiring the offered load on the entire system to remain within the capacity of $K+1$ workers, and the offered load on each station remains within the capacity of two workers. However, this is not quite correct. Consider a 4-stage line with $r_1 = 1.6$, $r_2 = 1.8$, $r_3 = 0.1$, and $r_4 = 0.1$. Here $r_s < 2$ for all s and $q_K < 4+1$; nevertheless, the system is not stable because FW cannot compensate for the excess workload at stations 1 and 2 given that the specialists at stations 3 and 4 are underutilized. To correct this, define set $\mathscr{S}$ as the set of indices of the queues with $r_s \geqslant 1$, and let $|\mathscr{S}|$ indicate the number of queues in that set. In addition to a stability condition, it is important to identify a policy that stabilizes the system. We focus on the Longest Queue (LQ) policy which in many cases is easy to implement and requires no complicated setting of parameters. At any decision epoch, define the LQ policy to assign the floater to the longest queue (excluding any job in service, because the specialists are always given preemptive priority when only one job is at a station). If two or more queues tie for the longest, FW is assigned to the furthest downstream of these.

Assumption A

$$q < |\mathscr{S}| + 1, \tag{6}$$

where

$$q := \sum_{s\in\mathscr{S}} r_s = \sum_{s\in\mathscr{S}} (\lambda/\mu_s) \quad \text{and}$$

$$\mathscr{S} := \{s | r_s \geqslant 1 \text{ and } s \in \{1, 2, \ldots, K\}\}.$$

Theorem 1. *Model FW-1 can be stabilized by the Longest Queue (LQ) policy if Assumption A holds. Moreover, FW-1 cannot be stabilized by any policy if $q > |\mathscr{S}| + 1$.*

The Proof of Theorem 1 and all our results are given in Appendix A. LQ is very effective at stabilizing the system (although it is not specifically tailored to holding cost minimization), because it uses state feedback to dynamically allocate the worker without any need to explicitly know the system parameters (as in the case of the sophisticated generalized round-robin policy of Andradottir et al., 2003). If any queue is neglected, the rising queue length triggers the FW to allocate more effort there.

The following theorem, our main result, shows that an optimal stationary policy for the continuous time, infinite state model Ψ exists, and an optimal control policy and cost may be accurately determined by computations using value iteration on Δ_N, the discrete-time, finite state space model.

Theorem 2. *Assume that Assumption A holds. Then, the average cost in Δ_N is a constant J_N and an optimal stationary policy e_N may be computed using value iteration. Moreover, $J_N \to J^{(1)}$ and the limiting policy of the e_N, denoted by e, is optimal for Ψ.*

3.3. Value iteration equations

The (relative) value iteration equations for calculation of an optimal policy in Δ are given below, and we now show how to modify them for Δ_N. We follow the notation of Sennott (1999, 6.6.4) or see Puterman (1994, 8.5.5).

The value iteration algorithm involves sequences $u_n(\mathbf{i})$ and $w_n(\mathbf{i})$. We begin by setting $u_0 \equiv 0$ as an initial guess. Choose the zero vector $\mathbf{0}$ as our base point or reference state and define

$$\begin{aligned} w_n(\mathbf{i}) = g(\mathbf{i}) &+ \tau\lambda u_n(\mathbf{i}+\mathbf{e}_1) \\ &+ \tau\sum \mu_s I(i_s \geqslant 1) u_n(\mathbf{i}-\mathbf{e}_s+\mathbf{e}_{s+1}) \\ &+ \min_a\{\tau\mu_a I(i_a \geqslant 2) u_n(\mathbf{i}-\mathbf{e}_a+\mathbf{e}_{a+1}) \\ &\qquad + (1-\tau v(\mathbf{i},a)) u_n(\mathbf{i})\}. \end{aligned} \tag{7}$$

Update by setting $u_{n+1}(\mathbf{i}) = w_n(\mathbf{i}) - w_n(\mathbf{0})$.

Modifying Δ to create Δ_N is easily accomplished as follows. If $i_1 = N$, then we replace e_1 in (7) with the zero vector, so that the new job is turned away. The service completion cases are handled similarly, so that a job is eliminated whenever its transfer causes a queue to overflow.

The programs to numerically compute the MDP are set up to compute using (7), modified for Δ_N. It is the case that $w_n(\mathbf{0}) \to J_N$ and the stationary policies (dependent on n) realizing the minimum in (7) converge to e_N, an optimal policy for Δ_N. If desired, one may then recompute for larger values of N and use Theorem 2 to determine $J^{(1)}$ and an optimal policy for Ψ. It is usually the case that an optimal policy is already determined for relatively small values of N.

We now prove a scaling result, so that our experimental investigation of the problem is simplified.

Theorem 3. *Let b and c be positive constants. Consider the modified system model $(*)$ with parameters $\lambda^* = b\lambda$, $\mu_s^* = b\mu_s$, and $H_s^* = cH_s$ for $s = 1, 2, \ldots, K$. An optimal policy for the original system computed using VI is also optimal for $(*)$. The average cost for $(*)$ is equal to $cJ^{(1)}$.*

Intuitively, the parameter b scales the time unit, which does not affect an optimal policy. The parameter c scales the monetary unit, which results in multiplying the average cost by c. Theorem 3 allows us to assume, in all our calculations, that $\lambda = 1$. Because of Assumption A, we then require that $\mu_s \leqslant 1$. On a production line, it is usually the case that the holding cost increases or stays the same as a job moves down the line and value is added to the product. Hence, we will assume that H_s is an increasing (not necessarily strictly) sequence of positive numbers with $H_K = 1$.

3.4. Simple models of FW-1 for benchmarking and insights

One way to gain insight into the importance of effective system control is to develop simple, closed-form models that approximate the behavior of the system and also quantify the behavior of other, related systems. In addition to simple formulas, these models provide us some technical benefit in assuring that the optimal average cost is finite and can be approximated using our methods. On the other hand, we will see evidence that these simple models are very rough approximations of what a properly controlled flexible worker can achieve. We present two models: The *division* model is based on a "division" of the effort of the FW, while the *PR rule* model is based on a "splitting" of the arrival process and the use of the Pick and Run policy for the FW.

3.4.1. The division model

To begin, consider a single queue with two servers, one serving at rate μ and the other at rate $\alpha\mu$, where $0 \leqslant \alpha \leqslant 1$. When a job arrives to an empty system, it is handled by the faster server. Servers never idle when jobs are available. When there are two jobs in the system and the faster server finishes first, he or she will preempt the slow server. Denote this queue by $M/M/(1+\alpha)$, which we characterize as follows:

Theorem 4. *Consider an $M/M/(1+\alpha)$ queue with utilization factor $\rho = \lambda[\mu(1+\alpha)]^{-1} < 1$. This queue is stable with average number L of jobs in the system, where*

$$L = \frac{(1+\alpha)\rho}{(1+\alpha\rho)(1-\rho)}. \tag{8}$$

A division model divides FW into multiple slower workers operating independently at each station. This division is specified by a K-dimensional vector $\bar{\alpha}$ with positive entries summing to 1 (the total effort of the FW is 1). Allocating α_s fraction of effort of FW permanently to station s, station s becomes an $M/M/(1+\alpha_s)$ queue. It follows by the reversibility of any birth–death process that the steady-state distribution has product form. It follows from (8) that the average cost under this division is given by

$$J_{\bar{\alpha}}^{(1)} = \sum H_s r_s \left[\left(1 + \frac{\alpha_s r_s}{1+\alpha_s}\right)\left(1 - \frac{r_s}{1+\alpha_s}\right)\right]^{-1}. \tag{9}$$

Having characterized the general form of a division, we now emphasize a particular allocation of FW, $\bar{\alpha}$, which serves as our benchmark division model and will be needed for model FW-2.

Theorem 5. *If a stable division exists, then Assumption A holds. Further, if Assumption A holds, then defining $\alpha_s = r_s - 1 + \varepsilon_s$, and letting $\varepsilon_s := [|\mathscr{S}| + 1 - q]/|\mathscr{S}|$ for $s \in \mathscr{S}$, and $\alpha_s = 0$ for $s \notin \mathscr{S}$, the division model using vector α is stable.*

Since we cannot divide the effort of the FW at the same instant in time, unfortunately a division model does not correspond to a policy that can be implemented in practice with a single floater. However, if we allow a time-dependent and state-dependent policy, we can approximate the division arbitrarily closely. The server must preemptively "chatter", visiting each queue in a manner approximating the "shared processor" structure of the division. In addition, a division can be given an interesting economic interpretation. Assume that the FW earns an hourly wage of W. Instead of the original model, imagine a system consisting of the original K dedicated workers plus $|\mathscr{S}|$ additional slow workers. At station s, place a slow worker who will serve at exponential rate $\alpha_s \mu_s$ and earn $W\alpha_s$ per hour. Under the economic interpretation, the result will be a division with average cost given by (9), and the personnel wage cost will be the same as with the single FW.

3.4.2. The pick and run (PR) model

Next, we present the PR rule benchmark, which decomposes the system into two parts. One is a series of $M/M/1$ queues, each attended by one specialist. The other is the FW attending the set of secondary machines as an isolated queueing system (call it the FW queue). The FW queue will handle a fixed proportion p of the jobs (implemented by means of a randomized splitting of the arrival process, so that the arrival stream to the FW queue is a Poisson process with rate $p\lambda$). The jobs sent to the FW will queue at the first machine. The FW will take one of these jobs and process it at each of the machines before returning to the first machine and taking on another job, as in the Pick and Run policy of Van Oyen et al. (2001). It is readily seen that the FW queue is an $M/G/1$ queue. The specialized workers will handle $(1-p)$ fraction of the jobs, and they form a tandem series of independent $M/M/1$ queues. The two systems are independent. Let us call this a *Pick and Run* (PR) rule system approximation.

Let μ be the smallest (i.e., bottleneck) service rate, and let $r = \lambda/\mu$. Thus, $r \geqslant r_j$ for $j = 1, 2, \ldots, K$.

Assumption B

$$q_K^{-1} > 1 - r^{-1}, \tag{10}$$

where

$$q_K = \sum_{k=1}^{K} r_k = \sum_{k=1}^{K} (\lambda/\mu_k) \quad \text{and} \quad r = \lambda/\mu.$$

Theorem 6. *There exists a stable PR rule for model FW-1 if, and only if, Assumption B holds. Moreover, when p satisfies $1 - r^{-1} < p < q_K^{-1}$, the average cost of the entire system under PR is given by*

$$J_p^{(1)} = \sum_{s=1}^{K} \left(\frac{H_s(1-p)r_s}{1-(1-p)r_s} \right) + p\sum_{s=1}^{K} H_s r_s + \frac{H_1 p^2 \left(q_K^2 + \sum_{s=1}^{K} r_s^2\right)}{2(1-pq_K)}. \tag{11}$$

In the above, setting p such that $1 - r^{-1} < p$ ensures that the specialists can stabilize their line, while $p < q_K^{-1}$ ensures that FW can stabilize the PR line. The next result indicates that the division model stabilizes a broader range of systems than does the PR rule. When the specialist line has a bottleneck that limits flow, the PR model does not allow full utilization of the specialists at the non-bottleneck stations.

Theorem 7. *Assumption B* (*the PR rule's stability condition*) *implies Assumption A* (*the stability condition for the division model*), *but the converse fails. If all the station service rates are equal, then these conditions are equivalent.*

Throughout our numerical results, we compute (by numerical search) the value of p that minimizes the expression in (11), and we refer to the resulting value of (11) as the PR upper benchmark.

3.4.3. M/M/2 lower bound

To obtain a lower bound, we relax the constraints on worker placement. Assume that, instead of a single FW, there is a second worker permanently assigned to each station (for a total of $2K$ workers). This is a tandem $M/M/2$ system, and clearly, model FW-1 can never perform better than this. The average cost under this scheme is the lower benchmark (LBM). Using (8) with $\alpha = 1$ yields

$$\mathrm{LBM} = 2\sum_{s=1}^{K} \frac{H_s \rho_s}{1 - \rho_s^2}; \quad \text{where } \rho_s = \lambda/2\mu_s. \qquad (12)$$

3.5. Numerical results for FW-1

The value iteration equations (7) were implemented for 2 and 3 stations under Assumption B, so FW must assist at every station. Table 1 gives results for $K = 2$, while Table 2 treats $K = 3$. For both tables, the fourth column gives the average cost under the lower benchmark (LBM) from (12). The fifth column, $J^{(1)}$, gives the average cost under an optimal policy (OP). The sixth column gives two upper benchmarks (UBM's). First we show in parentheses the average cost under the best PR policy and below it that of the division model (DIV) as specified in Theorem 5. Although LBM and the two UBM's can be viewed as rough approximations of $J^{(1)}$, they are more useful in interpreting and measuring the opportunity of optimal cross-training compared with alternate approaches, thereby addressing Issues 1 and 2 of Section 1. To address Issue 1 (does a single floater under optimal control yield a significant opportunity for cost savings?), PR considers a suboptimal, yet implementable, approach that uses binomial splitting to direct some of the arrivals to FW, who then follows a craft (PR) policy. Hence, (PR – $J^{(1)}$) is a measure of the value of optimal dynamic scheduling of FW. Compared to PR, DIV almost always provides a better bound, and (DIV – $J^{(1)}$) represents the replacement of the FW with multiple dedicated (and appropriately slower) workers. An advantage of PR relative to DIV is that PR dynamically controls the fast, flexible server in a suboptimal, but fairly effective way. On the other hand, PR suffers because it has to simplistically split the arrival process between the specialists and FW. The fact that DIV is a better benchmark than PR shows that the benefit gained in the PR system by a flexible, fast server does not outweigh the loss suffered by having to split the arrival process in the PR model. LBM models a traditional approach with two workers at each station (which performs better at the cost of an additional worker). The difference ($J^{(1)}$ – LBM) quantifies the additional cost of having a single floater (Issue 2 of Section 1).

The remaining columns give calculations relating to an optimal policy. The seventh column gives the average number of jobs at each station and on the line (the sum of the other two numbers), under OP. If $H_1 = 1$, then the latter number equals $J^{(1)}$. The eighth column gives the utilization factor (proportion of time spent serving) of each specialized worker, and the ninth column gives the utilization factor of the FW at each station, as well as the total time spent serving by the FW (the sum of the other two numbers).

To calculate the entries in columns seven through nine, there is no need to find the steady-state probabilities under OP. Rather the loop that calculates (7) can be re-used, with two differences. First, the minimization is replaced with the action

Table 1
Results for model FW-1 with $K = 2$, $H_2 = 1$, and $\lambda = 1$

Scenario	H_1	μ_1	LBM	$J^{(1)}$	UBM	Avg. # jobs	Util.	Util. FW
		μ_2			(PR)	1	1	1
						2	2	2
					DIV	Line		Total
1	1.0	0.75				6.01	0.90	0.44
		0.75	4.80	9.10	(22.12)	3.09	0.89	0.45
					15.73	9.10		0.89
2	1.0	0.90				2.47	0.78	0.33
		0.90	3.21	4.04	(8.00)	1.57	0.74	0.37
					6.18	4.04		0.70
3	1.0	0.70				4.76	0.89	0.54
		0.90	4.52	7.18	(21.23)	2.42	0.82	0.30
					11.14	7.18		0.84
4	1.0	0.90				4.01	0.84	0.27
		0.70	4.52	6.64	(21.33)	2.63	0.89	0.56
					10.96	6.64		0.83
5	0.5	0.75				6.85	0.91	0.42
		0.75	3.60	5.90	(14.96)	2.47	0.87	0.47
					11.30	9.32		0.89
6	0.5	0.70				5.55	0.91	0.42
		0.90	3.07	4.64	(12.11)	1.87	0.78	0.33
					7.84	7.42		0.85
7	0.5	0.90				4.47	0.85	0.26
		0.70	3.72	4.52	(16.13)	2.29	0.85	0.57
					8.03	6.76		0.84
8	0.5	0.80				5.03	0.88	0.37
		0.80	2.56	2.95	(7.07)	1.69	0.79	0.46
					5.57	6.72		0.83

under OP. This converts the value iteration algorithm to a standard successive substitution performance evaluation algorithm. Second, the objective function is changed to reflect the desired quantity. For example, to calculate the average number of jobs at the first station, we replace $g(\mathbf{i})$ by i_1. The other quantities are calculated in a similar way.

Our experiments suggest that in general an optimal policy is characterized by a switching curve. Figs. 1 and 2 give these for scenarios 1, 6, and 7. For a fixed number of jobs at station 1, the switching curve is the minimum number of jobs that must be present at station 2 for the optimal policy (OP) to place the FW there; otherwise the FW must work at station 1. The large dot is the point corresponding to the average number of jobs at each station. As jobs enter and are processed, the queue lengths will cross and re-cross the switching curve, necessitating movement of the FW.

Consider scenario 1. Since $H_1 = 1$, we are minimizing average WIP (or CT). Under OP, on average there will be 9.1 jobs on the line, as opposed to 4.8 jobs under the four worker system and 22.12 jobs under PR. Note that although a policy similar to PR was shown to be effective in systems with full cross-training for all workers (Van Oyen et al., 2001), here we observe that the PR policy does not perform well when there are specialists on the line.

Recall condition (9). Since $q_K^{-1} + r^{-1} = 1.125$, the FW will be pushed to stabilize the line. Indeed, each worker is busy roughly 90% of the time. Under the LBM (using four permanent workers),

Table 2
Results for model FW-1 with $K = 3$, $H_3 = 1$, and $\lambda = 1$

Scenario	H_1	μ_1	LBM	$J^{(1)}$	UBM	Avg. # jobs	Util.	Util. FW
	H_2	μ_2			(PR)	1	1	1
		μ_3			DIV	2	2	2
						3	3	3
						Line		Total
1	1.00	0.85				4.62	0.87	0.30
	1.00	0.85	5.40	10.40	(27.89)	3.21	0.87	0.30
		0.85			23.18	2.58	0.86	0.31
						10.4		0.91
2	1.00	0.95				2.76	0.79	0.26
	1.00	0.95	4.37	6.38	(14.05)	1.93	0.78	0.28
		0.95			11.88	1.69	0.76	0.29
						6.38		0.83
3	1.00	0.80				3.96	0.86	0.39
	1.00	0.95	4.96	8.40	(23.74)	2.34	0.81	0.24
		0.95			16.26	2.08	0.80	0.25
						8.40		0.88
4	1.00	0.95				3.51	0.82	0.23
	1.00	0.80	4.96	8.17	(23.74)	2.64	0.85	0.39
		0.95			16.26	2.02	0.70	0.26
						8.17		0.88
5	1.00	0.95				3.43	0.82	0.23
	1.00	0.95	4.96	8.04	(23.74)	2.36	0.81	0.24
		0.80			16.26	2.25	0.84	0.41
						8.04		0.88
6	0.25	0.90				4.92	0.87	0.24
	0.50	0.90	2.81	3.76	(10.00)	2.09	0.81	0.30
		0.90			9.13	1.48	0.76	0.35
						8.49		0.89
7	0.20	0.80				6.43	0.91	0.34
	0.60	0.95	2.74	3.77	(9.07)	1.78	0.78	0.27
		0.95			9.35	1.42	0.75	0.30
						9.63		0.91
8	0.20	0.95				5.25	0.87	0.18
	0.60	0.80	2.98	3.88	(11.79)	2.37	0.84	0.41
		0.95			9.76	1.41	0.74	0.31
						9.03		0.90
9	0.20	0.95				5.13	0.87	0.19
	0.60	0.95	3.22	3.93	(14.51)	2.03	0.78	0.27
		0.80			10.16	1.69	0.80	0.44
						8.84		0.90

the utilization of each worker is only 67%, which indicates the unused labor potential. We observe here that the use of a single floater is an effective alternative to the use of multiple (moderately utilized) specialists.

The average queue length vector shown in Fig. 1 is approximately six jobs in station 1 and 3 in station 2. Despite this difference, Table 1 shows that the FW is busy almost equally at 1 and 2. This gives insight into Issue 4 raised in Section 1—does

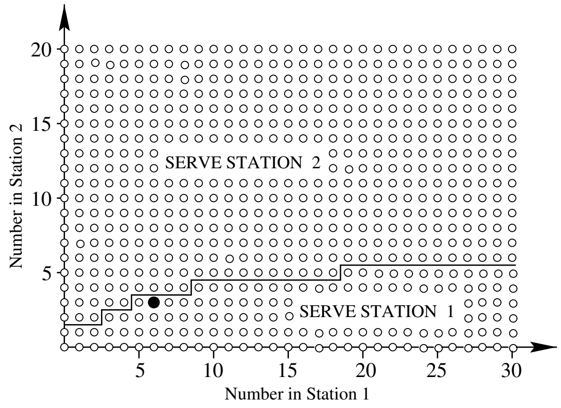


Fig. 1. Switching curve for scenario 1 of Table 1.

the more skilled worker, FW, take on an unfair share of the burden? Although bottleneck stations and variation in holding cost cause some imbalance, Table 1 suggests that the system under an OP will tend to be "fair" (utilizations within 8%) in most cases. Scenarios 3 and 4 minimize WIP (or CT) under unequal service rates and show success in fairly utilizing labor. We note that under the four worker system (LBM), the utilization of each worker is 71% at the bottleneck station and 56% at the faster station. Using the FW, labor utilization is relatively fair in spite of substantially increased utilization (to an average of 85% since it has 1/4 less capacity than LBM).

In scenario 2 the service rates are larger, which results in a lower utilization of the workers and the result is that FW has reduced ability to allocate the work fairly. Regarding scenarios 1, 2, 5, and 8, the reason worker 1 is more highly utilized is that whenever the holding costs are equal (or increasing) and the system is balanced, FW favors working in queue 2, thus worker 1 gets less assistance. To clearly see this characteristic of an OP, Fig. 1 shows how dramatically the FW is favored at station 2. This is a combination of factors, including the fact that service of a job at station 2 moves the job to a zero holding cost location, the exit, while service at station 1 does not immediately eliminate holding cost. Nevertheless, an optimal policy is clearly not a strict priority rule favoring station 2, because FW must switch to station 1 to leave enough jobs at station 2 to keep the specialist at station 2 from starvation.

Scenarios 5 through 7 examine a situation in which the holding cost at station 1 is half that at 2. As expected, more jobs will be allowed to build up at 1. From Fig. 2(i) we see that the switching curve (compared to Fig. 1) modestly increases the set of states in which the server attends station 2, a result due to a lower holding cost of 0.5 at station 1. Fig. 2(ii) shows that, compared to Fig. 2(i), reversal of the bottleneck position lowers the threshold curve.

Table 2 gives results for a range of three station scenarios. Scenarios 1 through 5 are average WIP (or CT) minimizations. Scenarios 3 through 5 explore the effects of bottleneck location, placing it at stations 1, 2, and 3, respectively. As the bottle-

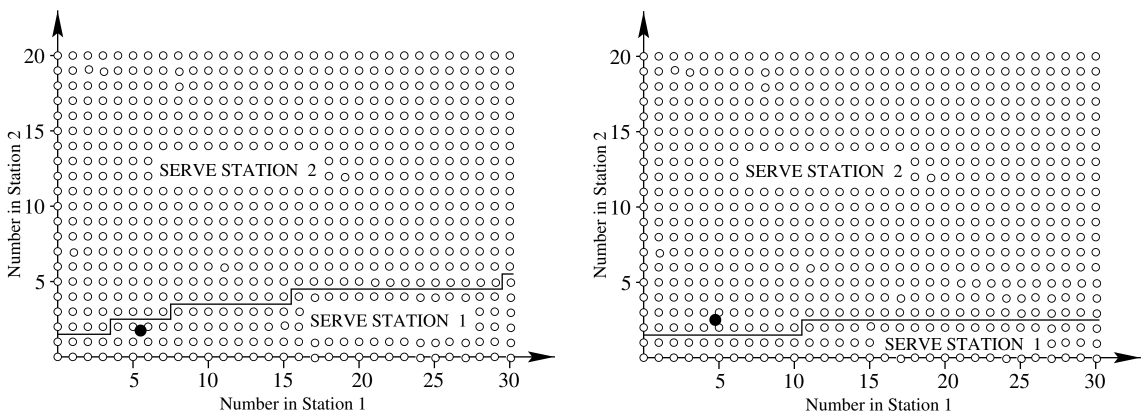


Fig. 2. Switching curves for (i) left: scenario 6 of Table 1, (ii) right: scenario 7 of Table 1.

neck moves down the line, the average WIP decreases, although it impacts performance by less than 5% in these cases.

In scenario 6 the value of a job doubles as it moves from 1 to 2, and then doubles again from 2 to 3. This can happen, for example, when each station performs a very high precision machining operation. The service rates are equal. As seen before, more jobs are allowed to accumulate at station 1. The utilization of the FW is 89%, whereas that of the specialized workers varies from 76% at station 3 to 87% at station 1. This reflects an increased burden placed upon the FW to (1) minimize the holding costs and (2) keep the specialists from starvation (the case of unit holding costs). Even a single flexible worker can exert some control on the WIP levels; while PUSH systems without cross-training cannot do this.

4. Second model with set-ups: FW-2

In this section we develop model FW-2. To prevent the model from becoming overly complex, let us combine the walking time from one station to another in with the set-up time, so that the dominant consideration is station set-up time, independent of the prior station. This assumption will be approximately true in a U-shaped line, the geometry of which minimizes walking times. The set-up time at station s is exponentially distributed with parameter σ_s.

4.1. The MDP formulation for FW-2 model

For the MDP formulation of the FW-2 model we have:

- *State of the system*: The state space is given by $(\mathbf{i}, k, z)$, where $\mathbf{i}$ is the vector of buffer occupancies, k is the current station in which FW currently resides, and $z \in \{1, 2\}$, where $z = 1$ means that the FW is setting up at k, and $z = 2$ means that the FW has already been set up at station k.
- *Decision epochs*: Decision epochs are service completions, job arrivals and set-up completions.
- *Action space*: When the system is in state $(\mathbf{i}, k, z)$ and either an arrival or a service completion has just occurred, then an action $a \in \{1, 2, \ldots, K\}$ may be chosen. If action $a = k$ is chosen, then the FW proceeds as before, either setting up or, if set up and there are two or more jobs in the buffer, then serving. However, if an action $a \neq k$ is chosen, then a set-up at the new station a is initiated, and an instantaneous set-up cost D_a is imposed. Set-ups may be preempted, but the entire set-up cost is incurred as a sunk cost.

The set-up cost D_a might involve preparations the FW must go through before serving, such as putting on special gloves or disposable clean-room clothing, the amortized costs of flexible tooling used to enable a fast set-up, etc. Roughly speaking, one can use the set-up cost as a surrogate for set-up time. The holding cost rate depends only on the buffer occupancies and is given by $g(\mathbf{i}) = \sum_{s=1}^{K} H_s i_s$.

The time between state transitions is exponentially distributed with transition rate

$$v(\mathbf{i}, k, z, a) = \begin{cases} \lambda + \sum \mu_s I(i_s \geqslant 1) + \sigma_a; & a \neq k, \\ \lambda + \sum \mu_s I(i_s \geqslant 1) + \sigma_k I(z = 1) \\ \quad + \mu_k I(i_k \geqslant 2,\ z = 2); & a = k. \end{cases} \tag{13}$$

The transition probabilities follow much as in model FW-1.

To compute the average cost for model FW-2, denoted $J^{(2)}$, and an optimal stationary policy the discrete-time structures Δ and Δ_N are formed much as they were for model FW-1. The costs are given by $C(\mathbf{i}, k, z, a) = g(\mathbf{i}) + D_a v(\mathbf{i}, k, z, a) I(a \neq k)$, and τ satisfies

$$\tau = \left(\lambda + \sum_{s=1}^{K} \mu_s + \max\{\mu_s, \sigma_s\} \right)^{-1}. \tag{14}$$

4.2. Stability condition for FW-2

For the case of FW-1, we found it useful to analyze the PR rule. With significant set-up costs or times; however, having FW follow the PR rule can result in excessive switching. For this reason, we consider the following policy, which we refer to as *batching policy* (*BP*): The BP policy has a cyclic structure, and begins with the set-up of station 1. When the set-up is completed, FW will start serving jobs until B jobs are completed, whereupon she or he starts setting up station 2. He or she then completes serving these B jobs in the second station and starts setting up station 3. Proceeding in this manner, the worker finishes all jobs in station K and sets up station 1. Note that with $B=1$, BP reduces to the PR rule.

The batching policy is a special case of the Triple Threshold (TT) policy introduced in Iravani et al. (1997a) for a two-stage tandem queue attended by a moving server. According to the TT policy, upon set-up completion in stage 1, the service begins only if the number of customers in that stage is at least M_w; otherwise, the worker waits there until the number of customers reaches that limit. When the server does start serving customers in state 1, she or he continues serving until either M_n ($M_n \geqslant M_w$) services have been completed without interruption, or the first stage becomes empty and there are at least M_e customers in stage 2 ($M_w \leqslant M_e \leqslant M_n$), whichever occurs first. After servicing at stage 1, the server sets up stage 2, and serves all customers in that stage. Note that the number of customers in that stage is between M_e and M_n. Iravani et al. (1997a) shows that the TT policy is a very cost effective policy with the total cost of being very close to the global optimal policy that minimizes the total average holding and set-up costs. Note that our batching policy is a special case of the TT policy where $M_w=1$, and $M_e=M_n=B$.

Assumption C. With B as the batch size, μ as the bottleneck rate, $t_B=\lambda\sum_{k=1}^{K}(\mu_k^{-1}+\sigma_k^{-1}/B)$, and $r=\lambda/\mu$, assume that

$$t_B^{-1} > 1 - r^{-1}.$$

Theorem 8. *A batching policy with a given batch size, B, for model FW-2 is stable if, and only if, Assumption C holds.*

Although Assumption C is more restrictive than B, when either the batch size, B, is very large (i.e., $B\to\infty$) or as set-up times go to zero, then Assumption C becomes Assumption B. Under Assumption B, it is the case that the analog of Theorem 2 (on using value iteration to compute an optimal policy and its finite average cost) holds for model FW-2, but we omit the details for brevity. The sufficiency of Assumption B is described in the following corollary to Theorem 8.

Corollary 1. *There exists a batching policy with a sufficiently large batch size to stabilize model FW-2 if Assumption B holds.*

The performance analysis of the batch policy gets extremely complex in general. When set-up times are sufficiently small, however, then PR may use a batch of size 1 and a simple performance expression can be obtained. Then, when Assumption C holds for a batch of size $B=1$, $D^* := \sum_{s=1}^{K} D_s$, and p is chosen to satisfy $1-r^{-1} < p < t_B^{-1}$, the average cost under PR is given by

$$J_p^{(2)} = (1-p)\sum_{k=1}^{K}\left(\frac{H_k r_k}{1-(1-p)r_k}\right) + p\lambda\left(D^* + \sum_{k=1}^{K} H_k(\sigma_k^{-1}+\mu_k^{-1})\right) + \frac{H_1 p^2\left(t^2 + \sum_{k=1}^{K}(\lambda\sigma_k^{-1})^2 + \sum_{k=1}^{K} r_k^2\right)}{2(1-pt_B)}. \tag{15}$$

4.3. Value iteration equations for FW-2

We now construct the equations for value iteration. The base point is arbitrary, and we choose it as $(\mathbf{0},1,2)$, i.e. the system is empty and the FW is set up at the first station. For the MDP, let $z^* = I(a\neq k) + zI(a=k)$ as a device to set z^* to z when $a=k$, and z^* to 1 in transitions where a set-up is in progress and an arrival or specialist service completion occurs. To simplify, we omit the stage subscript n and obtain an expression that looks complex, but follows the general idea of (7).

$$
\begin{aligned}
&w(\mathbf{i},k,z)\\
&\quad = g(\mathbf{i}) + \min_a\{D_a v(\mathbf{i},k,z,a)I(a\neq k)\\
&\qquad + \tau\lambda u(\mathbf{i}+\mathbf{e}_1,a,z^*)\\
&\qquad + \tau\sum_s \mu_s I(i_s \geqslant 1)u(\mathbf{i}-\mathbf{e}_s+\mathbf{e}_{s+1},a,z^*)\\
&\qquad + \tau\sigma_a I(a\neq k,\ \text{or}\ (a=k,z=1))u(\mathbf{i},a,2)\\
&\qquad + \tau\mu_a I(i_a\geqslant 2,\ a=k,z=2)u(\mathbf{i}-\mathbf{e}_a+\mathbf{e}_{a+1},a,2)\\
&\qquad + (1-\tau v(\mathbf{i},k,z,a))u(\mathbf{i},k,z)\}. \qquad (16)
\end{aligned}
$$

The proof of the next result is omitted since it follows from minor modifications to the Proof of Theorem 3. Note that the instantaneous (lump sum) costs receive both a monetary and time scaling, because of the conversion of the continuous time system to a discrete-time, uniformized one.

Theorem 9. *Let b and c be positive constants. Consider the modified system model, $(*)$, with parameters $\lambda^* = b\lambda$, $\mu_s^* = b\mu_s$, $\sigma_s^* = b\sigma_s$, $H_s^* = cH_s$, and $D_k^* = cb^{-1}D_k$ for $s = 1,2,\ldots,K$. An optimal policy for the original system computed using VI is also optimal for $(*)$. The average cost for $(*)$ is equal to $cJ^{(2)}$.*

4.4. Numerical results for FW-2

The VI equations (16) were implemented for $K = 2$ and 3 stations. Table 3 gives the key scenarios for $K = 2$, while Table 4 treats $K = 3$. For both tables, in all scenarios the set-up cost, D, is equal for all stations. Our lower benchmark (column five) is the average cost $J^{(1)}$ calculated for the same

Table 3
Results for model FW-2 with $K = 2$, $H_2 = 1$, and $\lambda = 1$

Scenario	H_1	μ_1 μ_2	σ_1 σ_2 D	$J^{(1)}$	$J^{(2)}$	Avg. # jobs 1 2 Line	Util. 1 2	Util. FW 1 2 Total	Prop. time setting up
1	1.00	0.75 0.75	5.0 5.0 0.0	9.10	10.06	6.17 3.89 10.06	0.90 0.90	0.43 0.43 0.86	0.050
2	1.00	0.75 0.75	5.0 5.0 5.0	9.10	10.98	6.12 4.17 10.28	0.90 0.91	0.43 0.42 0.85	0.027
3	1.00	0.75 0.75	5.0 5.0 10.0	9.10	11.61	6.03 4.40 10.42	0.90 0.92	0.43 0.42 0.85	0.023
4	1.00	0.70 0.90	5.0 5.0 10.0	7.18	9.15	4.71 3.49 8.21	0.89 0.87	0.54 0.24 0.78	0.019
5	1.00	0.90 0.70	5.0 5.0 10.0	6.64	9.06	4.00 3.74 7.74	0.84 0.91	0.27 0.52 0.79	0.026
6	0.50	0.75 0.75	5.0 5.0 10.0	5.90	8.37	6.90 3.72 10.62	0.91 0.91	0.42 0.42 0.84	0.024
7	0.25	0.80 0.80	5.0 5.0 0.0	2.95	3.40	5.42 2.04 7.46	0.89 0.83	0.36 0.42 0.79	0.088
8	0.25	0.80 0.80	5.0 5.0 0.0	2.95	4.61	5.04 2.55 7.59	0.89 0.85	0.36 0.40 0.76	0.032

Table 4
Results for model FW-2 with $K = 3$, $H_3 = 1$, and $\lambda = 1$

Scenario	H_1	μ_1	σ_1	$J^{(1)}$	$J^{(2)}$	Avg. # jobs	Util.	Util. FW	Prop. time setting up
	H_2	μ_2	σ_2			1	1	1	
		μ_3	σ_3			2	2	2	
			D			3	3	3	
						Line		Total	
1	1.00	0.85	5.0			4.82	0.88	0.29	
	1.00	0.85	5.0			3.89	0.88	0.29	
		0.85	5.0	10.40	11.90	3.17	0.88	0.29	0.075
			0.0			11.9		0.88	
2	1.00	0.85	5.0			4.70	0.88	0.30	
	1.00	0.85	5.0			4.02	0.89	0.29	
		0.85	5.0	10.40	13.36	3.40	0.88	0.29	0.046
			5.0			12.12		0.87	
3	1.00	0.85	5.0			4.65	0.88	0.30	
	1.00	0.85	5.0			4.21	0.89	0.28	
		0.85	5.0	10.40	14.36	3.59	0.89	0.28	0.036
			10.0			12.46		0.86	
4	1.00	0.80	5.0			4.14	0.87	0.38	
	1.00	0.95	5.0			2.77	0.83	0.22	
		0.95	5.0	8.40	9.45	2.55	0.82	0.23	0.087
			10.0			9.45		0.84	
5	1.00	0.95	5.0			3.68	0.83	0.22	
	1.00	0.80	5.0			3.08	0.87	0.38	
		0.95	5.0	8.17	9.23	2.47	0.82	0.23	0.089
			0.0			9.23		0.39	
6	1.00	0.95	5.0			3.60	0.83	0.22	
	1.00	0.95	5.0			2.80	0.83	0.22	
		0.80	5.0	8.04	9.13	2.73	0.86	0.39	0.093
			0.0			9.13		0.83	
7	0.25	0.90	5.0			4.79	0.87	0.24	
	0.50	0.90	5.0			3.19	0.87	0.24	
		0.90	5.0	3.76	6.61	2.38	0.81	0.30	0.036
			8.0			10.35		0.78	

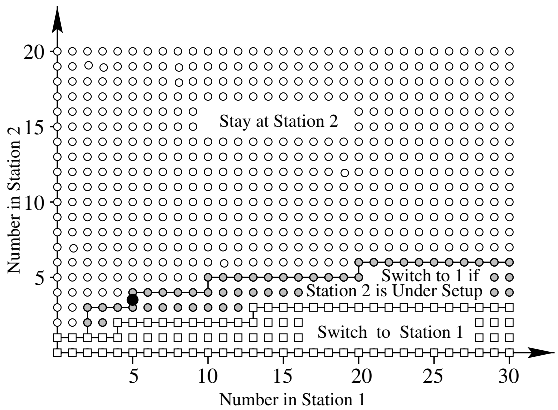
service rates and holding costs, but omitting set-up costs and times (model FW-1). This gives us a measure of how much additional cost is incurred in the presence of set-ups (Issue 3); equivalently, a manager would view this as the maximum value of set-up time and set-up cost reduction. We can see that $J^{(2)} - J^{(1)} > 0$, and the difference will decrease as the set-up costs are reduced or the set-up rates are increased. Addressing Issue 4 of Section 1, the last three columns indicate utilization information for the specialists and FW. The last column gives the proportion of time the FW spends setting up. Although the effect is not dramatic in these cases, with set-ups FW tends to have a lesser utilization than in the corresponding cases of model FW-1. We believe the reason for this is the increased fraction of time FW spends setting up and because FW is more reluctant to switch with model FW-2.

In our numerical study, we consider set-up times to be smaller than the service times for two reasons. (i) In most serial lines with a floater, the set-up times for the floater corresponds to the worker walking times between stations, or preparations the floater must go through before serving, such as putting on special gloves or disposable clean-room clothing. These operations are often shorter than the job processing times. (ii) One set-up time reduction approach converts "internal" set-ups (defined as those requiring a line stoppage) to "external" set-ups, which invest in resources (e.g., set-up kits) to prepare for the changeover off-line to allow for a quick changeover. Set-up times are thus reduced at the expense of kit preparation (which is appropriately modeled as a set-up cost from the perspective of line operation).

Scenarios 1–3 of Table 3 examine the effect of increasing D, for fixed service and set-up rates. Compared with a lower bound on the average cost, $J^{(1)} = 9.10$, we note that the increase in average cost is quite modest, even for a set-up cost of 10. This indicates that the OP compensates well for the extra demands of setting-up. From scenario 1 of Table 1, the utilizations of the specialized workers are comparable in models FW-1 and FW-2, and the utilization of the FW is only marginally decreased, from 89% to around 86%. It is intuitive that the flexible worker is less utilized than the others because the specialists have priority when there is only one job at a station, and because the FW must accept periods of starvation in a queue to avoid unnecessary set-ups. On the other hand, it is surprising that the difference is so modest in size, a feature that is helpful for implementation to be fair to the workers.

Fig. 3 shows the boundaries of the switching curves under an OP for scenario 1, which has an offered load of 1.18 at each station. In Fig. 3(ii), we have $k = 1$, i.e. the FW is located at station 1. The number of jobs present at station 2 for the FW to switch there is given by squares for $z = 2$ (the FW is already set up at station 1), and by a shaded circle when $z = 1$ (the FW is still under set up at station 1). When these two symbols coincide, only the square is shown. Note that the hysteresis (the vertical distance between the two curves) is due to being under set-up. This gap reflects the "investment" made in the time spent setting up a queue. In Fig. 3(i), we have $k = 2$, i.e. the FW is at station 2, and a threshold in the length of queue 1, i_1, is presented for each value of i_2. Note that this graph is read differently than the previous one, with the threshold in the horizontal direction. With regard to Issue 3, our testing (including scenarios 1, 2, and 3), revealed that increasing the set-up costs increases the gap between thresholds because the set-up times have not changed; however, the increased lump sum cost sets the thresholds higher so as to inhibit switching.

Scenarios 4 and 5 involve the interaction between the bottleneck station and the set-up times and costs. We can see that, as in Table 1, it is slightly more favorable to have the bottleneck at station 2. Scenario 6 is the same as scenario 3 except that the holding cost at the second station is



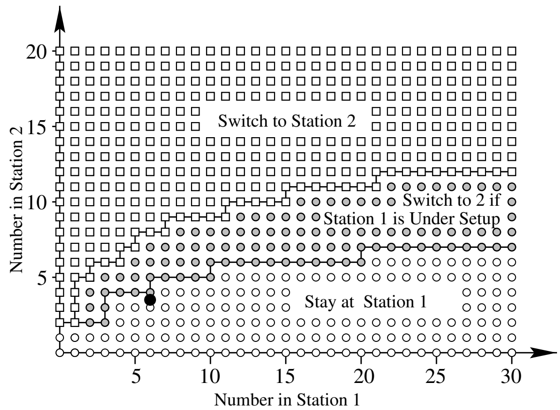


Fig. 3. Left: Switching curve, station 2 to 1, when FW is at station 2 (Table 3, scenario 1), right: switching curve, station 1 to 2, when FW is at station 1 (Table 3, scenario 1).

twice that at the first station. The resulting policy is able to achieve a significant cost reduction by holding more WIP in station 1, but the utilizations are hardly affected. Scenarios 7 and 8 further reduce the holding cost at the first station and slightly speed up the servers (dramatically reducing holding costs). Scenario 7 increases the set-up costs relative to 8, clearly causing utilization to drop for FW.

Table 4 gives results for a range of three station scenarios to give greater confidence that the insights gained are not limited to two-station lines. Scenarios 1 through 3 explore the effect of increasing set-up costs. Scenarios 4 through 6 explore effects of bottleneck location, placing it at stations 1, 2, and 3, respectively. Again, we see that it is slightly more favorable to have the bottleneck at the last station. As the bottleneck moves down the line, the average WIP decreases, although the performance impact is less than 5%.

In scenario 7 the value of a job doubles as it moves from 1 to 2, then doubles again from 2 to 3. In contrast to scenario 6 of Table 2, the average cost increases dramatically (an outcome of set-up times and costs) and the FW is forced to spend less time working (77% instead of 89%) so as to prevent excessive switching. To observe the relatively dramatic impact of set-up costs versus set-up times when the system is imbalanced, we contrast the preceding observation with the following. Compare scenarios 4–6 of Table 4 with scenarios 3–5 of Table 2, and note that the addition of set-up times without lump sum costs has a surprisingly modest impact on the system.

5. Conclusion and further research

We have provided a literature survey of queueing-based modeling of flexible worker systems. We then formulated and analyzed models of lines attended by specialists and augmented by a fully cross-trained flexible worker. Problems combining both cross-trained workers and specialists are especially difficult to optimize, because a cross-trained worker must make effective use of his or her own effort and also keep the specialists from starvation. The burden of maximizing performance falls on the worker with the greatest flexibility. Furthermore, we have extended the issue by introducing set-up costs and times. Thus, we have emphasized numerical analysis and the development of insightful bounds that indicate the value of cross-training (LBM) and the value of optimal dynamic scheduling (PR and DIV UBM's).

In our tests, two-station systems with two specialists at each of two stations (four workers) did not perform that much better than the flexible system with only three workers (an observation that clearly depends on our choice of models that generate 75–90% server utilizations). Our results indicate that the value of the flexible worker rests not only on its ability to provide line capacity balancing (to stabilize the line and optimize capacity allocation), but also in its ability to buffer variability in the workload process.

We demanded an optimal solution accurate for an infinite state space system, and we have demonstrated the feasibility of computing optimal policies over a portion of the state space for small systems with two or three stations. Specifically, we have harnessed recent contributions in the computation of dynamic programs to compute policies for finite state space models that agree with the true values in an infinite state space formulation. Furthermore, optimal average holding and set-up cost policies also yield a fairly even distribution of worker utilization. Thus, they promise to be workable in implementation.

We have observed in practice that systems with floating workers are best suited to fairly short lines or U-shaped work cells, because long lines lead to growing inter-station walk times that will render the use of a single, fully cross-trained worker impractical. On the other hand, some operations such as networked computer-based processing replace worker movement with the movement of tasks to the workers, possibly electronically, thereby opening up the potential for longer lines. The primary impediment to implementation may be the complexity of implementing an optimal policy. We leave it as a topic of further research to identify easily implementable heuristic policies that are also effective and do not require complex optimization procedures.

Acknowledgement

The second and third authors gratefully acknowledge the support of the National Science Foundation under grant DMI-0099821.

Appendix A

Proof of Theorem 1. The proof has two parts: in Part 1, we prove that if Assumption A holds, then LQ can stabilize the system, and in Part 2, we prove that if $q > |\mathscr{S}| + 1$, then the system is unstable.

Part 1: Proof for "if Assumption A holds, then LQ can stabilize the system"

We prove Part 1 by proving the contrapositive: If FW-1 is unstable under LQ, then Assumption A does not hold.

Let β_s^{FW} be defined to be the long run fraction of FW's time in queue s either idling or busy working as follows: Define $I_{\mathrm{FW}}^s(t)$ to be a stochastic process which is an indicator function that takes the value 1 whenever FW either serves a job or simply idles at station s. Then

$$\beta_s^{\mathrm{FW}} := \lim_{T\to\infty} \frac{1}{T} \int_0^T I_{\mathrm{FW}}^s(t)\mathrm{d}t. \tag{A.1}$$

In an analogous way, define γ_s^{FW} to be the long run fraction of FW's time in queue s busy working (and not idling), so $\gamma_s^{\mathrm{FW}} \leqslant \beta_s^{\mathrm{FW}}$.

We begin with some preliminary analysis to verify that the system is sufficiently well-behaved for our approach. Under LQ, the stationary Markov chain describing the system's behavior is easily recognized to be irreducible. Note that the the controlled Markov chain is established based on the rule that specialists never idle when a job is present. This guarantees that the Markov chain is aperiodic. From Definition 3.11 and Section 3.5.3 of Kulkarni (1995), the Markov chain is ergodic (weakly ergodic in the case of instability) and the steady-state distribution on the Markov chain exists. Since β_s^{FW} can be expressed in terms of the probability of being in a set of states in the Markov chain, therefore it exists (as a constant, almost surely) and lies in the range [0,1]. Knowing that β_s^{FW} exists, we now resume our proof and show that if FW-1 is unstable under the LQ policy, then Assumption A does not hold. The proof proceeds using two cases.

Part 1, Case 1: We hypothesize that all queues in $\mathscr{S}$ are unstable and we show that Assumption A does not hold. To add rigor in understanding the implications of server utilization (specialist and FW), we use Little's Law (see for example Theorem 7.4 of Kulkarni, 1995) applied individually to the specialist server only and to the floating worker only at s. For the specialist case, we focus on the *subsystem model* that captures only the server (without any queue), so the average waiting time is precisely the average time spent in service, μ_s^{-1}. The long run time average number of jobs in our subsystem model, $L_{\mathrm{sub}}^{\mathrm{SP}}$, becomes the long run average fraction of time the (specialist) server is busy at s, which we will call γ_s^{SP}. It clear that when arrival rate increases, and thus the flow rate through the specialist approaches μ_s from below, Little's Law proves that γ_s^{SP} exists, is finite, and approaches from below the value of 1 (meaning a 100% busy specialist). On the other hand, repeating this exercise and focusing on a process of jobs completed by FW reveals that the FW will generate a throughput rate at s of $\Theta_s^{\mathrm{FW}} = \gamma_s^{\mathrm{FW}}\mu_s$ jobs per unit time. Hence, there will be a combined departure rate from s of $\Theta_s^{\mathrm{SP}} + \Theta_s^{\mathrm{FW}} = (1+\gamma_s^{\mathrm{FW}})\mu_s$. It is easily seen from the dynamics of the entire queueing system that the job output rate at any workstation cannot exceed the job arrival rate to the entire system. Thus, we have

$$(1+\gamma_s^{\mathrm{FW}})\mu_s \leqslant \lambda. \tag{A.2}$$

Under LQ, FW never idles in the system when two or more jobs are present in at least one queue. When all queues in $\mathscr{S}$ are unstable, they grow unboundedly over an infinite length of time, so under the LQ policy the long run expected fraction of time FW is busy working in queues in $\mathscr{S}$ is 1, and

$$\gamma_s^{\mathrm{FW}} = \beta_s^{\mathrm{FW}}. \tag{A.3}$$

Combining (A.2) and (A.3), multiplying both sides by μ_s^{-1} and using the definition of r_s, we get

$$(1+\beta_s^{\mathrm{FW}})\mu_s \leqslant \lambda, \tag{A.4}$$

$$\beta_s^{\mathrm{FW}} \leqslant (r_s - 1). \tag{A.5}$$

Combining this with the fact that by definition $\sum_{s\in\mathscr{S}}\beta_s^{\mathrm{FW}} = 1$ and the fact that queues in $\{1,2,\ldots,K\} - \mathscr{S}$ are stabilized by specialists alone and therefore do not receive help from FW in the long run when queues in $\mathscr{S}$ are unstable, we get

$$1 = \sum_{s\in\mathscr{S}} \beta_s^{\mathrm{FW}}, \tag{A.6}$$

$$\leqslant \sum_{s\in\mathscr{S}} (r_s - 1) = q - |\mathscr{S}|. \tag{A.7}$$

So, $q \geqslant |\mathscr{S}| + 1$, which is the negation of Assumption A.

Part 1, Case 2: We now consider the possibility that under the LQ policy some of the queues in $\mathscr{S}$ are unstable (there exists a non-empty set of unstable queues, $U \subset \mathscr{S}$), while other queues in $\mathscr{S}$ are stable (there exists a non-empty set of stable queues, $Y = \mathscr{S} - U$). We show that this case cannot exist under the LQ policy, so Case 1 establishes the result. It follows from $(r_s - 1) \geqslant 0$ for $s \in Y$ and the assumption that queues in Y are stable, that under any policy (including LQ) FW spends a positive long run average fraction of her or his time in every queue in Y to stabilize it ($\gamma_s^{\mathrm{FW}} > 0,\ s \in Y$). It is possible that FW spends time in queues in $\{1,2,\ldots,K\} - \mathscr{S}$, but that is not necessary because all queues in this set can be stabilized by the specialist without any help from FW. In contrast, any queue in U is unstable and will have a larger queue length than any queue in Y or in $\{1,2,\ldots,K\} - \mathscr{S}$ over an unbounded length of time, and thus LQ always assigns FW only to queues in U or in $\{1,2,\ldots,K\} - \mathscr{S}$ over an unbounded length of time. This implies that LQ never allocates FW to any queue in Y (that is, $\gamma_s^{\mathrm{FW}} = 0,\ s \in Y$), which is a contradiction!

Part 2: Proof for "if $q > |\mathscr{S}| + 1$, then the system is unstable"

We show the contrapositive, i.e., if the system is stable, then $q \leqslant |\mathscr{S}| + 1$. We refer to the approach using Little's Law taken in Case 1 above. When the system is stable, all workstations are stable and so there is no long-term buildup in any queue. Hence, the combined departure rate from any station $s \in \mathscr{S}$, (i.e., $\Theta_s^{\mathrm{SP}} + \Theta_s^{\mathrm{FW}}$) must equal the long run system arrival rate:

$$\Theta_s^{\mathrm{SP}} + \Theta_s^{\mathrm{FW}} = \lambda. \tag{A.8}$$

Because the average queue length of the *subsystem model* of the specialist server only can be at most 1, the departure rate from the specialist can be at most μ_s, i.e., $\Theta_s^{\mathrm{SP}} \leqslant \mu_s$. Thus, we get

$$\Theta_s^{\mathrm{FW}} = \lambda - \Theta_s^{\mathrm{SP}} \geqslant \lambda - \mu_s. \tag{A.9}$$

Similarly, the departure rate from the FW is limited to $\Theta_s^{\mathrm{FW}} = \gamma_s^{\mathrm{FW}}\mu_s \leqslant \beta_s^{\mathrm{FW}}\mu_s$, so using (A.9) we get:

$$\begin{aligned} \beta_s^{\mathrm{FW}}\mu_s &\geqslant \Theta_s^{\mathrm{FW}} \geqslant \lambda - \mu_s, \\ \beta_s^{\mathrm{FW}} &\geqslant (\lambda - \mu_s)/\mu_s = r_s - 1. \end{aligned} \tag{A.10}$$

Observe that it is a basic constraint that FW allocate exactly 100% of her or his time (working or idle) to the entire system of queues. Under our definition of β_s^{FW} and using (A.10) we have:

$$\begin{aligned} &\sum_1^K \beta_s^{\mathrm{FW}} = 1, \\ &\sum_{s\in\mathscr{S}} \beta_s^{\mathrm{FW}} \leqslant 1, \\ &\sum_{s\in\mathscr{S}} (r_s - 1) \leqslant 1, \\ &\sum_{s\in\mathscr{S}} r_s \leqslant 1 + \sum_{s\in\mathscr{S}} 1, \\ &q \leqslant |\mathscr{S}| + 1. \qquad \square \end{aligned}$$

Proof of Theorem 2. We follow the approach of Sennott (1999, pp. 243–248) except that we deal with the average cost as defined in (3) and (4) rather than as defined in Sennott (1999). The latter definition is as follows. Given a policy θ and initial state $\mathbf{i}$, let F_n be the total cost incurred during the first n transition periods, and let T_n be the total amount of time for these transitions. Then define

$$G_\theta^{(1)}(\mathbf{i}) = \limsup_{n\to\infty} \frac{E_\theta^{(\mathbf{i})}[F_n]}{E_\theta^{(\mathbf{i})}[T_n]},$$

and let $G_\theta^{(1)}(\mathbf{i})$ be the infimum over all policies.

The argument has several steps. We first employ Sennott (1989, Prop. 5) to prove that (5) holds for some stationary policy. The stationary policy used to verify the hypotheses of this result is the LQ policy. Such a rule induces an irreducible positive recurrent Markov chain on the whole state space and produces a stable system with finite mean queue lengths. Thus, it has finite average cost, the conditions are satisfied. It is then the case that there exists a stationary policy f satisfying Sennott (1999, Eq. (10.15)) with Z replaced by $J^{(1)}$. This implies that $G_f^{(1)}(\mathbf{i}) \leqslant J^{(1)}$. Moreover, by Sennott (1989, Prop. 4) it is the case that $G_f^{(1)}(\mathbf{i}) = J_f^{(1)}(\mathbf{i})$. This implies that (5) holds so that f is average cost optimal for Ψ.

It then follows from Sennott (1999, Lemmas 10.3.2 and 7.2.1) that $J_f^{\Delta}(\mathbf{i}) \leqslant J^{(1)}$, where J_f^{Δ} is the usual average cost per unit time in Δ under f.

We now turn our attention to Δ and verify that the (AC) assumptions from Sennott (1999, p. 169) hold. These assumptions guarantee that there exists an optimal stationary policy with constant average cost in Δ_N, that the limit of these average costs converges to the average cost in Δ, and that any limit point of optimal stationary policies computed by value iteration is optimal in Δ.

(AC 1) states that there is a solution to the value iteration equation in Δ_N. To show this, it is sufficient by Sennott (1999, p. 117) to show that any optimal stationary policy has aperiodic positive recurrent classes. Since the set-up of Δ (and hence of Δ_N) always includes self-loops, this is clear. To verify the other assumptions, we employ Sennott (1999, Prop. 8.2.1, Steps 2–4).

For the policy in Step 2 we may take the LQ rule. We must verify conformity for this rule. This means that if the queues in the LQ rule are limited to N, as in Δ_N, then the steady-state probabilities and average cost converge to those for the LQ rule. This follows from the ideas in Sennott (1999, Prop. C.5.3). As our "base point" we take the empty system, with the FW at station 1. We need to compare the expected time and expected holding cost to go from an "overflow" state back to the base point with the respective quantities in the restricted system. The former must exceed the latter. This is intuitively clear, since if jobs are ejected from the system, then things only get better.

To verify Step 3 we argue that the minimum expected cost of operating in Δ_N for n steps does not exceed the respective quantity in Δ. Again this is intuitively clear since throwing jobs away can only decrease the cost. To verify Step 4, take as a base point $\mathbf{0}$. We operate in Δ_N for n steps, and argue that the minimum expected cost from any initial state is at least as great as the cost from $\mathbf{0}$. This is intuitively clear.

This verifies that the AC assumptions hold and hence an optimal stationary policy e for Δ may be computed as a limit point from optimal stationary policies in Δ_N. Moreover, the average cost J^{Δ} in Δ and e satisfy Sennott (1999, Eq. (10.20)). But this implies that $G_e^{(1)}(\mathbf{i}) \leqslant J^{\Delta}$.

Putting together what has been obtained so far, we see that $G_e^{(1)}(\mathbf{i}) \leqslant J^{\Delta} \leqslant J_f^{\Delta}(\mathbf{i}) \leqslant J^{(1)} \leqslant J_e^{(1)}(\mathbf{i})$. We now argue informally that the policy e must induce a positive recurrent Markov chain on the whole state space, since the holding costs are unbounded. This implies by Ross (1970, Theorem 7.5), that the outer terms of the string of inequalities are equal. This proves that e is average cost optimal for Ψ with constant average cost J^{Δ}. Hence the computational procedure is valid. □

Proof of Theorem 3. Substituting the modified values into τ yields $\tau^* = \tau/b$. This implies that the products $\tau\lambda$ and $\tau\mu_s$ remain constant. We may then prove by induction on n that $u_n^* = cu_n$, and the result follows. □

Proof of Theorem 4. Using standard balance equation techniques, it may be shown that the steady-state probabilities are $\pi_0 = (1-\rho)/(1+\alpha\rho)$, and $\pi_i = (1+\alpha)\rho^i\ \pi_0$, for $i \geqslant 1$. Then (8) easily follows. □

Proof of Theorem 5. We begin by assuming a stable division, and we show that Assumption A must hold. Using Theorem 4, the definition of α_s implies that in a stable system $r_s < 1 + \alpha_s$ for $\forall s \in \mathscr{S}$. If $|\mathscr{S}| = K$ (all stations get helped by FW), then summing over all stations yields

$$\sum_{s=1}^{K} r_s < \sum_{s=1}^{K}(1+\alpha_s),$$
$$q < K + \sum_{s=1}^{K}\alpha_s,$$
$$q < K + 1 = |\mathscr{S}| + 1$$
$$\left(\text{since } \sum_{s=1}^{K}\alpha_s = 1 \text{ and } |\mathscr{S}| = K\right).$$

Otherwise, if $|\mathscr{S}| < K$, then we have at least one station $j \notin \mathscr{S}$, which implies that station j is stable with $\alpha_j = 0$ and it does not enter into the condition. Then, summing over all stations $s \in \mathscr{S}$ we get $q < |\mathscr{S}| + 1$. Hence, the condition is necessary.

To conclude, we now assume Assumption A, and we show that the division model is stable. We observe that the cost (and hence the queue lengths) of our division model are finite. This is seen from Eq. (9) as follows: If $s \notin \mathscr{S}$, then $r_s < 1$, and the terms of the sum reduce to $(H_s r_s)/(1 - r_s)$, which is well-defined (and bounded). When $s \in \mathscr{S}$, then $\alpha_s > 0$, and the denominator of the reciprocal term in the sum reduce to

$$\left(1+\frac{\alpha_s r_s}{1+\alpha_s}\right)\left(\frac{1+\alpha_s-r_s}{1+\alpha_s}\right) = \left(1+\frac{\alpha_s r_s}{1+\alpha_s}\right)\left(\frac{\varepsilon_s}{1+\alpha_s}\right).$$

By construction, $\varepsilon_s > 0$, so the product is positive (i.e., non-zero) and $J^{(1)}_{\bar{\alpha}}$ is finite. □

Proof of Theorem 6. We first show that if Assumption B holds, then there exists a stable PR rule. Under Assumption B, it easy to check that p can be chosen so that $0 < p < 1$, and $1 - r^{-1} < p < q_K^{-1}$. We show that if Assumption B holds, Eq. (11) is finite, and hence the system is stable. For the first term of (11), observe that $1 - r^{-1} < p$ implies that $1 - (1 - p)r > 0$. We noted prior to Assumption B that $r \geqslant r_s$ for all s. This, together with $(1 - p) > 0$ guarantees that $1-(1 - p)r_s > 0$ for all s. Thus, the denominator of the first term on the right hand side of (11) is positive, which implies that the term is finite. It is obvious that the remaining terms are also finite.

We prove necessity by contradiction. Suppose that there exists a stable PR rule, but Assumption B does not hold, i.e., we have $q_K^{-1} + r^{-1} \leqslant 1$. However, for stability of the FW queue, we must have $p < q_K^{-1}$. This is because the FW queue is in fact a $G/G/1$ queue with arrival rate $p\lambda$ and average service time $\sum_{k=1}^{K} 1/\mu_k$. On the other hand, we have $q_K^{-1} \leqslant 1 - r^{-1}$. This then implies that $p < 1 - r^{-1}$, which means that the tandem system (i.e., the queues of specialists) is unstable—a contradiction! Hence, the condition is also necessary.

The average cost is then given by (11). The first term is the average cost associated with the tandem system, and it follows from standard $M/M/1$ theory. The second term is the average holding cost incurred during service in the FW queue. It equals

$$\text{(average number of jobs completed per unit time)} \times \text{(average cost of a job)} = (\lambda p)\left(\sum H_s \mu_s^{-1}\right).$$

The third term is the average holding cost for jobs waiting in the queue at the first station, and follows from standard $M/G/1$ results. □

Proof of Theorem 7. We begin by showing the Assumption B implies Assumption A. First, observe that $q_K = \sum_{k=1}^{K} r_k > q$, so we have $q_K^{-1} < q^{-1}$. Using this, together with Assumption B, we get

$$1 < q^{-1} + r^{-1}. \tag{A.11}$$

Next, we use the definition of the bottleneck utilization, r, to place an upper bound on r^{-1} using the definition of q.

$$q = \sum_{s\in\mathscr{S}} r_s \leqslant \sum_{s\in\mathscr{S}} r = |\mathscr{S}|r,$$

so $r^{-1} \leqslant |\mathscr{S}|q^{-1}$. Using this in (A.11) we get

$$1 < q^{-1} + |\mathscr{S}|q^{-1} < q^{-1}(1 + |\mathscr{S}|), \tag{A.12}$$

and so $q < 1 + |\mathscr{S}|$, which is (6).

To see that Assumption B does not imply Assumption A, let $K = 2$, $\lambda = 1$, $\mu_1 = 0.8$, and $\mu_2 = 0.6$. It is easy to check that (6) holds but (10) fails. If μ_2 is increased to 0.65, then (10) will hold.

If the rates are equal, then $q_K = q = Kr = |\mathscr{S}|r$. It is easy to check that (10) and (6) are equivalent. $\square$

Proof of Theorem 8. One can show that the condition $1 - r^{-1} < t_B^{-1}$ holds if, and only if $\max\{(1-p)r, pt_B\} < 1$. We first show that if $\max\{(1-p)r, pt_B\} < 1$ holds, the system is stable. According to Iravani et al. (1997a) the stability condition for the TT policy in a two-stage tandem queue with arrival rate $p\lambda$ is

$$p\lambda[\mu_1^{-1} + \mu_2^{-1} + (\sigma_1^{-1} + \sigma_2^{-1})/M_n] < 1.$$

As it is clear, the stability of the TT policy does not depend on M_w and M_e. For our batching policy, this translates into

$$p\lambda[\mu_1^{-1} + \mu_2^{-1} + (\sigma_1^{-1} + \sigma_2^{-1})/B] < 1$$

and it can be easily extended to

$$p\lambda \sum_{s=1}^{K} (\mu_s^{-1} + \sigma_s^{-1}/B) < 1$$

or $pt < 1$ for a K-stage tandem queue (i.e., serial line). On the other hand, it is also clear that the stability condition for the line with specialists and arrival rate $(1-p)\lambda$ is $(1-p)r < 1$.

The entire system is stable when both the line with specialists and the FW line are stable. Assumption C implies $\max\{(1-p)r, pt_B\} < 1$, which guarantees the stability of both lines, and thus the stability of the entire system.

It can easily be shown using contradiction that if the system is stable, then $\max\{(1-p)r,\ pt_B\} < 1$ will hold. We omit it for brevity. $\square$

Proof of Corollary 1. First, we define

$$\delta = q_K^{-1} - (1 - r^{-1}).$$

By Assumption B, $\delta > 0$. Next, observe that

$$\lim_{B\to\infty} t_B = \lim_{B\to\infty} \lambda \sum_{k=1}^{K} (\mu_k^{-1} + \sigma_k^{-1}/B)$$
$$= \lambda \sum_{k=1}^{K} (\mu_k^{-1}) = q_K.$$

Because $t_B^{-1} < q_K^{-1}$ and in the limit $\lim_{B\to\infty} t_B^{-1} = q_K^{-1}$, there exists a sufficiently large natural number, B^*, such that $t_{B^*}^{-1} > q_K^{-1} - \delta/2$. It suffices then to show that Assumption C holds if we choose batch size B^*:

$$t_{B^*}^{-1} - (1 - r^{-1}) > (-\delta/2 + q_K^{-1}) - (1 - r^{-1})$$
$$= -\delta/2 + \delta = \delta/2,$$

and the result follows. $\square$

References

Ahn, H.-S., Duenyas, I., Zhang, R.Q., 1999. Optimal scheduling of a 2-stage tandem queue with parallel servers. Advances in Applied Probability 31, 1095–1117.

Andradottir, S., Ayhan, H., Down, D.G., 2001. Server assignment policies for maximizing the steady-state throughput of finite queueing systems. Management Science 47 (10), 1421–1439.

Andradottir, S., Ayhan, H., Down, D.G., 2003. Dynamic server allocation for queueing networks with flexible servers. Operations Research 51 (6), 952–968.

Askin, R.G., Chen, J., 2002. Dynamic task assignment for throughput maximization with worksharing. Working paper, SIE Department, University of Arizona, Tucson, AZ.

Bartholdi, J.J., Eisenstein, D.D., 1996. A production line that balances itself. Operations Research 44, 21–34.

Bartholdi, J.J., Bunimovich, L.A., Eisenstein, D.D., 1999. Dynamics of 2- and 3-worker 'bucket brigade' production lines. Operations Research 47, 488–491.

Bartholdi, J.J., Eisenstein, D.D., Foley, R.D., 2001. Performance of bucket brigades when work is stochastic. Operations Research 49 (5), 710–719.

Bischak, D.P., 1996. Performance of a manufacturing module with moving workers. IIE Transactions 28, 723–733.

Downey, B.S., Leonard, M.S., 1992. Assembly line with flexible work-force. International Journal of Production Research 30, 469–483.

Farrar, T.M., 1993. Optimal use of an extra server in a two station tandem queueing network. IEEE Transactions on Automatic Control 38, 1296–1299.

Gel, E.S., Hopp, W.J., Van Oyen, M.P., 2000. Workforce agility in systems with hierarchical cross-training. Working paper, Northwestern University.

Gel, E.G.S., Hopp, W.J., Van Oyen, M.P., 2002. Factors affecting opportunity of worksharing as a dynamic line balancing mechanism. IIE Transactions 34, 847–863.

Gross, D., Harris, C.M., 1998. Fundamentals of Queueing Theory, third ed. John Wiley, New York.

Hopp, W.J., Spearman, M.L., 2000. Factory Physics: Foundations of Manufacturing Management, second ed. Irwin/McGraw-Hill, Burr Ridge, IL.

Hopp, W.J., Van Oyen, M.P., 2004. Agile workforce evaluation: A framework for cross-training and coordination. IIE Transactions, 36.

Hopp, W.J., Tekin, E., Van Oyen, M.P., 2004. Benefits of skill chaining in production lines with cross-trained workers. Management Science 50 (1), 83–98.

Iravani, S.M.R., Posner, M.J.M., Buzacott, J.A., 1997a. A two-stage tandem queue attended by a moving server with holding and switching costs. Queueing Systems 46, 203–228.

Iravani, S.M.R., Posner, M.J.M., Buzacott, J.A., 1997b. U-shaped lines with switchover times and costs. Department of Industrial Engineering and Management Sciences Technical Report, Northwestern University.

Iravani, S.M.R., Buzacott, J.A., Posner, M.J.M., 2002. A robust strategy for serial agile production systems. Working paper, Department of Industrial Engineering and Management Sciences, Northwestern University.

Kulkarni, V.G., 1995. Modeling and Analysis of Stochastic Systems. Chapman and Hall, London.

Mandelbaum, A., Reiman, M.I., 1998. On pooling in queueing networks. Management Science 44, 971–981.

McClain, J.O., Schultz, K.L., Thomas, L.J., 2000. Management of worksharing systems. Manufacturing & Service Operations Management (M&SOM) 2 (1), 49–67.

McClain, J.O., Thomas, L.J., Sox, C., 1992. On-the-fly line balancing with very little WIP. International Journal of Production Economics 27, 283–289.

Ostolaza, J., McClain, J., Thomas, J., 1990. The use of dynamic (state-dependent) assembly-line balancing to improve throughput. Journal of Manufacturing and Operations Management 3, 105–133.

Puterman, M.L., 1994. Markov Decision Processes. John Wiley, New York.

Ross, S.M., 1970. Applied Probability Models with Optimization Applications. Holden-Day, San Francisco, CA.

Schultz, K.L., Juran, D.C., Boudreau, J.W., McClain, J.O., Thomas, L.J., 1998. Modeling and worker motivation in Just In Time production systems. Management Science 44, I595–I607.

Schweitzer, P., 1971. Iterative solution of the functional equation of undiscounted Markov renewal programming. Journal of Mathematical Analysis and Applications 34, 495–501.

Sennott, L.I., 1989. Average cost semi-Markov decision processes and the control of queueing systems. Probability in the Engineering and Informational Sciences 3, 247–272.

Sennott, L.I., 1999. Stochastic Dynamic Programming and the Control of Queueing Systems. John Wiley, New York.

Upton, D.M., 1995. What really makes factories flexible. Harvard Business Review (Jul.–Aug.).

Van Oyen, M.P., Gel, E.G.S., Hopp, W.J., 2001. Performance opportunity for workforce agility in collaborative and non-collaborative work systems. IIE Transactions 33 (9), 761–777.

Zavadlav, E., McClain, J.O., Thomas, L.J., 1996. Self-buffering, self-balancing, self-flushing production lines. Management Science 42, 1151–1164.