

## SAMPLE PATH OPTIMAL POLICIES FOR SERIAL LINES WITH FLEXIBLE WORKERS

DIMITRIOS G. PANDELIS,\* *University of Thessaly*

MARK P. VAN OYEN,\*\* *University of Michigan*

### Abstract

We study the dynamic assignment of cross-trained workers in serial production lines characterized by stochastic process times and inventory buffers between stations. Throughput maximization is the objective. Each worker is trained for a subset of tasks, where emphasis is placed on systems with each worker trained for a zone of stations with stations near the zone boundaries being served (shared) by one or more other workers as well. Using sample path comparisons, we identify structural properties of optimal worker allocation policies. We identify when (i) a worker can prioritize the job in the most downstream station (last-buffer–first-served), and (ii) only the downstream (as opposed to upstream) server should serve a single task.

*Keywords:* Stochastic scheduling; worker flexibility; throughput maximization; serial production line

2010 Mathematics Subject Classification: Primary 90B36

Secondary 90B22; 60G17

### 1. Introduction

Motivated by the increasing use of flexible workers in manufacturing systems, we study the optimal dynamic assignment of such workers in serial production lines. Each job entering the line consists of  $N$  tasks that have to be performed in order from the first of  $N$  stations until the last. We seek properties of worker allocation strategies that maximize the job completion rate, that is, the throughput of the line. A flexible workforce is sometimes used as a means of improving the performance of nonpaced manufacturing lines under challenging conditions, such as varying demand, processing times, resource availability, etc. As opposed to traditional settings where each worker performs a single specific task, flexible workers are trained for more than one task (usually with some overlap) so that instead of being forced to idle because of lack of work, they can assist with bottlenecks, thus enhancing productivity in lines that are not paced but have significant variability. Hopp and Van Oyen [13] provided a literature survey on workforce flexibility as well as a framework for evaluating a flexible workforce in an organization. Specifically, they established the connection between cross-training and the organization's objectives, and discussed methods for identifying and implementing an appropriate worker flexibility strategy.

As pointed out by Askin and Chen [7], there are two types of work sharing, dynamic assembly line balancing (DLB) and moving worker modules (MWM). In DLB, which was

---

Received 2 November 2010; revision received 16 June 2011.

\* Postal address: Department of Mechanical Engineering, University of Thessaly, Pedion Areos, 38334 Volos, Greece. Email address: d\_pandelis@mie.uth.gr

\*\* Postal address: Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109-2117, USA.

first introduced by Ostolaza *et al.* [19], there is an equal number of machines and workers. Each worker stays with his/her machine, but machines are flexible (e.g. hand tools, welding equipment, sewing machines, computers, etc.) so that workers can also perform the tasks of their upstream/downstream neighbors. McClain *et al.* [18] showed that even in lines without buffers, DLB can be effective. In a later paper Anuar and Bukchin [5] used simulation to evaluate several worker allocation rules in DLB systems and for most of them they showed significant improvements in cycle times versus systems with no work sharing. In MWM, each worker is trained to work within a ‘zone’ of stations. Machines are not flexible as in DLB, but workers can move from station to station. The well-known bucket brigades introduced by Bartholdi and Eisenstein [8] are an example of such a system. For deterministic processing times, they showed that sorting workers from slowest to fastest and allowing preemptions ensures the balancing of the line without the need for buffers. Similar systems with random processing times were studied by Bischak [10], Bartholdi *et al.* [9], and McClain *et al.* [17].

In general, each cross-training pattern is characterized by the sets of tasks that workers are capable of performing. It is not important to our purposes whether this is accomplished by using flexible machines or by moving from machine to machine within a zone. The most extensive form of cross-training is full cross-training, in which workers are trained for all tasks. Van Oyen *et al.* [22] showed that when workers can collaborate on the same job, throughput is maximized by a policy where workers are assigned to the same job throughout the production process, as in the traditional craft mode of production. Throughput maximizing policies were also determined by Andradóttir *et al.* [3], [4], Andradóttir and Ayhan [2], Arumugam *et al.* [6], and Kirkizlar *et al.* [15], [16] for a variety of two- and three-stage models. From a practical point of view, full cross-training is often too expensive and difficult to implement, so partial cross-training systems have naturally received increasing attention. Gel *et al.* [12] studied a two-stage system with two workers, one flexible (trained for both tasks) and one specialist (trained for one task). For a CONWIP (constant-work-in-process) structure, they established the optimality of the ‘fixed-before-shared’ principle for the flexible worker: fixed tasks (those only the flexible worker can perform) should be given strict priority over shared tasks. Gel *et al.* [11] studied a DLB system with two workers and three tasks. Assuming exponential processing times, they used MDP (Markov decision process) analysis to investigate the extent to which various factors affect the performance improvement that can be achieved relative to static worker allocations. For the same system with Erlang-distributed processing times, Askin and Chen [7] defined the tradeoff between the cost of WIP and the cost of cross-training. Hopp *et al.* [14] demonstrated the effectiveness of  $D$ -skill chains where each worker is trained for the task in his/her base station and  $D - 1$  more tasks down the line (U-shaped lines are assumed), with strong emphasis on the merits of 2-skill chains. Finally, Parvin *et al.* [21] presented a zone chaining pattern with limited cross-training that can achieve high throughput.

For any given cross-training pattern, an efficient worker allocation rule needs to be implemented. There are two kinds of decision to be made: (i) the task to which a worker should be assigned when he/she becomes available, and (ii) the worker to perform a task when two or more workers are available for a single task. As seen from the literature survey, optimal worker allocation policies have only been obtained for systems with few workers/tasks (see [12]) or under restrictive assumptions such as fully cross-trained and collaborating workers (see [22]). For general systems with many workers/tasks and possibly differing processing requirements and worker speeds, it is not to be expected that one can completely determine the optimal policy. Ahn and Righter [1] partially characterized optimal task selection policies for general production lines. In this paper we extend some of their results as we prove properties of optimal policies

under more general assumptions on service time distributions and worker skills. In addition, we derive properties of optimal worker selection strategies. Our paper also differs from theirs in the methodology used in the proofs. We use sample path comparisons in continuous time to characterize policies that maximize throughput along any sample path. On the other hand, they used a discrete-time model and applied inductive schemes on the number of periods defining the length of the time horizon.

The paper is organized as follows. The model is described in Section 2. Optimal task and worker selection policies are characterized in Sections 3 and 4, respectively. We conclude in Section 5.

## 2. Model description

We consider a tandem system consisting of  $N$  stations ( $N \geq 3$ ), where each station requires a specific task type that has to be performed for each job entering the system. We consider both open systems with external arrivals, and closed (CONWIP) systems, where each departure triggers an arrival so that the total WIP remains constant. The system is attended by  $M$  cross-trained workers, where  $M \leq N$ . Tasks may have different processing requirements and workers may have different speeds (possibly depending on tasks as well), that is, service times may be task type and/or worker dependent. We assume that (i) workers cannot collaborate (but multiple workers are allowed to work in *parallel* at the same station if a distinct job is available for each one), (ii) idling is allowed, (iii) task preemption is not allowed, and (iv) workers can move from station to station instantaneously. Assumptions (i)–(iii) make the problem more difficult. Task nonpreemption is both a practical reality in many operations and a reasonable assumption for tractability when service times are general in distribution.

We denote by  $T_i$ ,  $i = 1, 2, \dots, M$ , the set of tasks (which is identical to the labels of the stations) that can be performed by worker  $i$ . We assume that successive tasks performed at station  $i$  by the same worker are independent and identically distributed; however, the distribution on the task processing time may vary by station. Adopting the terminology in Gel *et al.* [12] we distinguish tasks into fixed tasks that can be performed by only one worker, and shared tasks that can be performed by more than one worker. Workers that have at least one fixed task are termed *exclusive*. Analytical results are only possible for sets  $\{T_i, i = 1, 2, \dots, M\}$  of special form, which also arise from practical considerations. For example, it is reasonable to train workers to work in adjacent stations, so each  $T_i$  corresponds to a zone in the line, or possibly a zone that has a generalized zone structure of adjacency as in the case of U-shaped lines where workers can work on both ends of the line, e.g. stations  $\{N - 1, N, 1, 2\}$ . This approach can greatly reduce worker walk times, may improve the ability of the system to utilize ‘visual control’, may enhance efforts to create accountability for product quality, and seems to simplify the challenge of identifying an effective dynamic control policy that can be implemented. Ahn and Righter [1] studied systems with  $T_i = \{k: k_i \leq k \leq \ell_i\}$ , where  $k_i$  and  $\ell_i$  are nondecreasing sequences with  $k_1 = 1$  and  $\ell_M = N$ . The two workers in Gel *et al.* [11] had training sets given by  $T_1 = \{1, 2\}$  and  $T_2 = \{2, 3\}$ . The lines for which Gel *et al.* [12] proved the ‘fixed-before-shared’ optimality had  $T_1 = \{1\}$ ,  $T_2 = \{1, 2\}$  or  $T_1 = \{1, 2\}$ ,  $T_2 = \{2\}$ . The  $D$ -skill chains introduced by Hopp *et al.* [14] are examples of U-shaped lines with  $T_i = \{i, (i + 1) \bmod N, \dots, (i + D - 1) \bmod N\}$ ,  $i = 1, 2, \dots, N$ . Finally, the FTZC model of Parvin *et al.* [21] is a U-shaped CONWIP line with  $W$  workers with 0 to  $N - W$  fixed tasks per worker for which every worker has one downstream shared task and one upstream shared task such that the fixed tasks, if any, are located between the two shared tasks.

In this paper we clarify principles of optimal worker control (task selection by available workers and worker selection for available tasks) that apply to open and/or closed queueing models. Our work is also distinguished by its generality of formulation regarding service time distributions and worker skills.

We determine properties of worker allocation policies that maximize job completions along any sample path. For a realization  $\omega$  that fixes the sequence of arrival and service times, we denote by  $D_t^\pi(\omega)$  the number of job completions by time  $t$  under a worker allocation strategy  $\pi$ . Therefore, we aim to characterize a policy  $\pi^*$  such that  $D_t^{\pi^*}(\omega) \geq D_t^\pi(\omega)$  for all  $t, \pi$ , and  $\omega$ . Starting with an arbitrary policy  $\pi$ , we define  $\pi^*$  as a modification of  $\pi$  and show that, for any sample path, the number of job completions under  $\pi^*$  is not less than those under  $\pi$ . In the following sections we provide partial characterizations of optimal policies, most of which apply to systems with fixed tasks and exclusive workers.

### 3. Optimal task selection

In this section we consider time instants when only a single worker is available to serve two or more jobs of different types and must choose which to serve.

#### 3.1. General processing times

Our first theorem addresses problems with arbitrary service time distributions and exclusive workers that are trained to work in a zone where the only shared task is the most upstream task of the zone. We show that the nonidling last-buffer–first-served (LBFS) policy is optimal when fixed tasks are present. In other words, once a job becomes available in the worker’s most upstream fixed station, he/she serves this job through all the stations in his/her skill set. This is a pick-and-run-type policy introduced by Van Oyen *et al.* [22].

**Theorem 1.** Consider worker  $i$  with  $T_i = \{m: k_i \leq m \leq \ell_i\}$ , where only  $k_i$  may be a shared task. Assuming arbitrary distributions for service times, the job completion process is maximized along every sample path by a policy that

- (i) does not idle worker  $i$  when fixed tasks are present, and
- (ii) follows the LBFS rule among the worker’s tasks.

*Proof.* We prove part (ii) first. Let  $\tau$  be a time instant when worker  $i$  becomes available with at least two stations in set  $T_i$  having waiting jobs, and let  $k$  be the most downstream such nonempty station. For any policy  $\pi$  that assigns worker  $i$  to some other station  $\ell$  in set  $T_i$ , where  $\ell < k$ , we construct an alternative policy  $\tilde{\pi}$  that assigns worker  $i$  to station  $k$  and is such that  $D_t^{\tilde{\pi}}(\omega) \geq D_t^\pi(\omega)$  along every sample path  $\omega$ . Because service times are task and worker dependent, sample path  $\omega$  is defined by sequences  $S_j^w(l, \omega)$ ,  $l = 1, 2, \dots$ , where  $S_j^w(l, \omega)$  is the time duration of the  $l$ th service performed by worker  $w$  in station  $j$ . We construct policy  $\tilde{\pi}$  to be identical to  $\pi$  except that at time  $\tau$  worker  $i$  is assigned to station  $k$  and then follows the sequence of actions taken under  $\pi$  after time  $\tau$ ; the first action in that sequence is a service in station  $\ell$ , and the rest (that may include idle periods) are determined from the realization of the arrival and service processes. Policy  $\tilde{\pi}$  is well defined as it can always mimic  $\pi$  with respect to actions taken by workers other than  $i$  because at any time all stations where these workers can be assigned (those not fixed to worker  $i$ ) have at least as many jobs under  $\tilde{\pi}$  as under  $\pi$ . The two policies are coupled at the time of the first service completion under  $\pi$  in station  $k$ , denoted by  $\tau_c(\omega)$ . Time periods  $[\tau + S_\ell^i(1, \omega), \tau_c(\omega) - S_k^i(1, \omega)]$  under  $\pi$  and  $[\tau + S_k^i(1, \omega) + S_\ell^i(1, \omega), \tau_c(\omega)]$  under  $\tilde{\pi}$  include the same sequence of actions by worker  $i$ . Comparing the two policies, we see

that we have the same number of job completions unless  $k$  is the last station ( $k = N$ ), in which case we have  $D_i^{\tilde{\pi}}(\omega) = D_i^{\pi}(\omega) + 1$  for  $\tau + S_k^i(1, \omega) \leq t \leq \tau_c(\omega)$ . Repeating the argument every time policy  $\pi$  does not give priority to the most downstream station proves part (i) of the theorem.

The proof of part (i) is based on a similar construction of the alternative policy  $\tilde{\pi}$ . After the service completion in station  $k$ , worker  $i$  idles for the same amount of time he/she would have idled under  $\pi$ , say  $I(\omega)$ , which can be calculated by observing the sample path realization. Then he/she replicates the actions taken under  $\pi$  until the two policies are coupled the first time a service is completed in station  $k$ . The evolution of policies  $\pi$  and  $\tilde{\pi}$  along some sample path  $\omega$  is essentially identical to that in part (ii) with  $S_k^i(1, \omega)$  replaced with  $I(\omega)$ .

Theorem 1 provides a very general result as it holds for general processing times and workers with different speeds. Unfortunately, optimal policies are complex and it is not clear whether the worker should work on a shared task or idle when fixed tasks are not present. It is also restrictive with respect to the task set of the exclusive worker under consideration (while the task sets of all other workers may be arbitrary). However, as shown in the detailed technical report of Pandelis and Van Oyen [20], Theorem 1 also holds for worker  $i$  with  $T_i = \{m : k_i \leq m \leq \ell_i\}$  in settings where  $k_i$  and  $\ell_i$  are the only shared tasks.

It could be argued that the LBFS rule is optimal within a worker's fixed tasks, irrespective of the number of shared tasks. Ahn and Righter [1] showed the optimality of this rule for fixed tasks with identically distributed processing times (see Theorem 2.1(ii) therein), a result that we will generalize in the following section. For general processing times and worker zones with an arbitrary number of shared tasks on both ends, it can only be shown that the LBFS rule is optimal within the class of policies that give priority to fixed tasks (see [20]). Note that this is a reasonable heuristic for preventing starvation of downstream workers that could arise from bottlenecks in upstream fixed stations.

### 3.2. Identically distributed processing times

Theorem 1(ii) can be extended to exclusive workers with more general training/skill sets under the assumption that service times are identically distributed within zones with at most one shared task upstream and no other shared tasks. As an example, consider worker  $i$  with  $T_i = \{1, 2, 3, 5, 7, 12\}$ , where tasks 1, 7, and 12 are shared. Then, the following theorem applies to the set  $\tilde{T}_i = \{1, 2, 3\}$  (a set of consecutive stations such that only station 1 is shared) if the service times for tasks 1, 2, and 3 are identically distributed.

**Theorem 2.** Consider worker  $i$ , and let  $\tilde{T}_i \subseteq T_i$  such that  $\tilde{T}_i = \{m : k_i \leq m \leq \ell_i\}$ , where

- (i) only  $k_i$  may be a shared task, and
- (ii) processing times for all tasks performed by worker  $i$  in set  $\tilde{T}_i$  are identically distributed.

Then the job completion process is maximized along every sample path by a policy that follows the LBFS rule for tasks in set  $\tilde{T}_i$ .

*Proof.* As in the proof of Theorem 1, we consider a time instant  $\tau$  when some policy  $\pi$  assigns worker  $i$  to some station  $\ell \in \tilde{T}_i$  other than  $k$ , the most downstream nonempty station in set  $\tilde{T}_i$ . The result will be proved by constructing a policy  $\tilde{\pi}$  that assigns worker  $i$  to station  $k$  and is such that  $D_i^{\tilde{\pi}}(\omega) \geq D_i^{\pi}(\omega)$  for every sample path  $\omega$ . Because service times are task independent in set  $\tilde{T}_i$ , a sample path  $\omega$  is defined by sequences  $S_j^w(l, \omega)$ ,  $j \notin \tilde{T}_i$ ,  $S_{k_i}^w(l, \omega)$ ,  $w \neq i$ , and

$S_{\tilde{T}_i}^i(l, \omega)$ ,  $l = 1, 2, \dots$ , where  $S_{T_i}^i(l, \omega)$  is the time duration of the  $l$ th service performed by worker  $i$  in set  $\tilde{T}_i$ . We define  $\tilde{\pi}$  as follows. After time  $\tau$  policy  $\tilde{\pi}$  is identical to  $\pi$  except when worker  $i$  is assigned under  $\pi$  to a station that is empty under  $\tilde{\pi}$ , in which case  $\tilde{\pi}$  assigns worker  $i$  to the most downstream nonempty station in set  $\tilde{T}_i$  upstream of the one served under  $\pi$ . This construction is valid because assumption (ii) ensures that service completions by worker  $i$  occur simultaneously under  $\pi$  and  $\tilde{\pi}$ . Moreover, for workers other than  $i$ , policy  $\tilde{\pi}$  can always mimic  $\pi$  because these workers can only be assigned to station  $k_i$  and the stations outside  $\tilde{T}_i$  where along any sample path there are at least as many jobs under  $\tilde{\pi}$  as under  $\pi$ . It can be shown that eventually we have coupling of  $\pi$  and  $\tilde{\pi}$  in terms of the number of jobs present in each station and the number of job completions. The details of the proof for both open and CONWIP lines are relegated for the sake of brevity to Pandelis and Van Oyen [20].

**Remark 1.** Under certain assumptions, Theorem 2 also holds for the minimization of the total holding cost along any sample path. Let  $h(j)$ ,  $j = 1, 2, \dots, N$ , be the holding cost per unit time incurred by each job in station  $j$ , and let  $H_t^\pi(\omega)$  be the total holding cost accrued on  $[0, t)$  under policy  $\pi$  along sample path  $\omega$ . Also, let  $h_t^\pi(\omega)$  be the total holding cost rate (summed over all jobs present in the system) at time  $t$  under policy  $\pi$ . Assuming that  $h(j)$  is a concave function, it can be shown by the same sample path arguments that  $h_t^\pi(\omega) \leq h_t^\pi(\omega)$  for all  $t$  and  $\omega$ . Therefore,  $H_t^\pi(\omega) \leq H_t^\pi(\omega)$  for all  $t$  and  $\omega$ .

As already stated, Theorem 2 extends Theorem 2.1(ii) of [1] which establishes the optimality of the LBFS rule among a worker’s fixed tasks in the class of nonidling, nonpreemptive policies (idling is permitted in our formulation). The result of Theorem 2 has also been proved in [1, Lemma 3.1] under more restrictive assumptions: workers of equal speeds and deterministic processing times that are either equal to one time unit or can take arbitrary values provided that preemptions are allowed. It may seem that giving priority to a worker’s fixed tasks over any shared task may always be optimal; however, Pandelis and Van Oyen [20] provided a counterexample.

#### 4. Optimal worker selection

In this section we consider instances in time at which two or more workers are competing for a single task available to them. There may be other jobs and workers in the system, but our focus is on those workers that are available and there is only one available job to which they can be assigned at time  $t$ . We restrict attention to cross-training patterns with workers arranged in consecutive training sets  $T_i = \{k : k_i \leq k \leq \ell_i\}$  with  $(k_i)$  and  $(\ell_i)$  being nondecreasing sequences. To permit U-shaped lines, we allow  $\ell_i > N$ , so to speak, in the sense that  $k > N$  and  $\ell_i > N$  are interpreted as their modulo  $N$  stations. For example, with a U-shaped line with  $N$  stations,  $T_i = \{k_i, k_i + 1, \dots, N, N + 1\}$  corresponds to  $T_i = \{k_i, k_i + 1, \dots, N, 1\}$ . When the workers have equal speeds, i.e. they are equally skilled, it is intuitively reasonable that preference should be given to downstream workers, so that the other workers are kept available for new tasks arriving from upstream. In the rest of the section we formalize conditions under which priority to downstream workers maximizes the job completion process along any sample path.

The following theorem refers to the case when two or more workers are competing for a single shared task,  $S$ . Let  $W = \{j, j + 1, \dots, j + K - 1\}$  be a set of  $K \geq 2$  workers arranged in consecutive training sets with  $j$  being the most upstream worker (with respect to the workflow direction).

**Theorem 3.** *Consider a time instant when the only task available to the workers in set  $W$  is  $S$ . Assuming that*

- (i) *the workers in set  $W$  have equal speeds,*
- (ii)  *$j$  is an exclusive worker,*
- (iii) *each of the workers  $j + 1, \dots, j + K - 1$  has at most one task shared with workers not in set  $W$ ,*

*there exists a policy that assigns a worker other than  $j$  to task  $S$  and has at least as many completions along any sample path as any policy that assigns worker  $j$  to  $S$  in the cases of CONWIP systems, and open systems where the first task in the line does not belong to the training sets of workers  $j + 1, \dots, j + K - 1$ .*

*Proof.* Let  $\pi$  be a policy that assigns  $j$  to  $S$ . We define an alternative policy  $\tilde{\pi}$  that assigns some other worker  $k \in W$  to  $S$ , keeps  $j$  idle until task  $S$  is completed, and is identical to  $\pi$  afterwards. Because workers  $j$  and  $k$  have equal speeds, the service time of task  $S$  is equal under policies  $\pi$  and  $\tilde{\pi}$  along any sample path. Our assumptions ensure that along any sample path it is not possible for  $\pi$  to assign  $k$  to some task during the service of  $S$ . Such a task could become available in one of three ways: by an external arrival in an open system, a service completion by an upstream neighbor of  $j$ , and a service completion by some worker  $m \notin W$ . The first scenario is not possible because  $k$  is not trained for the first task in the line. The second is not possible because  $j$  is busy at  $S$  and has at least one fixed task upstream of  $S$  (assumption (ii)). The third scenario would only be possible if  $k$  had more than one shared task with workers not in  $W$ , which is not the case (assumption (iii)). Therefore, the two policies are coupled at the time of service completion at  $S$ , implying that they have the same performance.

Theorem 3, which applies to all CONWIP and almost all open models, shows that an optimal policy would not give preference to the most upstream worker. However, it holds under rather restrictive assumptions and does not specify the worker to whom preference should be given. Nevertheless, it can provide a basis for good heuristic policies, particularly for systems with limited cross-training.

## 5. Conclusions

We studied the problem of the dynamic assignment of cross-trained workers in serial production lines in order to maximize throughput. We considered quite general models with respect to processing time distributions, worker speeds, and worker cross-training patterns. Our research into this class of systems suggests that optimal policies are very complex. Using exchange arguments, we were able to determine structural properties of policies that maximize the job completion process along any sample path, thereby providing some structure to a general class of models. Our results extended previous work in the literature and provided valuable insights that we expect will lead to good heuristic policies.

## Acknowledgement

The work of the second author was partially supported by the NSF, under grants DMI-0542063 and CMMI-1068638.

## References

- [1] AHN, H.-S. AND RICHTER, R. (2006). Dynamic load balancing with flexible workers. *Adv. Appl. Prob.* **38**, 621–642.
- [2] ANDRADÓTTIR, S. AND AYHAN, H. (2005). Throughput maximization for tandem lines with two stations and flexible servers. *Operat. Res.* **53**, 516–531.
- [3] ANDRADÓTTIR, S., AYHAN, H. AND DOWN, D. G. (2001). Server assignment policies for maximizing the steady-state throughput of finite queueing systems. *Manag. Sci.* **47**, 1421–1439.
- [4] ANDRADÓTTIR, S., AYHAN, H. AND DOWN, D. G. (2007). Dynamic assignment of dedicated and flexible servers in tandem lines. *Prob. Eng. Inf. Sci.* **21**, 497–538.
- [5] ANUAR, R. AND BUKCHIN, Y. (2006). Design and operation of dynamic assembly lines using work-sharing. *Internat. J. Production Res.* **44**, 4043–4065.
- [6] ARUMUGAM, R., MAYORGA, M. E. AND TAAFFE, K. M. (2009). Inventory based allocation policies for flexible servers in serial systems. *Ann. Operat. Res.* **172**, 1–23.
- [7] ASKIN, R. G. AND CHEN, J. (2006). Dynamic task assignment for throughput maximization with worksharing. *Europ. J. Operat. Res.* **168**, 853–869.
- [8] BARTHOLDI, J. J., III AND EISENSTEIN, D. D. (1996). A production line that balances itself. *Operat. Res.* **44**, 21–34.
- [9] BARTHOLDI, J. J., III, EISENSTEIN, D. D. AND FOLEY, R. D. (2001). Performance of bucket brigades when work is stochastic. *Operat. Res.* **49**, 710–719.
- [10] BISCHAK, D. P. (1996). Performance of a manufacturing module with moving workers. *IIE Trans.* **28**, 723–733.
- [11] GEL, E. S., HOPP, W. J. AND VAN OYEN, M. P. (2002). Factors affecting opportunity of worksharing as a dynamic line balancing mechanism. *IIE Trans.* **34**, 847–863.
- [12] GEL, E. S., HOPP, W. J. AND VAN OYEN, M. P. (2007). Hierarchical cross-training in work-in-process-constrained systems. *IIE Trans.* **39**, 125–143.
- [13] HOPP, W. J. AND VAN OYEN, M. P. (2004). Agile workforce evaluation: a framework for cross-training and coordination. *IIE Trans.* **36**, 919–940.
- [14] HOPP, W. J., TEKIN, E. AND VAN OYEN, M. P. (2004). Benefits of skill chaining in production lines with cross-trained workers. *Manag. Sci.* **50**, 83–98.
- [15] KIRKIZLAR, H. E., ANDRADÓTTIR, S. AND AYHAN, H. (2012). Flexible servers in understaffed tandem lines. To appear in *Production Operat. Manag.*
- [16] KIRKIZLAR, E., ANDRADÓTTIR, S. AND AYHAN, H. (2010). Robustness of efficient server assignment policies to service time distributions in finite-buffered lines. *Naval Res. Logistics* **57**, 563–582.
- [17] McCLAIN, J. O., SCHULTZ, K. L. AND THOMAS, L. J. (2000). Management of worksharing systems. *Manufacturing Service Operat. Manag.* **2**, 49–67.
- [18] McCLAIN, J. O., THOMAS, L. J. AND SOX, C. (1992). ‘On-the-fly’ line balancing with very little WIP. *Internat. J. Production Econom.* **27**, 283–289.
- [19] OSTOLAZA, J., THOMAS, J. AND McCLAIN, J. O. (1990). The use of dynamic (state-dependent) assembly-line balancing to improve throughput. *J. Manufacturing Operat. Manag.* **3**, 105–133.
- [20] PANDELIS, D. G. AND VAN OYEN, M. P. (2011). Sample path optimal policies for serial lines with flexible workers in zones. Tech. Rep. 11-03, Department of Industrial and Operations Engineering, University of Michigan.
- [21] PARVIN, H. *et al.* (2012). Fixed task zone chaining: worker coordination and zone design for inexpensive cross-training in serial CONWIP lines. To appear in *IIE Trans.*
- [22] VAN OYEN, M. P., GEL, E. S. AND HOPP, W. J. (2001). Performance opportunity for workforce agility in collaborative and noncollaborative work systems. *IIE Trans.* **33**, 761–777.