Manufacturing & Service Operations Management

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Benefits of Skill Chaining in Production Lines with Cross-Trained Workers: An Extended Abstract

Eylem Tekin, Wallace J. Hopp, Mark P. Van Oyen,

Benefits of Skill Chaining in Production Lines with Cross-Trained Workers: An Extended Abstract

Eylem Tekin
Department of Industrial Engineering and Management Sciences, Northwestern University, 2145 Sheridan Road C223, Evanston, Illinois 60208-3119
eylem@nwu.edu

Faculty Mentors: Wallace J. Hopp • Mark P. Van Oyen
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208
Department of Information Systems and Operations Management, Loyola University, Chicago, Illinois 60626

In recent decades, the shift from mass production of highly standardized products to small batch production of customized products has forced most manufacturing and service organizations to use some form of flexible capacity. One attractive form of flexible capacity is the use of agile workers that shift their capacity dynamically to where it is needed. The main motives for pursuing workforce agility can be classified as improved efficiency, enhanced flexibility, increased quality, and improved culture (Hopp and Van Oyen 2001). In this paper, we concentrate on the first motive and study the logistical benefits of workforce agility.

Designing an agile work system requires three levels of decisions. The first level of decision is whether or not to use flexible workers at all in a given system (e.g., as opposed to using some other form of flexibility or a traditional inflexible system). The second level of decision is how to decide which skill(s) are strategically most desirable for workers to gain (i.e., what skill pattern to introduce). The third level is how to assign the cross-trained workers to tasks over time to respond dynamically to congestion (i.e., what worker-allocation policy to use). In general, the third decision must be resolved to address the second decision, which must be resolved to address the first decision.

The literature on workforce agility focuses primarily on the third decision. Some consider scheduling of multiskilled workers based on clock time. Generally, this involves solving a constrained optimization problem to produce a workforce schedule that addresses a given demand forecast (e.g., see Berman et al. 1997, Campbell 1999, Lee and Vairaktarakis 1997). Other literature on worker allocation addresses systems with process variability and/or differing worker speeds and develops rules for dynamic allocation of cross-trained workers to tasks (e.g., Zavadlav et al. 1996, McClain et al. 2000, Bartholdi and Eisenstein 1996, Van Oyen et al. 2001). Typically, the objective is to describe optimal or near optimal policies that maximize the throughput or minimize the cycle time of the systems under consideration.

Although research on worker allocation (clock-based or dynamic) can offer valuable insights into the behavior of various skill patterns, it does not directly address the question of what skill pattern is best for a given environment. In this paper, we focus on the second level decision by contrasting the performance of different skill patterns operating under a good worker-allocation policy for each pattern. We also provide insights at the third decision level as we test a variety of heuristic policies. Our goal is to provide practical insights into the selection of an effective skill pattern.

Our analysis is set within a domain of asynchronous flow lines with the following characteristics: We assume there are N stations, with one worker assigned to each station (as his/her base station). Service times at a station are i.i.d. and independent of other stations. A CONWIP (CONstant Work-In Process) release policy is employed, because it is a simple, broadly applicable release policy and because it allows us to quantify the trade-off between WIP and cross-training as methods for buffering variability. Workers are assumed to be identical, always available, and able to switch between tasks without cost or changeover time. As such, our models are most applicable to short or U-shaped production lines (where workers can walk quickly between stations), or systems where tasks are performed on networked computers or phones (so tasks are routed to workers). Workers cannot work on the same job simultaneously, but more than one worker can work at a station if there are jobs available. This assumption of ample resources is especially suited to environments with generic or low-cost equipment (e.g., service systems where equipment consists of phones or computers and manual manufacturing environments where equipment consists of simple tools). Tasks cannot be preempted once begun, which is often the case because preemption often takes time and can affect quality.

In such environments, workforce agility can improve efficiency in two primary ways. First, lines that are imbalanced with respect to average work content will cause some workers to idle periodically. Cross-training allows such a worker to split her/his effort over time to improve her/his utilization and the resulting throughput (TH) of the line. We refer to this first phenomenon as *capacity balancing.* Second, variability in processing times will cause workers to idle occasionally even if average workloads are balanced between stations. This variability in processing times can be caused by a variety of factors, including supply disruptions, equipment failures, changes in product mix, etc. Cross-training can improve productivity by allowing a worker starved at one station to switch to another station to process available work there. We refer to this second benefit as *variability buffering.* An effective agile workforce design uses a reasonable amount of cross-training to achieve higher operational efficiency by both balancing capacity and buffering variability.

Now suppose we have a line with a specialist trained for each station and imbalanced work content across stations. The first question we address is which skills to provide to workers to balance the line. For this purpose, we introduce two skill pattern strategies, which we term *Cherry Picking* and *D-Skill Chaining.* Cherry-picking strategy assigns workers who have excess capacity to one or more station(s) that require help, so that the line is balanced with the minimum number of additional skills. This approach, which suggests borrowing capacity from underutilized workers to help at the bottleneck stations, is often used in industry (e.g., we have observed it in an IBM circuit board plant). It is also consistent with most optimization-based scheduling models for cross-trained workers. We show that determining the skill sets under cherry-picking strategy is equivalent to solving a transportation problem with fixed edge costs, where fixed edge costs are equal to one and flow costs are equal to zero. Although this problem is known as NP-Hard, it is not difficult to find the optimal cherry-picking strategy for moderately sized lines. In addition, we can bound the number of additional skills required to balance the line by $N - 1$, where N denotes the number of stations.

On the other hand, under the D-skill chaining strategy, each worker is cross-trained for her/his base station and for the next $D - 1$ stations (a total of D skills per worker). This skill pattern allows every worker to either directly or indirectly (via paths through other workers) redirect effort from her/his original task type to any other task type. As a simple example of 2-skill chaining, consider a line with four task types denoted by 1, 2, 3, 4, where the workers are trained for the following skill sets: $(1, 2)$, $(2, 3)$, $(3, 4)$, $(4, 1)$. Notice that every task type has exactly two workers capable of covering it and that stations are *chained* together by overlapping work assignments. We frame the capacity balancing problem under the D-skill chaining strategy as one of identifying the minimal value of D, such that D-skill chaining would be able to balance the line as WIP tends to infinity and provide the methodology to compute this minimal value.

To examine how cherry-picking and D-skill chaining strategies behave under conditions of variability, we specify effective dynamic allocation policies to assign the cross-trained workers to tasks over time. Numerical experiments with Markov Decision Process (MDP) models show that the optimal allocation policy under both strategies does not have any simple structural form and is therefore not generally practical to implement. So, we examine a range of easy-to-implement policies motivated by the literature and industrial practice to determine which are most effective under cherry picking and chaining. We classify these policies into three groups as static, queue-length-based and workload-based policies. Static policies consist of predetermined rules for workers to follow, regardless of the system state, and include the *fixed before shared*, *zoned craft*, *randomized*, and *priority* policies. Queue-length-based policies such as *maxqueue, maxqueuegap,* and *uniform buffer* allocate workers based on the information regarding the number of jobs at each station. The last group of policies is more sophisticated in the sense that it makes decisions based on the workloads at the stations rather than simple queue lengths. We term them *maxload, maxgap*, and *time buffer.*

To investigate the performance of cherry picking and D-skill chaining under different worker-allocation policies, we conducted a simulation study of various environments. To highlight the effects of capacity imbalance and variability, we varied the mean and coefficient of variation of process times and the number of bottleneck stations in the line. We observed each system at different throughput levels and found the WIP levels required to achieve these throughputs. We compared the performance of different policies by the average, minimum, and maximum percentage deviations from the best WIP level obtained by one of the policies, as well as the number of instances that each policy is best. All of the environments we considered can be balanced under chaining with $D = 2$. We also discussed situations where $D = 2$ is not enough to achieve capacity balancing separately and suggested a hybrid approach for designing an agile workforce strategy for such systems. The purpose of this simulation study was two fold. First, we used it to determine which allocation policies perform best for each of the two cross-training strategies. Second, we used it to compare the robustness of the two strategies in various environments.

Our first important observation is that the queue-length-based policies outperform the more sophisticated workload-based policies for both strategies in almost all cases. When process times differ greatly, a large queue can be needed to produce a sufficient workload to attract a flexible worker. This requires extremely high WIP levels for workers to achieve high utilization, and therefore high throughput. We have also observed that, in general, the performance of an allocation policy under the cherry-picking strategy depends on the number of bottlenecks in the system and the number of workers who are cross-trained for a station. This could make implementation more difficult in practice. On the other hand, although static policies perform poorly under the two-skill chaining strategy, queue-length-based policies, particularly uniform buffer, achieve the best performance in almost every case.

The number of additional skills needed to balance capacity under cherry picking is clearly less than that of the D-skill chaining. This raises the question of whether the additional cross-training required in chaining is worthwhile. Because by definition, both cherry picking and skill chaining balance the line, the value of chaining rests on how effectively it buffers variability. We address this question in three steps by: (1) comparing systems with equal numbers of skills by using a partially chained skill pattern rather than a full chain (i.e., workers at the bottleneck stations are not cross-trained but others are cross-trained according to the chained pattern), (2) comparing the economics of a fully-chained system to those of a system using a cherry-picking strategy, and (3) investigating whether there is a loss in the system's performance if we omit the skills given to the bottleneck workers in the two-skill chaining strategy. In the first step, we show that partial chaining achieves higher throughput than cherry picking at low WIP levels. Hence, chaining is effective for running the system lean. The improvement in throughput via partial chaining is most pronounced in high variability cases, and up to a 31% increase can be observed. Hence, indirect cross-training via chaining can be superior to direct cross-training to help bottleneck because it makes spare capacity more flexible. As a second step, to fully evaluate the benefits of chaining we propose a simple model for comparing the relative economics of cross-training versus efficiency improvements. For highly variable processing times and low capacity imbalance our results suggest that two-skill chaining is highly desirable. This is consistent with our previous results and confirms that chaining is very attractive whenever cross-training is easy (low cost) and may well be a good idea even when cross-training is costly, provided there is enough variability in the system.

After analyzing strategies with partial chaining and placing specialist workers at the bottleneck station(s), we addressed the question of whether one needs a full chain or only a partial chain that avoids cross-training bottleneck workers, because they can be fully utilized with only one skill. This would appear to be very intuitive in practice and suggest that the value of ''completing the chain'' should be minimal because, in general, cross-training exhibits diminishing returns in the number of skills. However, it turns out that when there is variability in the system, this may not be the case. The incremental improvement from the last skill (i.e., training the bottleneck worker to help out elsewhere) can be the largest. Especially for the cases where there is no severe bottleneck, the benefit of completing the chain is surprisingly large (e.g., 22%) for almost all WIP levels. Completing the chain allows capacity from any station to be shifted (indirectly) to any other station, and this effect benefits the entire line. Moreover, completing the chain has the most significant effect when we complete the chain by cross-training a bottleneck worker at a low-variability station to help out at a high-variability station.

In summary, our results provide the following managerial insights into the choice of skill-pattern strategies and effective worker-coordination policies. (1) It can be ineffective to simply cross-train to achieve capacity balance in a direct way (i.e., cherry picking). Instead, strategies that rely on chaining to shift capacity indirectly between stations are much more effective, particularly in systems with high variability. (2) Simple queue-length-based policies are more effective than complicated workload-based policies for two-skill chaining skill patterns. Two-skill chaining is relatively robust across several queue-length-based policies. This means that the allocation policy can be chosen to fit worker preferences or other practical system considerations without significant performance loss. (3) The flexibility created by two-skill chaining is so substantial that, rather than yielding diminishing returns to the number of skills added, two-skill chaining frequently gains the greatest marginal benefit from the addition of the last skill to the bottleneck worker.

Most real life systems possess both short-term variability in their operations and longer-term shifts that can cause system bottlenecks to move over time. The fundamental issue that makes an agile workforce design attractive to a firm is its ability to shift capacity in response to changes in the system. An investment in flexible capacity (in our case, investment in an additional skill) has the most value when it can impact any operation in the system. This is the key idea motivating skill-chaining strategies. Our investigations suggest that two-skill chaining or partial-chaining strategies may offer tremendous potential as a tool for implementing workforce agility for serial lines. Queue-based policies are good ways to allocate workers under two-skill chaining, while more sophisticated and harder to implement workload-based policies are less effective. Hence, the robustness of two-skill chaining across a number of simple (i.e., easy to understand, easy to implement) queue-length-based policies makes it even more attractive.

References

Bartholdi III, J.J., D.D. Eisenstein. 1996. A production line that balances itself. *Oper. Res.* **44**(1) 21–34.

Berman, O., R. Larson, E. Pinker. 1997. Scheduling workforce and workflow in a high volume factory. *Management Sci.* **43**(2) 158–172.

Campbell, G.M. 1999. Cross-utilization of workers whose capabilities differ. *Management Sci.* **45**(5) 722–732.

Hopp, W.J., M.P. Van Oyen. 2001. Agile workforce evaluation: A framework for cross-training and coordination. *Proc. 2001 NSF Design and Manufacturing Grantees Conf.*, Tampa, FL.

Lee, C.Y., G.L. Vairaktarakis. 1997. Workforce planning in mixed model assembly systems. *Oper. Res.* **45**(4) 553–567.

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

Van Oyen, M.P., E.G.S. Gel, W.J. Hopp. 2001. Performance opportunity of workforce agility in collaborative and noncollaborative work systems. *IIE Trans.* **33**(9) 761–777.

Zavadlav, E., J.O. McClain, L.J. Thomas. 1996. Self-buffering, self-balancing, self-flushing production lines. *Management Sci.* **42**(8) 1151–1164.